

博士論文

最適設計における
逐次感度近似法の提案と
その流れ場の
トポロジー最適化への適用

米倉一男

概要

本論文では最適設計手法とその流れ場への適用を取り扱う。最適設計は学問的な研究と並行して実務上でも急速に応用が進んでいる。実用上の課題として、最適設計の計算コストが大きくなりがちであることが挙げられる。既存の最適化手法では、場の変数を解き、その結果を用いて設計変数を更新する、という方法が主流である。この場合、場の変数の数値計算を繰り返す必要があり、非線形方程式に対しては計算コストが増大する一因になっている。特に流れ場の方程式は非線形性が強く、繰り返し計算が大きな負担になる。これに対し 固体の最適設計に対して提案された SAND 法 (Simultaneous Analysis and Design) や流体に対する one-shot approach のように、設計変数と場の変数をそれぞれ最適性条件と場の支配方程式を満たすように同時並行して更新する方法も提案されている。本論文では、固体に対する SAND 法などと同様に場の変数の計算と設計変数の更新を並行して行う手法を構築する。特に非定常な流れの方程式に従う場に適用することを考え、ある時刻における場の変数を用いて、計算上の一ステップ先の目的関数を小さくするような最適化方法を構築する。この工夫により、繰り返し計算の負担が小さくなり、計算コストが減少することが期待される。場の変数の計算は既存の数値計算法を用い、設計変数の更新には感度法に基づいた更新式を用いる。このとき感度の計算は、支配方程式を必ずしも満たさない場の変数をもとに計算される近似的な値である。そこでこの手法を逐次感度近似法 (instantaneous Sensitivity Approximation method; iSA 法) と呼ぶ。このように近似した感度を用いて設計変数を更新し解を得るが、このとき得られる解は必ずしも最適解ではない。本論文ではまず iSA 法のアルゴリズムを構築し、それにより得られる解の性質について述べる。

次に iSA 法を流れ場のトポロジー最適化問題に適用する。流れ場の数値計算法としては格子ボルツマン法 (Lattice Boltzmann Method; LBM) を用い、対象とする流れ場はまず遅い流れとする。遅い流れに適用する場合には、領域の表現方法として既存手法でも広く採用される多孔質媒体法を用いる。この手法は、設計領域が多孔質媒体で充たされていることを想定し、その空隙率を変数として最適化を行う手法である。数値計算例を用いて手法の妥当性と計算速度について述べる。多孔質媒体法を用いる場合には流体領域と固体領域を明確に区別することが難しく、流体領域とも固体領域とも判別の難しい領域が存在することも示す。

次に設計変数の更新方法を改良し、部分的に目的関数の Hesse 行列を用いることでより良い収束速度になるように工夫する。Hesse 行列が対角行列になるような定式化を行うことで、疎行列計算だけで最適化計算を実施し、計算時間をおさえる。Hesse 行列を用いることにより、最適解近傍での収束速度が速くなる。数値計算例を用いて勾配法を用いる場合との収束速度の違いを述べる。また、勾配法を用いる場合には流体領域と固体領域の境界部分が明確でないという特徴があるが、Hesse 行列を用いた場合には、両領域が視覚的に明確に判別できることを示す。次に iSA 法の適用対象をより速い流れに広げることを想定し、境界条件を適切に考慮した手法を構築する。多孔質媒体法では流体領域と固体領域が明確に区別されず、実際の流れを必ずしも模擬しない場合がある。これは特に早い流れにおいて顕著である。そこで多孔質媒

体法ではなく、level-set 関数を用いることで固体領域と流体領域を明確に区別してその境界上に壁面境界条件を適用する。壁面境界条件としては非滑り条件である bounce-back 法を用いる。その後、境界条件を適切に考慮した感度を定式化して最適化アルゴリズムを構築する。この方法では、最適化の過程で壁面が移動することにより、ある時点で固体領域であった領域を次のステップでは流体領域に変化させる必要が生じる。このような場合には、流れ場に不必要な乱れをひきおこさないために、格子点の初期化を行う手法を構築する。

最後に、工学的に重要な対象である、時間周期的な境界条件のもとでの流れ場に適用対象を広げる。このとき、感度の計算に随伴変数法を用い、境界条件の一周期分の流れ場の変化を考慮したアルゴリズムを構築して最適化を行う。随伴変数法を用いることでより早い流れにおいても妥当な最適解を得ることができる。随伴変数法では、主変数を解いた後で、時間を逆行するようにして随伴変数を求める必要がある。このとき、随伴変数の計算には同じ時刻における主変数の情報が必要になる。したがって本手法を適用するためには計算時間と計算に必要な記憶容量が増大するという問題がある。そこで並列計算を行うことでこれらの問題を解決する。並列数を変えた計算を実施し、並列化による高速化率について述べる。

以上のように、本論文では iSA 法を構築し、それを様々な流れ場のトポロジー最適化に適用する。iSA 法は計算時間が短いことが期待されるが、実施した数値計算例では実用上十分短い時間で計算できる。特に遅い流れに対して多孔質媒体法を用いる場合には極端に短い時間で実施できる。iSA 法により得られる解は必ずしも最適解ではないが、近似的な解を求める代わりに計算時間を短くすることができるという特徴がある。このとき、解は近似的な最適解であっても工学的には十分に有意義であり、計算時間が短縮される点は大きな利点である。

キーワード 逐次感度近似法, トポロジー最適化, 流れ場, 格子 Boltzmann 法

目次

第 1 章	諸言	1
1.1	最適設計とその実用化	1
1.2	既存の形状の最適化手法とその問題点	2
1.3	本論文の目的	4
1.4	本論文の構成	6
1.5	記号の定義	6
第 2 章	準備：格子 Boltzmann 法の定式化	10
2.1	格子 Boltzmann 法	10
2.2	格子 Boltzmann 法の基礎方程式	11
2.3	衝突モデル	13
2.4	多孔質媒体中の流れの BGK モデルによる解法	15
2.5	Bounce-back 条件	17
第 3 章	iSA 法の提案とその応用	18
3.1	最適化問題の定義	18
3.2	定常な場に関する最適化問題に対する iSA 法の定式化	21
3.3	時間周期的な場に関する最適化問題に対する iSA 法の定式化	23
3.4	未定乗数の決定方法	26
3.5	iSA 法の典型的なアルゴリズム	28
3.6	iSA 法の解の存在とアルゴリズムの停止に関する十分条件	29
3.7	iSA 法の応用可能性	29
第 4 章	多孔質媒体モデルによる最適化	35
4.1	多孔質媒体モデルによる流れ場のトポロジー最適化	35
4.2	エネルギー損失最小化問題と iSA 法による解法	36
4.3	数値計算例	39
4.4	iSA 法と多孔質モデルを用いた場合のトポロジー最適化の結論	62
第 5 章	Newton 法を利用した高速化	63

iv 目次

5.1	多孔質モデルの欠点と感度法の収束性	63
5.2	部分的に Newton 法を用いた最適化手法	65
5.3	数値計算例	69
5.4	解の最適性と収束速度に関する議論	79
5.5	部分的に Newton 法を利用した手法の結論	80
第 6 章	境界条件を適切に考慮した定常流れの最適化	81
6.1	トポロジー最適化問題における境界条件の取り扱い	81
6.2	最適化問題の定式化	82
6.3	最適化の過程における場の変数の初期化	85
6.4	最適化問題の解法	88
6.5	数値計算例	91
6.6	iSA 法を用いて境界条件を適切に考慮した場合のトポロジー最適化の結論	112
第 7 章	時間周期的な境界条件の下での流れ場の最適化	113
7.1	トポロジー最適化における時間周期的な流れ場の取り扱い	113
7.2	最適化問題の定式化と iSA 法による解法	114
7.3	iSA 法と既存手法の関係	121
7.4	数値計算例	122
7.5	並列効率	124
7.6	iSA 法を用いた時間周期的な境界条件の下での流れ場のトポロジー最適化の結論	125
第 8 章	結言	127
	謝辞	129
付録 A	LBM の定式化に関する補遺	139
A.1	BGK モデルにおける壁面境界条件の定式化	139
A.2	BGK モデルにおける速度境界条件の定式化	139
A.3	D3Q19 モデル MRT-LBM における速度境界条件の定式化	140
A.4	D3Q19 モデル MRT-LBM における圧力境界条件の定式化	141
A.5	行列 M の定義	143
付録 B	既存の文献における KKT 条件の定式化	145

第 1 章

諸言

1.1 最適設計とその実用化

設計は最適化と不可分である。機械製品の設計に当たって、設計者は要求性能を満たしたうえでより性能が良くなるように設計を行う。この、最適な設計解を探求するという考えは機械設計の本質であるといつて良い。実務の場面における最適設計は広く実用化されており [29, 40, 51, 75, 85, 87], さらにその適用範囲が広がり続けている。その基盤となるのが数理的な最適化手法の研究である。ただし数理的な意味での最適化と、上述の“最適設計”とは、厳密には異なる。数理的な意味での最適化は厳密な最適性を求める。またそこには興味深い数理構造が隠れていることも多い。一方で実際の設計最適化は、先に目標とする性能値が設定され、それを上回る性能を目指すことが多く、必ずしも最適である必要はない。この意味で、実際の設計における“最適設計”は、数理的な最適化よりも曖昧な概念である。

今日の最適設計は計算機と数値計算手法の発達とともに発展した。有限要素法 [8] や有限体積法 [74] などの登場により、場の変数の挙動を数値的に解析することが可能になり、これを製品開発や設計に応用するようになってきている。このような設計手法は CAE (Computer Aided Engineering) と呼ばれ、現在に至るまで重要度を増している。近年では計算機性能の向上に伴って CAE で取り扱える対象が広がっており、スーパーコンピュータを使用した大規模計算を行う例も報告されている。例えば流体力学分野では坪倉ら [22] による京コンピュータを使用した精密な流体計算などが挙げられる。これに伴い、最適化もより大規模な計算を行う方向に向かっている。大山ら [54] は京コンピュータを利用することでこれまでよりも多くの設計点を考慮して発見的な手法による最適化を行った。数値的な最適設計の基本的な考え方は、CAE において数値的に表された形状を少しずつ変えながら、目的関数を最小 (最大) 化するような形状を探索することである。最適化計算はその原理上、場の変数の数値計算よりも計算コストが大きい。そのため、場の変数が計算できても、最適化計算は困難であるような対象が存在する。特に大規模な問題においては、場の変数の計算が可能な対象と最適化計算が可能な対象の乖離が大きい。そこで、大規模計算を用いたより良い設計を行うため、計算コストの小さな最適化手法が求められている。また京コンピュータ [26] に見られるように、近年の計算機性能の向上は CPU 単体の能力向上よりも、並列化ないしクラスタ化による処理能力の向上に

依っており、アルゴリズムの構築においても並列化を念頭に置いた実装が求められる。

数値的な最適設計における手法は、発見的な手法と感度法に基づく手法に大別される。発見的な手法は遺伝的アルゴリズム [20, 27], ニューラルネット [61], 応答曲面法 [43], 実験計画法 [50, 52] などを含む手法である。これらの発見的な手法は適用できる数理モデルの範囲が広く、汎用ソフトウェアも多く利用しやすい。ただし変数の数が限られ、また得られた解に局所的な最適性が保証されないという特徴がある。少ない数でなるべく効果的な変数を定義するために主成分分析を用いる [85] などの工夫がされ一定の成果を上げているものの、本質的な解決には至っていない。これらの報告の多くは発見的な最適化手法である。その理由として、企業における実際の製品設計では必ずしも厳密な最適性が必要とされない点あげられる。製品設計において考慮すべき点は性能だけでなく、設計にかけられる期間やコストなど、多くの事象が関係する。また外力や材料定数などの設計条件は不確実性が大きく、幅広い設計条件を考慮する必要がある。さらに製造方法や製造コストなどは数値化が難しく、たとえ数値化するにしても何らかのモデル化を用いた近似的な表現になる。このように実際の製品設計では不確実性が大きく、全ての事象を考慮した数理モデルを作ることは困難である。そのため、想定する数理モデル上で厳密な最適性を議論することが、あまり意味を持たない場合もある。従って最適設計の実用化と普及を考えた場合、短時間で計算でき、なるべく最適解に近い設計解が得られれば十分である場合が多い。

一方で感度法に基づく手法は、感度あるいは設計感度と呼ばれる値に基づいて最適化計算を行う手法である。感度とは目的関数や Lagrange 関数の設計変数による一階微分であり、多くの場合に感度が零になったとき、最適性の一次の必要条件を満たすように定義される。感度法に基づく手法では変数の数を多く取ることができ、また局所最適性が保証されることが多い。そのため幅広い設計空間から解を探索することができ、概念設計など設計の初期段階で使用されることが多い。ただし適用できる数理モデルの範囲が狭いことが実務における障害になっている。前述した不確実性を考慮した最適化手法の研究もある [48, 68, 83] が、一般的には考慮が難しい。このような違いのある両者であるが本論文では感度法をもとに、自由度が大きいという条件のもとで最適化を行う。次節以下では感度法を念頭に述べる。

1.2 既存の形状の最適化手法とその問題点

最適設計問題は、設計変数の設定方法によって大きく三種類に大別される。一つは部材寸法を設計変数とする寸法最適化問題である。この場合は寸法のみが変化するので、構造物の形状が大きく変化することはない。寸法最適化問題は三種類の中ではもっとも単純な問題といえ、実際の製品設計に適用された例は多い。二つ目は形状最適化問題である。形状最適化問題では、形状を自由曲線で表し、そのパラメータを設計変数とする。自由曲線で表せる分、寸法最適化に比べて設計解の空間が広い。一方で形状のトポロジーは変化しないという特徴がある。形状最適化手法を実際の製品に応用する研究としては倉田ら [40] などが挙げられ、実用が進んでいる。三つ目はトポロジー最適化問題である。トポロジー最適化問題は、トポロジーの変化を許容するような最適化問題と定義される。図 1.1 にトポロジー最適化の概念図を示す。初期

状態（図 1.1 (a)）では流体領域は円板と位相同型だが、最適化の後（図 1.1 (b)）では流体領域は穴あき円板と位相同型である。このように最適化の前後で位相（トポロジー）が変化することを許容する最適化手法をトポロジー最適化と呼ぶ。

形状最適化とトポロジー最適化の用語の定義はすでに述べたとおりだが、実用上は違う見方が可能である。一般的な形状最適化手法では、初期形状においてメッシュを生成し、そのメッシュを変形させることで最適化を行う。このとき、メッシュの変形の度合いが一定値を超えるとメッシュが破綻する。またメッシュが破綻しないとしても、数値計算に適さないようないびつなメッシュ形状になることも多い。従って形状最適化では大きな形状変更を可能にするためには様々な工夫が必要である。これに対しトポロジー最適化手法は、一般にメッシュの形状を変更せず、各点での密度などを変更して最適化を行う。そのため大きな形状変更をしたとしてもメッシュは変形せず、破綻することはない。このように、トポロジー最適化は形状変化の自由度が大きいという点が実用上重要な点である。以上のように、三種類の最適化手法のうち、トポロジー最適化が自由度が最も大きいために注目を集めており、実際の製品設計へも普及しつつある。

トポロジー最適化手法は、Bendsøe and Kikuchi [3] により最適化問題を構造物の材料分布問題に帰着して解く密度法が提案されて以来、構造的な性能を考慮した分野で研究が進んだ。最初は構造分野で研究が進み、基礎的な知見も多く蓄積された。特に境界を明確に表現して取り扱う level-set 法 [2, 80] は広く利用されている。その後、構造トポロジー最適化手法を元に、熱伝達問題 [19] などに対象を変えた手法も提案され、適用先が広がっている。構造トポロジー最適化で広く用いられている SIMP 法や MMA 法 [4] では、場の方程式を最適化問題の外側で解き、その値を用いて最適化問題を解く。これに対し SAND 法 (Simultaneous analysis and desing) [24] という、場の変数と設計変数の両方を最適化問題の変数としてとらえ、場の方程式を制約条件として最適化問題を記述し、両方の変数を同時に更新する手法もある。SAND 法は場の方程式を含む最適化問題が半正定値計画問題などの性質の良い最適化問題に帰着される場合に用いられる一方で、汎用の有限要素法ソルバなどと組み合わせることができず、簡易に使用できないという特徴があるとされている [4]。

構造に対する最適化手法と同様に流れ場のトポロジー最適化手法も研究されている [49]。流れ場の数値計算 (Computational Fluid Dynamics; CFD) 手法は各種提案されているが、トポロジー最適化においても有限要素法 [21]、X-FEM 法 [37]、有限体積法 [19]、格子ボルツマン法 [32, 35, 36, 38, 57, 58, 79] など様々な手法に対して最適化手法が提案されている。対象とする流れ場は低 Reynolds 数流れが、高 Reynolds 数流れに対する手法よりも多い。これは、Reynolds 数が低い方が数値的に解きやすいためである。さらに固体と流体の連成問題に対するトポロジー最適化手法も提案されている [86]。

第 1.1 節でも触れたように、最適化手法は計算コストが大きくなりがちである。既存の形状の最適化手法の多くは、ある設計変数の値に対して場の変数を解き、目的関数に対する感度を計算する。その後、感度を元に設計変数を更新し、これを繰り返して最適解を得る、というアルゴリズムを採用している。このとき、設計変数を更新するたびに場の変数を計算する必要がある。ところが流れ場の方程式のように非線形性が強い場合には、場の変数を計算するコスト

4 第1章 諸言

が大きい。計算コストの大きな数値計算を繰り返し実施することで、全体の計算時間も長くなる。この点に既存の最適化手法の課題があった。構造最適化問題に関して言えば、線形弾性体を扱う場合には場の方程式の一回当たりの計算コストが小さく、前述の課題は大きな問題にはならない。ただし幾何学的非線形性を扱う場合には計算コストが問題になる。また近年になって流体のトポロジー最適化が注目を浴び、ここでも計算コストが問題となっている。流れ場の方程式である Navier–Stokes 方程式は非線形性が強く、一回当たりの計算コストが大きい。特に Reynolds 数が高くなった場合、一般的に格子点数を増やす必要があり、また収束に要する計算時間も増大するため、計算時間が飛躍的に増大する。このため、流れ場の最適化については実用的には単純な問題しか解けなかった。

実際に設計で用いる場合の流れ場の計算は、大規模になりがちである。特に中から高 Reynolds 数流れについては、実験結果を数値計算結果で模擬するためにはなるべく格子点数を多くする方が良いことが知られている [77]。また前述したように、実際の設計に活用するためには、大規模な問題であっても短時間で解けることが重要である。そのため、計算量の少ない最適化手法の提案は重要な課題であった。このような課題に対して、one-shot approach [56, 64] が提案され、流れ場の形状最適化において計算時間の短縮に成功している。One-shot approach では、偏微分方程式を制約条件とする最適化問題を対象とする。偏微分方程式を含む Lagrange 関数を定義し、Lagrange 関数の停留値問題を解くことで最適解を得る。このとき、偏微分方程式は Lagrange 関数に含まれており、直接偏微分方程式を解くことはない。このように one-shot approach により計算時間が短縮されたものの、流れ場のトポロジー最適化に対しては未解決であった。

1.3 本論文の目的

本論文の目的は二つある。一つ目の目的は計算コストの小さい最適設計手法を構築することである。先述のように、設計の最適化手法における課題の一つは、計算時間が長いことである。多くの設計の最適化手法では、ある設計変数に対して場の変数を求めたのちに、設計感度を計算し、設計変数を更新する。このとき、場の方程式を繰り返し解いて場の変数を求めることになる。特に流れ場の計算は計算量が多く、これを繰り返し解くことが負担になっている。

既存の最適化手法は、場の変数の更新と設計変数の更新を別々に行っている。これは常に実行可能空間を移動して設計変数を更新し、最適化計算を行うことに相当する。これに対し、本論文では場の支配方程式の収束計算と最適化の収束計算を同時に行う手法を構築する。これは、最適解以外の解は場の方程式を満たすとは限らず、最適解において場の方程式が満たされることを意味する。本手法の考え方は、場の変数と設計変数を並行して更新する点で SAND 法と似ているが、本手法は場の数値モデルとして格子ボルツマン法などのように変数が時間発展するようなモデルを想定して構築される点が SAND 法と異なる。この手法により、計算時間が短くなることが見込まれる。この見込みには理論的な裏づけはないが、実際に数値計算例により短時間で最適化計算ができることが示される。本手法を iSA (instantaneous Sensitivity Approximation) 法と呼ぶ。この名称は、本手法における感度が、ある瞬間におい

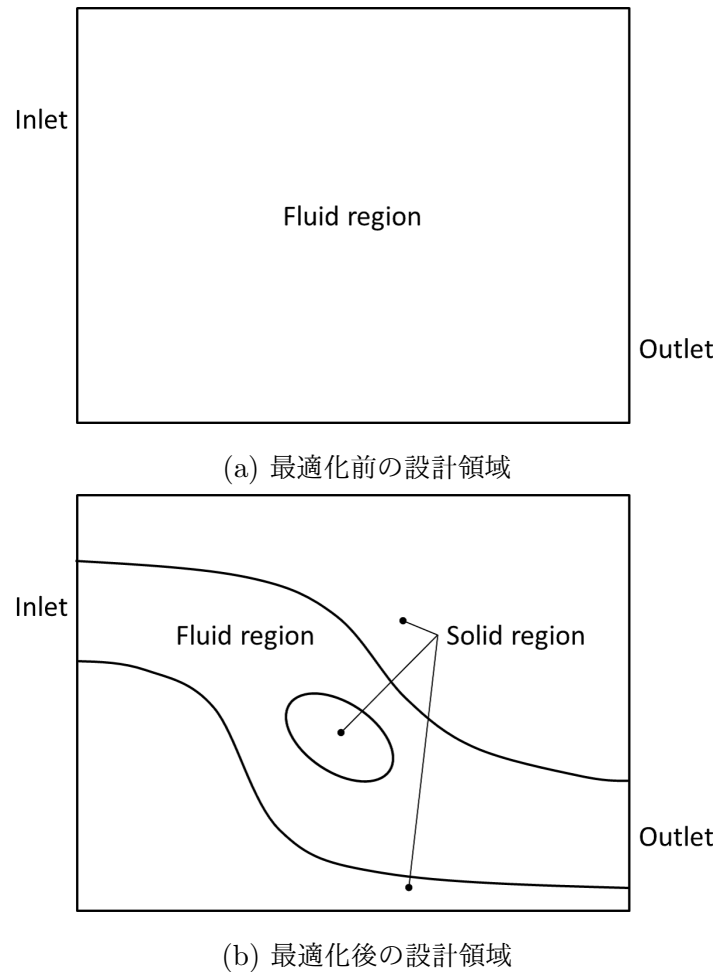


図 1.1. トポロジー最適化の概念図.

て一般的な手法による感度を近似することに由来する。iSA 法は流れ場の最適化に限らず一般的な最適設計問題に対して定式化される。またあるクラスの問題に対して、iSA 法により得られる解の最適性とアルゴリズムの停止性を保証できる。

二つ目の目的は、iSA 法を流れ場のトポロジー最適化問題に適用して短時間で最適解を求める手法を構築することである。本論文では大きく分けて三種類の流れ場を対象に手法を構築する。一つ目は低 Reynolds 数流れである。この流れ場に対しては、領域を多孔質媒体で近似し、その空隙率を変数として最適化する手法が有効とされている [7]。本論文でもこの多孔質媒体モデルを用いた上で iSA 法に基づいて手法を構築する。そのうえで数値計算例を用いて既存手法と比較を行う。二つ目は中程度の Reynolds 数の流れである。この流れ場では、力学的に妥当な流れ場を得るために、流体領域と固体領域の境界を明確に捉えて境界条件を適切に適用する必要がある。また格子点数も多く必要である。既存手法ではこのクラスの問題を解くことは簡単ではなかった。そこで iSA 法による最適化手法を構築し、実用的な時間で解くことができることを示す。三つ目は時間周期的な境界条件もとの流れ場である。実用上重要な流れ場として、レシプロエンジンの排気ガスや回転機械の出口流れなどがあるが、これらは境

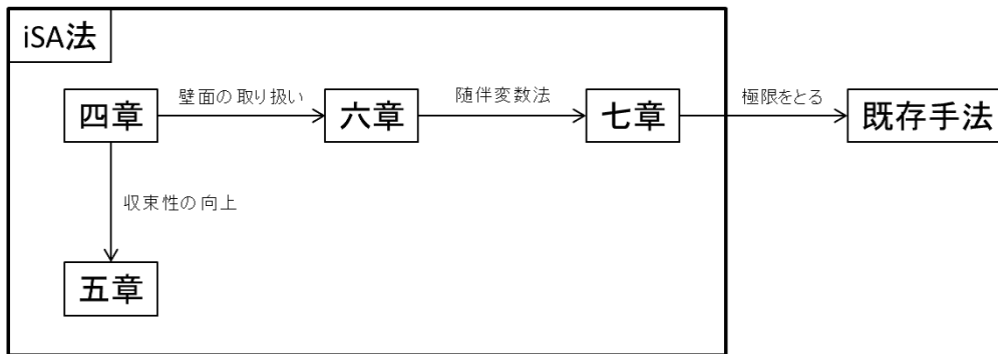


図 1.2. 各章の関係.

界条件が周期的に変動する流れ場と捉えることができる。このクラスの問題は既存の手法では計算時間が長く解くことは困難であった。そこで iSA 法を用いた最適化手法を構築し、またそれが実用的な時間内で解を求められることを示す。最後にアルゴリズムの並列実装について述べ、その効果を調べる。

1.4 本論文の構成

本論文は次のように構成される。第2章では流れ場の数値計算手法である格子ボルツマン法 (Lattice Boltzmann Method; LBM) を定式化する。第2章の内容は先行研究の内容を本論文で使用するために整理したものである。続いて第3章では iSA 法を導く。この定式化は一般的なものであり、流れ場に限定されない。また iSA 法の解の最適性や、応用の可能性についても述べる。次に第4章以降で流れ場のトポロジー最適化手法を iSA 法に基づいて構築する。第4章では Reynolds 数の低い流れに対して、多孔質媒体モデルを用いた定式化を行う。第5章では第4章で構築した手法を元に、目的関数の Hesse 行列を利用した定式化を行うことで収束性を向上させる。第6章では流体領域と固体領域の境界面上で境界条件を適切に考慮した最適化手法を構築する。第7章では境界条件が時間周期的な流れ場に対して最適化問題を定式化し、その解法を構築する。第7章の結果は、ある条件のもとで極限を取ることで既存手法 [79] と一致する。第3章から第7章までは図 1.2 のような関係にある。第4章から第7章ではそれぞれの章で数値計算例を示し、手法の妥当性を検討する。第8章で本論文全体の結論を述べる。

1.5 記号の定義

本論文で使用する記号は次のように約束する。まず設計領域全体を Ω とし、流体領域を Ω_f 、固体領域を Ω_s とする。流れ場の境界を Γ とし、そのうち速度境界を Γ_u 、圧力境界を Γ_p 、固体壁を Γ_w で表す。時間は t で表し、反復計算における計算ステップを下付き添え字 k で表す。ベクトルは縦ベクトルとし \boldsymbol{x} などの太字で表す。すべての要素が 1 であるベクトルを $\mathbf{1}$

とする。転置記号は \top とする。 \mathbf{u} は場の変数とし、 ϕ を設計変数とする。また \mathbf{x} は座標ベクトルとし、二次元の場合は $\mathbf{x} = (x, y)^\top$ 、三次元の場合は $\mathbf{x} = (x, y, z)^\top$ とする。 δt を時間の離散化幅とし、 δr を離散化した空間での格子点一つあたりの体積ないし面積とする。 Ω 上で定義される変数 $a(\mathbf{x})$ と $[0, T]$ で定義される変数 $b(t)$ に対し、時間と空間を離散化した場合には積分を次式で計算する：

$$\int_{\Omega} a(\mathbf{x}) d\mathbf{x} = \sum_{\mathbf{x} \in \hat{\Omega}} a(\mathbf{x}) \delta r,$$

$$\int_0^T b(t) dt = \sum_{k \in \{k \in \mathbb{Z} | 0 \leq k \delta t \leq T\}} b_k \delta t.$$

ただし $\hat{\Omega}$ は空間 Ω を離散化したものとし、 $b_k = b(k \delta t)$ とする。ベクトル $\mathbf{u} \in \mathbb{R}^3$ に対して空間座標についての微分演算子は次のように定義する：

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{pmatrix},$$

$$\Delta \mathbf{u} = \begin{pmatrix} \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \\ \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \\ \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \end{pmatrix}.$$

ある関数 E を任意のベクトル $\mathbf{u} \in \mathbb{R}^N$ で微分したものは横ベクトルとし、

$$\frac{\partial E}{\partial \mathbf{u}} = \left(\frac{\partial E}{\partial u_1}, \frac{\partial E}{\partial u_2}, \dots, \frac{\partial E}{\partial u_N} \right)$$

と定義する。またベクトル値関数 $\mathbf{v} \in \mathbb{R}^M$ をベクトル $\mathbf{u} \in \mathbb{R}^N$ で微分したものは

$$\frac{\partial \mathbf{v}}{\partial \mathbf{u}} = \begin{pmatrix} \frac{\partial v_1}{\partial u_1} & \frac{\partial v_1}{\partial u_2} & \cdots & \frac{\partial v_1}{\partial u_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_M}{\partial u_1} & \frac{\partial v_M}{\partial u_2} & \cdots & \frac{\partial v_M}{\partial u_N} \end{pmatrix}$$

と定義する。

また \mathbf{u} が速度ベクトルであるとき、歪み速度テンソル ε は $\varepsilon(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)$ で定義する。ベクトルのノルムは、特に断らない場合は l_2 -ノルムとするが、一部では明記した上で最大値ノルムも使用する。 $\mathbf{x} \in \Omega$ で定義される $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^{\dim}$ に対する l_2 -ノルムは

$$\|\mathbf{u}(\mathbf{x})\|_2 = \left(\int_{\Omega} \sum_{i=1}^{\dim} (u_i(\mathbf{x}))^2 d\mathbf{x} \right)^{\frac{1}{2}}$$

で定義し、同様に最大値ノルムは

$$\|\mathbf{u}(\mathbf{x})\|_{\infty} = \sup \left\{ \left(\sum_{i=1}^{\dim} (u_i(\hat{\mathbf{x}}))^2 \right)^{\frac{1}{2}} \mid \hat{\mathbf{x}} \in \Omega \right\}$$

で定義する.

行列は A などの大文字で表し, 単位行列を I とする. 二階以上のテンソルは ε や C などの細字で表す. テンソルのスカラー積は $\varepsilon : \varepsilon$ で表す. 二階テンソルのノルムは特に断らない場合は Frobenius ノルムとし, $\varepsilon \in \mathbb{R}^{\dim \times \dim}$ に対して

$$\|\varepsilon\| = (\varepsilon : \varepsilon)^{\frac{1}{2}} = \left(\sum_{i=1}^{\dim} \sum_{j=1}^{\dim} (\varepsilon_{ij})^2 \right)^{\frac{1}{2}}$$

で定義する. ベクトルのテンソル積を \otimes で表し, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{\dim}$ に対して

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_{\dim} \\ u_2 v_1 & u_2 v_2 & & u_2 v_{\dim} \\ \vdots & & \ddots & \vdots \\ u_{\dim} v_1 & u_{\dim} v_2 & \cdots & u_{\dim} v_{\dim} \end{pmatrix}$$

と定義する. 記号 $\text{diag}(o)$ は次のような対角行列を表すこととする:

$$\text{diag}(u_1, u_2, \dots, u_n) = \begin{pmatrix} u_1 & & & \\ & u_2 & & \\ & & \ddots & \\ & & & u_n \end{pmatrix}.$$

$\langle o, o \rangle$ は内積を表し, 特に断らない場合は $\Omega \times [0, T]$ で定義される $\mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t) \in \mathbb{R}^{\dim}$ に対して

$$\langle \mathbf{u}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t) \rangle = \int_0^T \int_{\Omega} \mathbf{u}^{\top} \mathbf{v} d\mathbf{x} dt$$

と定義する. 関数とその変数をセミコロンを用いて $f(x; y)$ と書いた場合, f は x を変数とする関数であるが, f は何らかの形で y と関係があることを表す.

場の変数と設計変数は本論文を通してそれぞれ \mathbf{u} および ϕ とする. 設計変数 ϕ は空間 Ω 上で定義される. ただし時間と空間に関して離散化された場合, 時間と空間を離散化した点の集合 $\hat{\Omega} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$ に対して,

$$\mathbf{u}_k = (u_x(\mathbf{x}_0, t_k), u_y(\mathbf{x}_0, t_k), u_z(\mathbf{x}_0, t_k), u_x(\mathbf{x}_1, t_k), \dots, u_z(\mathbf{x}_N, t_k))^{\top}, \quad (1.1a)$$

$$\phi_k = (\phi(\mathbf{x}_0, t_k), \phi(\mathbf{x}_1, t_k), \dots, \phi(\mathbf{x}_N, t_k))^{\top} \quad (1.1b)$$

と定義し, ベクトルとして扱う. ここで $u_x(\mathbf{x}_\ell, t_k), u_y(\mathbf{x}_\ell, t_k), u_z(\mathbf{x}_\ell, t_k)$ はそれぞれ時刻 t_k の点 \mathbf{x}_ℓ における変数の x, y, z 成分である. ただし時間と空間に関して離散化された場合に, $\phi(\mathbf{x}_i, t_k)$ や $\phi_k(\mathbf{x}_i)$ と表記した場合は, ベクトル ϕ_k の要素のうち, 点 \mathbf{x}_i に相当する要素を示すこととする. $\mathbf{u}(\mathbf{x}_i, t_k)$ や $\mathbf{u}_k(\mathbf{x}_i)$ についても同様に, ベクトル \mathbf{u}_k の要素のうち点 \mathbf{x}_i に相当する x, y, z 方向の要素を取り出して縦に並べたベクトルを表す. また, ベクトル \mathbf{v} について $(\mathbf{v})_j$ と表記した場合は, ベクトル \mathbf{v} の j 番目の要素を表す. さらに, 汎関数 $E(\mathbf{u}(\phi_k(\mathbf{x})), \phi_k(\mathbf{x}))$ の $\phi_k(\mathbf{x})$ による微分に関して, 連鎖側を考慮して

$$\nabla_{\phi_k(\mathbf{x})} E = \frac{\partial E}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \phi_k(\mathbf{x})} + \frac{\partial E}{\partial \phi_k(\mathbf{x})}$$

と定義し，これを縦に並べたベクトルを

$$\nabla_{\phi_k} E = \begin{pmatrix} \nabla_{\phi_k(\mathbf{x}_1)} \\ \nabla_{\phi_k(\mathbf{x}_2)} \\ \vdots \\ \nabla_{\phi_k(\mathbf{x}_N)} \end{pmatrix} E$$

と定義する．

その他，必要に応じて本文中で記号および演算子を定義する．

第 2 章

準備：格子 Boltzmann 法の定式化

本章では第 4 章から第 7 章で用いる格子 Boltzmann 法 (Lattice Boltzmann Method; LBM) を定式化する。格子 Boltzmann 法 (Lattice Boltzmann Method; LBM) は流れ場の数値解法の一つであり、流体の数値計算手法としては比較的新しい手法である [73]。LBM には格子モデルと衝突モデルにいくつかのモデルが提案されているが、本節ではその中でもそれぞれ二種類を定式化する。なお、入口および出口の境界条件と、一部の定行列については付録 A に示す。

2.1 格子 Boltzmann 法

LBM は均一な格子の上で Boltzmann 方程式を解く手法である。はじめは格子気体法 (lattice gas cellular automaton) と呼ばれる手法が希薄流体の数値解法として定式化され、利用されていた [13, 15]。その後、格子気体法をもとに格子モデルと衝突モデルが改良され、LBM として定式化され一般的な流体の数値解法として利用されている [41, 46, 66]。LBM による流れ場の解析結果は他の広く使われる CFD 手法と比較され、その妥当性が検証されている [9, 18, 23, 25, 28, 44, 45, 65]。

LBM は初期のころはマッハ数 $\text{Mach} < 0.3$ 以下の非圧縮流れを解くための手法であったが、その後 $\text{Mach} < 0.8$ の圧縮性流れにも適用対象が広がり [60]、現在では $\text{Mach} > 0.8$ の超音速流れを解くことができる手法 [67, 76] も提案されている。また、素朴な LBM は規則的な格子上で計算する必要があったため、格子に粗密がつけられず、計算量の増大を招いていたが、格子に粗密を付ける方法 [72] や、不規則な格子上で LBM [53] 等も提案され、計算時間が短縮されている。

格子気体法は、その英語名にあるようにセルオートマトン [78] と関係が深く、格子気体法より発展した LBM も同様にセルオートマトンと関係がある。素朴なセルオートマトンでは、各セルで 0-1 の二値関数が定義され、その値は隣接するセルの値をもとに一定のルールに従って変化する。LBM は、セルオートマトンにおいて各セルの関数を実数ベクトル値関数とし、関数値の変化のルールを Boltzmann 分布にしたがって定義して移流項を加えたものと解釈することができる [31, 73]。

本章は以下のように構成される．第 2.2 節では基礎方程式を定式化し，二種類の格子モデルを導入する．次に第 2.3 節では二種類の衝突モデルを定式化する．さらに第 2.4 節では多孔質媒体中の流れに対する LBM の解法を定式化する．第 2.5 節では非滑り境界条件として bounce-back 条件を定式化する．Bounce-back 条件以外の境界条件については第 A.1 節から第 A.4 節で述べる．

2.2 格子 Boltzmann 法の基礎方程式

LBM では，流れは粒子の挙動としてモデル化され，粒子は格子点上で衝突と移流を繰り返しながら粒子の分布が時間発展する．粒子の速度も離散化されており，ある格子点上の粒子は，決められた時間幅 δt の間に自分自身か隣接する格子点に移動することしかできない．各格子点において，格子の接続方向 i に移動する粒子を考える．このとき i 方向に動く粒子は密度 $f_i \in \mathbb{R}$ と速度 $\mathbf{c}_i \in \mathbb{R}^{\dim}$ を持つ．ただし \dim は考えている空間の次元である． i の数は格子の隣接関係によって決まる．ここでは n 個の点と隣接関係にあるとする．このとき f_i は，動かない粒子の方向，すなわち自分自身への隣接関係を含めて $n + 1$ 個存在する．従って i は自分自身 ($i = 0$) を含めて $\{0, 1, \dots, n\}$ の要素である． f_i を縦に並べたベクトル $\mathbf{f} = (f_0, \dots, f_n)^\top$ は密度分布関数と呼ばれる．格子の隣接関係と速度 \mathbf{c}_i は，使用する格子モデルによって異なる．格子モデルはいくつか提案されているが，本論文では二次元流れには FHP 格子モデル [66, 73] を使用し，三次元流れには三次元 19 速度 (D3Q19) モデル [66] を使用する．

まず，FHP 格子の格子形状を図 2.1 に示す．なお， δx は格子点間の距離を表す．FHP 格子は二次元平面上の六角形状の格子で，各点は六つの点と隣接関係にある．隣接する格子点へのベクトル \mathbf{e}_i は次式で定義される：

$$\mathbf{e}_i = \begin{cases} (0, 0)^\top & \text{for } i = 0, \\ (\pm\delta x, 0)^\top & \text{for } i = 1, 4, \\ (\pm\frac{1}{2}\delta x, \pm\frac{\sqrt{3}}{2}\delta x)^\top & \text{for } i = 2, 5, \\ (\mp\frac{1}{2}\delta x, \pm\frac{\sqrt{3}}{2}\delta x)^\top & \text{for } i = 3, 6. \end{cases} \quad (2.1)$$

さらに $\mathbf{c}_i = \frac{\mathbf{e}_i}{\delta t}$ と定義される．また， c_s は格子点間の距離 δx と時間ステップ幅 δt を用いて $c_s = \delta x / \delta t$ で定義される定数である．また，各隣接方向には重み ω_i が定義される． ω_i は FHP 格子では

$$\omega_i = \begin{cases} 1/2 & \text{for } i = 0, \\ 1/12 & \text{for } i = 1, 2, \dots, 6 \end{cases}$$

である． ω_i は LBM の具体的な方程式を書き下す時に使用する．

次に D3Q19 モデルの格子形状を図 2.2 に示す．各点は三次元空間上の 18 個の点と隣接関

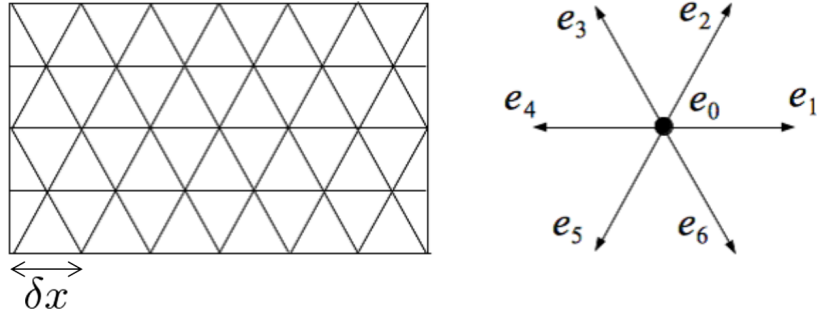


図 2.1. FHP 格子の隣接関係と速度ベクトル.

係にあり, e_i は次式で定義される :

$$e_i = \begin{cases} (0, 0, 0)^\top & \text{for } i = 0, \\ (\pm\delta x, 0, 0)^\top & \text{for } i = 1, 2, \\ (0, \pm\delta x, 0)^\top & \text{for } i = 3, 4, \\ (0, 0, \pm\delta x)^\top & \text{for } i = 5, 6, \\ (\pm\delta x, \pm\delta x, 0)^\top & \text{for } 7 \leq i \leq 10, \\ (\pm\delta x, 0, \pm\delta x)^\top & \text{for } 11 \leq i \leq 14, \\ (0, \pm\delta x, \pm\delta x)^\top & \text{for } 15 \leq i \leq 18. \end{cases} \quad (2.2)$$

重み ω_i は

$$\omega_i = \begin{cases} 1/3 & \text{for } i = 0, \\ 1/18 & \text{for } i = 1, 2, \dots, 6, \\ 1/36 & \text{for } i = 7, 8, \dots, 18 \end{cases}$$

により定義される.

f_i は格子上で移動と衝突を繰り返すが, この操作は移流操作 $\mathcal{F}^{\text{prop}}$ と衝突操作 \mathcal{F}^{col} を用いて次のように表現することができる :

$$\mathbf{f}(t_{k+1}) = \mathcal{F}(\mathbf{f}(t_k)) = \mathcal{F}^{\text{prop}} \circ \mathcal{F}^{\text{col}}(\mathbf{f}(t_k)).$$

ただし $t_k = k\delta t$ とし, 記号 \circ は関数の合成を表す. 時刻 t_k における分布関数に衝突操作と移流操作を施すことで, 時刻 t_{k+1} の分布関数が得られる. 表記を簡易にするため, 衝突操作後の分布関数として $\mathbf{f}^{\text{col}} = \mathcal{F}^{\text{col}}(\mathbf{f})$ という表記を使用する. 移流操作はどの格子モデルでも, 隣接関係の違いを除いて同じ操作である. すなわち, ある格子点の i 方向の粒子を, 時刻が δt だけ進むのにしたがって i 方向に隣接する格子に移動させると考えればよい. 一方で衝突操作は複数のモデルが提案されており, 本論文でも BGK モデルと MRT モデルという二種類のモデルを使用する. 両モデルについては 2.3 節で説明する.

以上の手続きにしたがって分布関数 \mathbf{f} が時間発展するが, 流れ場のマクロな変数である流

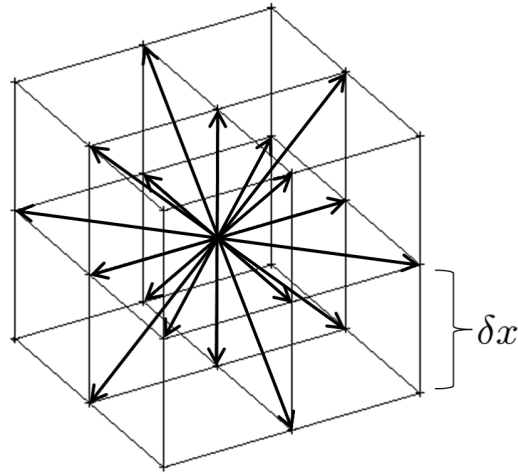


図 2.2. D3Q19 モデル.

速 \mathbf{u} と密度 ρ は, \mathbf{f} の関数として次式から得られる :

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t), \quad (2.3a)$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \sum_i f_i(\mathbf{x}, t) \mathbf{c}_i. \quad (2.3b)$$

以降, 表記を簡易にするために, 一連の LBM の手続きを経て得られる流速を $\mathbf{u}_{\text{LBM}}(\mathbf{x}, t_k)$ で表す. この表記では \mathbf{f} が陽に現れないが, 実際には $\mathbf{u}_{\text{LBM}}(\mathbf{x}, t_k)$ を得るために \mathbf{f} の時間発展の計算を行う. 最後に, 格子の単位系と SI 単位系は dx [m], dt [s], ρ [$\text{kg} \cdot \text{m}^{-3}$] により変換される.

2.3 衝突モデル

衝突操作は複数のモデルが提案されている. 大別すると単一緩和時間 LBM (Single Relaxation Time LBM; SRT-LBM) と, 多緩和時間 LBM (Multiple Relaxation Time LBM; MRT-LBM) に分けられる. 本節では SRT-LBM の一種である格子 BGK モデルと MRT-LBM を定式化する. 本論文では前述の FHP 格子モデルは BGK モデルと, D3Q19 モデルは MRT-LBM と組み合わせて用いる.

2.3.1 BGK モデル

BGK モデルは Bhatnager et al. [6] によって提案されたモデルで, LBM の歴史の中でも最初に提案されたモデルである. 当初は希薄流体の数値計算に利用されたが, その後一般の Newton 流体の数値計算にも利用されるようになった. BGK モデルに関しては [66, 73] が詳

しい。BGK モデルの衝突操作と移流操作は次式で定義される：

$$\text{Collision : } f_i^{\text{col}}(\mathbf{x}, t_k) = f_i(\mathbf{x}, t_k) - \frac{f_i(\mathbf{x}, t_k) - f_i^{\text{eq}}(\mathbf{x}, t_k)}{\tau} + \delta t F_i(\mathbf{x}, t_k), \quad (2.4a)$$

$$\text{Propagate : } f_i(\mathbf{x} + \mathbf{e}_i, t_{k+1}) = f_i^{\text{col}}(\mathbf{x}, t_k). \quad (2.4b)$$

ここで τ は緩和時間係数と呼ばれる定数であり、レイノルズ数 Re と次のような関係がある：

$$\text{Re} = \frac{2L\bar{U}}{(2\tau - 1)c_s^2\delta t}.$$

ただし L は代表長さ、 \bar{U} は代表流速を表す。(2.4) において、 f^{eq} は平衡分布関数と呼ばれ、Boltzmann 分布から導かれる関数である。また F_i は外力に起因する項である。外力を $\hat{\mathbf{F}} \in \mathbb{R}^{\text{dim}}$ とするとき、これらはそれぞれ次式で定義される：

$$f_i^{\text{eq}} = \omega_i \rho \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{u} \otimes \mathbf{u}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \right], \quad (2.5)$$

$$F_i = \omega_i \rho \left(1 - \frac{1}{2\tau} \right) \left[\frac{\mathbf{c}_i \cdot \hat{\mathbf{F}}}{c_s^2} + \frac{(\mathbf{u} \otimes \hat{\mathbf{F}}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \right]. \quad (2.6)$$

2.3.2 MRT-LBM

本節では MRT-LBM [12] を定式化する。MRT-LBM では、分布関数 f をいったん粒子の持つモーメントの空間に写像し、モーメントの空間で衝突を模擬する。まず \mathbf{f} は次式によりモーメント $\mathbf{m} \in \mathbb{R}^n$ に変換される：

$$\mathbf{m} = M\mathbf{f}.$$

$M \in \mathbb{R}^{n \times n}$ は定行列であるが、使用する格子モデルによって係数が異なる。本論文で使用する Q3D19 モデルにおける定義を (A.3) に示す。

次にモーメントの空間で衝突を模擬する。BGK モデルでは f^{eq} を用いたが、モーメント空間では \mathbf{m}^{eq} を用いる。その後、再び M^{-1} により元の分布関数の空間に戻し、BGK モデルと同様に移流操作を行う。この操作は次式のように記述される：

$$\text{Collision step } \mathbf{f}^{\text{col}}(\mathbf{x}, t_k) = \mathbf{f}(\mathbf{x}, t_k) - M^{-1}S(M\mathbf{f}(\mathbf{x}, t_k) - \mathbf{m}^{\text{eq}}(\mathbf{x}, t_k)), \quad (2.7a)$$

$$\text{Propagate step } f_i(\mathbf{x}, t_{k+1}) = f_i^{\text{col}}(\mathbf{x} - \mathbf{e}_i, t_k). \quad (2.7b)$$

モーメント空間における平衡分布関数 m^{eq} は次式で定義される：

$$\begin{aligned}
m_0^{\text{eq}} &= 0, & m_1^{\text{eq}} &= -11\rho + \frac{19}{\rho} \mathbf{u}^\top \mathbf{u}, & m_2^{\text{eq}} &= \hat{\omega}_\varepsilon \rho + \frac{\hat{\omega}_{\varepsilon j}}{\rho_0} \mathbf{u}^\top \mathbf{u}, \\
m_3^{\text{eq}} &= 0, & m_5^{\text{eq}} &= 0, & m_7^{\text{eq}} &= 0, \\
m_4^{\text{eq}} &= -\frac{2}{3} u_x, & m_6^{\text{eq}} &= -\frac{2}{3} u_y, & m_8^{\text{eq}} &= -\frac{2}{3} u_z, \\
m_9^{\text{eq}} &= \frac{1}{\rho_0} (2u_x^2 - u_y^2 - u_z^2), & m_{10}^{\text{eq}} &= \hat{\omega}_{xx} m_9^{\text{eq}}, \\
m_{11}^{\text{eq}} &= \frac{1}{\rho_0} (u_y^2 - u_z^2), & m_{12}^{\text{eq}} &= \hat{\omega}_{xx} m_{11}^{\text{eq}}, \\
m_{13}^{\text{eq}} &= \frac{1}{\rho_0} u_x u_y, & m_{14}^{\text{eq}} &= \frac{1}{\rho_0} u_y u_z, & m_{15}^{\text{eq}} &= \frac{1}{\rho_0} u_z u_x, \\
m_{16}^{\text{eq}} &= m_{17}^{\text{eq}} = m_{18}^{\text{eq}} = 0.
\end{aligned}$$

ただし $\hat{\omega}_\varepsilon = 3$, $\hat{\omega}_{xx} = -\frac{1}{2}$, $\hat{\omega}_{\varepsilon j} = -\frac{11}{2}$ とする．また S は緩和行列と呼ばれ、次式で定義される：

$$S = \text{diag}(0, s_1, s_2, 0, s_4, 0, s_4, 0, s_4, s_9, s_2, s_9, s_2, s_9, s_9, s_9, s_{16}, s_{16}, s_{16}).$$

ここで S の要素がモーメント m のそれぞれの要素に対する緩和係数である．それぞれの値は [12] より、次の値を用いる．

$$s_1 = 1.19, \quad s_2 = s_{10} = 1.4, \quad s_4 = 1.2, \quad s_{16} = 1.98.$$

s_9 は動粘性係数 ν に基づいて

$$\nu = \frac{1}{3} \left(\frac{1}{s_9} - \frac{1}{2} \right) \frac{\delta x^2}{\delta t}$$

を満たすように定義される．なお、速度と圧力の境界条件については付録 A で定式化する．

2.4 多孔質媒体中の流れの BGK モデルによる解法

第 4 章と第 5 章の流れ場のトポロジー最適化では、多孔質媒体モデルを使用する．このモデルでは、領域は多孔質媒体で充たされていると考え、その空隙率を設計変数として最適化を行う．本節では、多孔質媒体中の流れ場を LBM により解く手法について述べる．多孔質媒体の空隙率を設計変数とし、 ϕ で表す． ϕ は Ω の全領域で定義される 0 以上 1 以下の実数である． $\phi = 1$ である領域は完全に空洞であり、領域が流体で充たされていると考えることができる．逆に $\phi = 0$ である領域は完全に固体で充たされていると考えられる．これら以外の、 $0 < \phi < 1$ である領域は、 ϕ が小さいほど流体の占める割合が大きく、逆に ϕ が大きいほど固体の占める割合が大きい．このような多孔質媒体中を流れる流れは、流速 \mathbf{u} と空隙率 ϕ に依存する抵抗力 $\hat{\mathbf{F}}$ を受ける．Brinkman モデル [23] によると、この抵抗力は次式で定義される：

$$\hat{\mathbf{F}} = -\alpha(\phi) \mathbf{u}, \quad (2.8)$$

$$\alpha(\phi) = \alpha_0 \frac{1 - \phi}{\phi + \frac{1}{2}}. \quad (2.9)$$

一般的な Brinkmann モデルでは (2.9) で $\phi = 0$ とした式を使う。しかしこの場合、 $\phi = 0$ の極限で (2.9) は発散する。そこで、正の定数 $\underline{\phi}$ を用いて発散しないようにしている。本論文では $\underline{\phi} = 0.001$ を用いる。通常、多孔質媒体中の流れを数値計算で求める場合、 $\phi = 0$ は領域が完全に固体であるから数値計算をする必要がなく、実用上は ϕ が零に近い値を取ることはないため、このような不都合は生じない。いま $\alpha(1) = 0$, $\alpha(0) = \alpha_0$ である。連続体においては α_0 が十分大きければ、流れは十分に大きな抵抗力を受け、 $\phi = 0$ の時に $\mathbf{u} = 0$ になる。

平衡分布関数と外力項は Boltzmann 分布から導かれ、Guo and Zhao [23] によると次式で定義される：

$$f_i^{\text{eq}} = \omega_i \rho \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{u} \otimes \mathbf{u}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2(\phi + \underline{\phi})c_s^4} \right], \quad (2.10)$$

$$\begin{aligned} F_i &= \omega_i \rho \left(1 - \frac{1}{2\tau} \right) \left[\frac{\mathbf{c}_i \cdot \hat{\mathbf{F}}}{c_s^2} + \frac{(\mathbf{u} \otimes \hat{\mathbf{F}}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2(\phi + \underline{\phi})c_s^4} \right] \\ &= -\alpha(\phi) \omega_i \rho \left(1 - \frac{1}{2\tau} \right) \left[\frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{u} \otimes \mathbf{u}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2(\phi + \underline{\phi})c_s^4} \right]. \end{aligned} \quad (2.11)$$

$\underline{\phi}$ は (2.9) 同様、 $\phi = 0$ の極限で値が発散しないようにするために用いる。さらに、流速に次の修正を加える：

$$\rho(\mathbf{x}, t) = \sum_i f_i^{\text{col}}(\mathbf{x}, t), \quad (2.12a)$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{\hat{\mathbf{u}}}{1 + 0.5\alpha}, \quad \hat{\mathbf{u}} = \frac{1}{\rho} \sum_i f_i(\mathbf{x}, t) \mathbf{c}_i. \quad (2.12b)$$

ただし、 $\phi = 0$ の場合は後述するように $\mathbf{u} = 0$ であり、 $\phi = 1$ の場合は $\alpha = 0$ であるから、これらの場合には実質的に $\mathbf{u} = \hat{\mathbf{u}}$ になる。またこの修正は物理的な観点から加えられるが [23]、流れ場のトポロジー最適化で重要な事は ϕ を 0 から 1 に変えた時に \mathbf{u} が連続に変化することであり、多孔質媒体モデルは便宜的に利用しているに過ぎない。

ここで α_0 の適切な値を導く。 α_0 は、 $\alpha = 0$ の場合に衝突操作の後で $\|\mathbf{u}\| = 0$ となるように与えることが望ましい。(2.4a) において、外力項が $F_i = 0$ であれば、衝突操作の前後で速度は変わらない。これは、定義より $\sum_i f_i \mathbf{c}_i = \rho \mathbf{u}$ 、また (2.10) より $\sum_i f_i^{\text{eq}} \mathbf{c}_i = \mathbf{u}$ であることから明らかである。ここで (2.4a) より

$$\begin{aligned} \sum_i f_i^{\text{col}} \mathbf{c}_i &= \rho \mathbf{u}(t_k) - \sum_i F_i \mathbf{e}_i \\ &= \rho \mathbf{u}(t_k) - \alpha \left(1 - \frac{1}{2\tau} \right) \mathbf{u}(t_k) \delta t \end{aligned}$$

であるから、左辺を零にするためには $\alpha_0 = \frac{2\tau\rho}{(2\tau-1)\delta t}$ とすれば良い。

2.5 Bounce-back 条件

流れ場の挙動は、特に流速が早い場合、境界条件に大きく左右される。固体壁との境界上で流速が零になるという非滑り条件は、流体力学における基本的な仮定の一つで、数値計算でも頻繁に用いられる。第 6 節および第 7 節においても、流体と固体の境界面に非滑り境界条件を適用する。LBM における非滑り条件は、bounce-back 条件が広く用いられており [17, 81, 88]、本論文でもこれを用いる。本節では bounce-back 条件を定式化する。

まず、領域 Ω が流体領域 Ω_f と固体領域 Ω_s に分けられているとする。二つの隣接する格子点について、一方が Ω_f に属し、もう一方が Ω_s に属しているとする。このとき、両者の間に固体壁が存在する。Bounce-back 条件では隣接する格子の midpoint に固体壁が存在する様子を模擬する。 $\mathbf{x} \in \Omega_f$, $\mathbf{x} + \mathbf{e}_i \in \Omega_s$ と仮定する。このとき、bounce-back 条件は移流操作を次の式で置き換えることで適用される：

$$f_i(\mathbf{x}, t_{k+1}) = f_{-i}^{\text{col}}(\mathbf{x}, t_k). \quad (2.13)$$

ここで下付き添え字 $-i$ は i の逆方向を表す。すなわち $\mathbf{e}_{-i} = -\mathbf{e}_i$ である。(2.13) は、隣接する格子点が固体であれば、その方向に進んでいた粒子を逆向きに跳ね返すことに相当し、その様子から bounce-back と呼ばれる。関数 χ を

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_f, \\ 0 & \text{if } \mathbf{x} \in \Omega_s \end{cases}$$

で定義する。(2.13) と χ を用いることで、bounce-back 条件付の MRT-LBM は次の式で記述される：

$$\text{Collision step} \quad \mathbf{f}^{\text{col}}(\mathbf{x}, t_k) = \mathbf{f}(\mathbf{x}, t_k) - M^{-1}S(M\mathbf{f}(\mathbf{x}, t_k) - \mathbf{m}^{\text{eq}}(\mathbf{x}, t_k)), \quad (2.14a)$$

$$\text{Propagate step} \quad f_i(\mathbf{x}, t_{k+1}) = \begin{cases} f_i^{\text{col}}(\mathbf{x} - \mathbf{e}_i, t_k) & \text{if } \chi(\mathbf{x} - \mathbf{e}_i, t_k) = 1, \\ f_{-i}^{\text{col}}(\mathbf{x}, t_k) & \text{if } \chi(\mathbf{x} - \mathbf{e}_i, t_k) = 0. \end{cases} \quad (2.14b)$$

(2.14) は流体領域 Ω_f における場の変数の計算のための式だが、式の上では領域全体 $\mathbf{x} \in \Omega = \Omega_f \cup \Omega_s$ で矛盾なく定義することができる。特に、このように定義域を拡張したとしても、 $\mathbf{x} \in \Omega_f$ の格子点における \mathbf{f} は、拡張しない場合の結果と同じである。ただし、 Ω_s の固体壁近傍、すなわち Ω_s に属する格子点 \mathbf{x} でその隣接格子点の少なくとも一つが Ω_f に属しているような \mathbf{x} については、(2.14) と (2.3) から得られる \mathbf{u} が一般に零でなく、力学的な観点からは不適切である。本論文では、(2.14) を領域 Ω 全体で定義し、前述の力学的に不適切な部分については、第 6.3 節で適切に処理する。

第 3 章

iSA 法の提案とその応用

本章では微分方程式を制約条件とする最適化問題に対して iSA 法を提案する。まず最適化問題を定義し、iSA 法の考え方を示して手法を構築したのち、iSA 法の典型的なアルゴリズムを提案する。さらに解の存在について述べる。また二次元弾性体のトポロジー最適化問題を iSA 法を用いて解き、その妥当性について述べる。

3.1 最適化問題の定義

流れ場は Navier–Stokes 方程式などの流体の方程式に従う。流れ場を最適化する場合、設計変数を変化させたときに流れ場の方程式が変化するような数理モデルを作成する。この状況は次のように一般化される。まず、パラメータ ϕ に依存する偏微分方程式を考え、場の変数 u がそれに従うものとする。本節では簡単のために u はスカラーとする。この方程式は

$$\mathcal{A}(u, \phi) = 0 \quad (3.1)$$

で表される。ただし \mathcal{A} は、 u の次元と等しい数の方程式が連立した偏微分方程式とする。 ϕ は数理モデルによって異なるが、例えばスプライン曲線の制御点の位置や、トポロジー最適化における密度関数などの、形状を表現するパラメータと考えればよい。この偏微分方程式のもとで ϕ を変数として関数 E を最小化する最適化問題は、次のように書くことができる。

$$(P) \quad \text{find} \quad \phi \quad (3.2a)$$

$$\text{min.} \quad E(u, \phi), \quad (3.2b)$$

$$\text{s.t.} \quad \mathcal{A}(u, \phi) = 0, \quad (3.2c)$$

$$\mathcal{G}(\phi) \leq 0. \quad (3.2d)$$

ただし $\mathcal{G}(\phi) \leq 0$ は ϕ に関する制約であり、本節では簡単のために制約は一つだけとする。任意の等式制約は不等式制約の組み合わせで表現できるため、本章では不等式制約のみを扱う。(3.2c) が場の変数の方程式である。最適化問題 (P) の Lagrange 関数は

$$\mathcal{L} = E(u, \phi) - \int_{\Omega} \mathcal{A}(u, \phi) v d\mathbf{x} - \int_{\Omega} \mathcal{G}(\phi) \lambda d\mathbf{x} \quad (3.3)$$

である。ただし v , λ は Lagrange の未定乗数であり、どちらも領域 Ω 上で定義され、時刻に依存する。したがって最適性の一次の必要条件である Karush–Kuhn–Tucker 条件 (KKT 条件) は形式的に

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial E}{\partial \phi} - \frac{\partial \mathcal{A}}{\partial \phi} v - \frac{\partial \mathcal{G}}{\partial \phi} \lambda = 0, \quad (3.4a)$$

$$\frac{\partial \mathcal{L}}{\partial u} = \frac{\partial E}{\partial u} - \frac{\partial \mathcal{A}}{\partial u} v = 0, \quad (3.4b)$$

$$\mathcal{A} = 0, \quad (3.4c)$$

$$\mathcal{G}\lambda = 0, \mathcal{G} \leq 0, \lambda \leq 0 \quad (3.4d)$$

と書ける。ここで (3.4a) の左辺は、Lagrange 関数の設計変数に対する一階微分であり、感度あるいは設計感度と呼ばれる。

KKT 条件を満たす ϕ と u を探すことが最適化問題の目的である。KKT 条件は最適性の必要条件であり、十分条件ではない。すなわち、KKT 条件を満たす解は極小点か極大点か鞍点のいずれかであり、最小解を与えるとは限らない。ただし最適化問題が凸計画問題である場合には、比較的ゆるい仮定の下で KKT 条件が必用十分条件であることが保証される。しかし、一般の最適化問題は凸とは限らない。特に第 4 章以降で対象とする流れ場の最適化問題は非凸計画問題である。一般の非凸計画問題で大域的な最小解を得ることは簡単ではない。本論文では KKT 条件を満たす解を探索することを目的とし、そのような解を最適解と呼ぶ。既存の文献等では、 u を独立変数と考えず ϕ の従属変数と考えるため、KKT 条件について本論文の定式化とは形式上異なる定式化が行われる。KKT 条件が結果的に等価であることを付録 B で示す。

最適化手法は、KKT 条件の整理の方法によって直接法と随伴変数法に分けられる。以下ではそれぞれの定式化を述べる。

【随伴変数法】

随伴変数法では (3.4b) を満たすような v を求め、その v を用いて $\frac{\partial \mathcal{L}}{\partial \phi}$ を計算する。 $\frac{\partial \mathcal{A}}{\partial u} \neq 0$ かどうかは自明ではないが、(3.4b) より、形式上 $v = \left(\frac{\partial \mathcal{A}}{\partial u}\right)^{-1} \frac{\partial E}{\partial u}$ と書く。これを (3.4a) に代入して v を消去して、

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial E}{\partial \phi} - \frac{\partial \mathcal{A}}{\partial \phi} \left(\frac{\partial \mathcal{A}}{\partial u}\right)^{-1} \frac{\partial E}{\partial u} - \frac{\partial \mathcal{G}}{\partial \phi} \lambda \quad (3.5)$$

を得る。(3.4b) を随伴方程式と呼び、 v を随伴変数と呼ぶ。□

【直接法】

$\mathcal{A}(u, \phi) = 0$ の両辺を ϕ で微分することで

$$\begin{aligned} \frac{\partial \mathcal{A}}{\partial \phi} + \frac{\partial \mathcal{A}}{\partial u} \frac{\partial u}{\partial \phi} &= 0 \\ \Leftrightarrow \frac{\partial \mathcal{A}}{\partial u} \frac{\partial \mathcal{A}}{\partial \phi} &= -\frac{\partial \mathcal{A}}{\partial \phi} \end{aligned}$$

の関係が得られる。これを (3.5) に代入して

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial E}{\partial \phi} + \frac{\partial E}{\partial u} \frac{\partial u}{\partial \phi} - \frac{\partial \mathcal{G}}{\partial \phi} \lambda \quad (3.6)$$

を得る。さらに記号 $\nabla_{\phi} E = \frac{\partial E}{\partial \phi} + \frac{\partial E}{\partial u} \frac{\partial u}{\partial \phi}$ を用いて、KKT 条件は次のように書ける。

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \phi} &= \nabla_{\phi} E - \frac{\partial \mathcal{G}}{\partial \phi} \lambda = 0 \\ \mathcal{A} &= 0, \\ \mathcal{G} \lambda &= 0, \mathcal{G} \leq 0, \lambda \leq 0 \end{aligned}$$

(3.6) を計算するためには $\frac{\partial u}{\partial \phi}$ を明示的に計算する必要がある。□

随伴変数法と直接法の違いは、(3.4b) を解くか $\frac{\partial u}{\partial \phi}$ を計算するかの違いである。どちらが計算量が小さいかは場合によって異なる。どちらの場合も、KKT 条件は (3.4b) を用いて (3.4a) から v を消去し、形式的に (3.5) ないし (3.6) に変形する。以降、特に断らない限りは $\frac{\partial \mathcal{L}}{\partial \phi}$ として (3.5) ないし (3.6) を用いることとする。結局、随伴変数法も直接法も、

$$\frac{\partial \mathcal{L}}{\partial \phi} = 0 \quad (3.7a)$$

$$\mathcal{A} = 0, \quad (3.7b)$$

$$\mathcal{G} \lambda = 0, \mathcal{G} \leq 0, \lambda \leq 0 \quad (3.7c)$$

を解く。(3.5) ないし (3.6) を感度と呼ぶ。

代表的な最適化手法の一つである最急降下法を用いて設計変数を更新する場合、

$$\frac{\partial \phi}{\partial t} = -\kappa \frac{\partial \mathcal{L}}{\partial \phi}$$

に基づいて ϕ を更新し、 $\frac{\partial \phi}{\partial t} = 0$ になった時点で最適化計算を終了する。ただし $\kappa \geq 0$ は更新幅を決める定数である。他の最適化手法においても基本的な考え方は同じである。このとき、既存の多くの設計最適化手法では、ある時点での ϕ に対して、 $\mathcal{A}(u, \phi) = 0$ を満たす u を求め、その u を用いて $\frac{\partial \mathcal{L}}{\partial \phi}$ を計算する。これに対して、SAND [24] や one-shot approach [56, 64] 等の手法では、場の変数と設計変数を並行して更新する。本論文でも同様にして場の変数と設計変数を並行して更新する手法を考える。方程式 $\frac{\partial \mathbf{u}}{\partial t} = \mathcal{A}(\mathbf{u}(t), \phi(t))$ を時間発展させて解いた結果、定常解として (3.7a) が満たされるような連立偏微分方程式 \mathcal{A} を考えると、本手法は、連立微分方程式

$$\frac{\partial \mathbf{u}}{\partial t} = \mathcal{A}(\mathbf{u}(t), \phi(t)), \quad (3.8a)$$

$$\frac{\partial \phi}{\partial t} = -\kappa \frac{\partial \mathcal{L}(\mathbf{u}(t), \phi(t))}{\partial \phi}, \quad (3.8b)$$

$$\mathcal{G}(\phi(t)) \lambda(t) = 0, \mathcal{G}(\phi(t)) \leq 0, \lambda(t) \leq 0. \quad (3.8c)$$

を用いて解釈することができる。(3.8a) と (3.8b) が共に時間に関して定常になった場合、すなわち $\frac{\partial \mathbf{u}}{\partial t} = 0$, $\frac{\partial \phi}{\partial t} = 0$ を満たした場合、(3.7a) と (3.7b) が満たされる。(3.7c) と (3.8c) は

等価な式である。従って、連立方程式 (3.8) を解いて定常解を求めることで、(3.7) の解を求めることができる。本論文で提案する iSA 法も、連立方程式 (3.8) を用いて解を求め、また t に関する離散化を工夫することで計算時間を短縮する。

3.2 定常な場に関する最適化問題に対する iSA 法の定式化

本節では、(3.8) を解くことを離散化して定式化する。第 3.1 節では \mathbf{u} や ϕ は空間 Ω 上で定義される関数であったが、本節では時間と空間について離散化し、(1.1) の定義を用いる。と定義する。ただし $u_x(\mathbf{x}_\ell)$, $u_y(\mathbf{x}_\ell)$, $u_z(\mathbf{x}_\ell)$ はそれぞれ点 \mathbf{x}_ℓ における変数の x , y , z 成分である。離散化された未定乗数 \mathbf{v} , $\boldsymbol{\lambda}$ についても同様に定義する。

まず $\mathcal{A}(\mathbf{u}, \phi) = 0$ を満たす \mathbf{u} を求めることを考える。更新式として記述するために、(3.1) を時間と空間に関して離散化して整理し、形式的に

$$\mathbf{u}_{k+1} = \mathcal{B}(\mathbf{u}_k, \phi) \quad (3.9)$$

と書けるとする。ここで $\mathbf{u}_{k+1} = \mathbf{u}_k$ の場合は、(3.1) と対比して \mathcal{A} と \mathcal{B} の間には $\mathcal{A}(\mathbf{u}, \phi) = \mathcal{B}(\mathbf{u}, \phi) - \mathbf{u}$ という関係が成り立つ。

(3.1) を離散化する際には、 $k-1$ ステップ目での設計変数と場の変数の組 $(\phi_{k-1}, \mathbf{u}_{k-1})$ が与えられた状態で、次の k ステップ目での変数の組 (ϕ_k, \mathbf{u}_k) を計算するように離散化を行う。従って \mathbf{u}_k は \mathbf{u}_{k-1} と ϕ_k を用いて計算するのが自然である。すなわち (3.9) は

$$\mathbf{u}_{k+1} = \mathcal{B}(\mathbf{u}_k, \phi_{k+1}) \quad (3.10)$$

と書き直される。

次に Lagrange 関数 (3.3) と感度 (3.6) を離散化する。 (ϕ_k, \mathbf{u}_k) を決める際は $E(\mathbf{u}_k, \phi_k)$ を最小化するように変数を更新することを考える。従って Lagrange 関数 \mathcal{L}_k は

$$\mathcal{L}_k = E(\mathbf{u}_k, \phi_k) - (\mathbf{u}_k - \mathcal{B}(\mathbf{u}_{k-1}, \phi_k))^\top \mathbf{v}_k - \mathcal{G}(\phi_k)^\top \boldsymbol{\lambda}_k \quad (3.11)$$

と離散化し、これを用いて感度 (3.6) は

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k} = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \phi_k} + \left(\frac{\partial \mathcal{B}(\mathbf{u}_{k-1}, \phi_k)^\top}{\partial \phi_k} \mathbf{v}_k - \frac{\partial \mathcal{G}(\phi_k)^\top}{\partial \phi_k} \boldsymbol{\lambda}_k \right)^\top, \quad (3.12a)$$

$$\frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k} = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \mathbf{u}_k} - \mathbf{v}_k^\top, \quad (3.12b)$$

と離散化する。直接法を用いて KKT 条件 (3.4b) より、求める解では (3.12b) が零になる必要があることを用いて (3.12a) から \mathbf{v}_k を消去すると、

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k} = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \phi_k} + \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \mathbf{u}_k} \frac{\partial \mathcal{B}(\mathbf{u}_{k-1}, \phi_k)}{\partial \phi_k} - \boldsymbol{\lambda}_k^\top \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k} \quad (3.13)$$

を得る。以上の式 (3.9), (3.4d), (3.13) を用いて, (3.8) は以下のように離散化される:

$$\mathbf{u}_{k+1} = \mathcal{B}(\mathbf{u}_k, \phi_{k+1}), \quad (3.14a)$$

$$\phi_{k+1} = \phi_k - \kappa \frac{\partial \mathcal{L}_{k+1}}{\partial \phi_{k+1}}{}^\top, \quad (3.14b)$$

$$\mathcal{G}(\phi_k)^\top \boldsymbol{\lambda}_k = 0, \mathcal{G}(\phi_k) \leq \mathbf{0}, \boldsymbol{\lambda}_k \leq \mathbf{0}. \quad (3.14c)$$

(3.14) を用いて変数の組 (ϕ_k, \mathbf{u}_k) を更新し, 収束解を得るのが iSA 法である。以下では収束解が得られたと仮定し, そのステップを N とする。すなわち

$$\phi_N = \phi_{N-1}, \quad (3.15a)$$

$$\mathbf{u}_N = \mathbf{u}_{N-1} \quad (3.15b)$$

と仮定する。

ここで, (3.14) を解いて得られる解について考察を加える。本節で用いた Lagrange 関数 (3.11) はステップ k に依存しており, 設計変数 ϕ だけでなく場の変数 \mathbf{u} にも依存する関数である。(3.11) を Lagrange 関数とするような最適化問題は, 次のように記述できる:

$$(P_k) \quad \text{find} \quad \phi_k \quad (3.16a)$$

$$\text{min.} \quad E(\mathbf{u}_k, \phi_k), \quad (3.16b)$$

$$\text{s.t.} \quad \mathbf{u}_k = \mathcal{B}(\mathbf{u}_{k-1}, \phi_k), \quad (3.16c)$$

$$\mathcal{G}(\phi_k) \leq \mathbf{0}. \quad (3.16d)$$

ただし (3.14) を用いて得られる変数の組 (ϕ_k, \mathbf{u}_k) は, 一般の k については (P_k) の最適性条件を満たさない。しかし $k = N$ の場合は, (3.14) と (3.15a) より一次の最適性条件を満たす。従って, 本節で提案した iSA 法は各ステップで (P_k) の降下方向を利用して変数を更新し, 定常解になったとき ($k = N$) に (P_N) の最適解が得られる方法とすることができる。

ここで, iSA 法で得られる解と, 最適化問題 (P) の最適解が必ずしも一致しないことを示す。iSA 法で得られる解は

$$\frac{\partial E(\mathbf{u}_N, \phi_N)}{\partial \phi_N} + \frac{\partial E(\mathbf{u}_N, \phi_N)}{\partial \mathbf{u}_N} \frac{\partial \mathcal{B}(\mathbf{u}_{N-1}, \phi_N)}{\partial \phi_N} - \boldsymbol{\lambda}_N^\top \frac{\partial \mathcal{G}(\phi_N)}{\partial \phi_N} = \mathbf{0}^\top \quad (3.17)$$

を満たす。一方で (P) の最適解を $(\tilde{\phi}, \tilde{\mathbf{u}})$ とし, 未定乗数を $\tilde{\boldsymbol{\lambda}}$ とし, 前節における \mathcal{A} の定義を本節に合わせて $\mathcal{A}(\mathbf{u}, \phi) \in \mathbb{R}^3$ とすると, (3.5) より, 最適解 $(\tilde{\phi}, \tilde{\mathbf{u}})$ は

$$\frac{\partial E(\tilde{\mathbf{u}}, \tilde{\phi})}{\partial \tilde{\phi}} - \frac{\partial E(\tilde{\mathbf{u}}, \tilde{\phi})}{\partial \tilde{\mathbf{u}}} \left(\frac{\partial \mathcal{A}(\tilde{\mathbf{u}}, \tilde{\phi})}{\partial \tilde{\mathbf{u}}} \right)^{-1} \frac{\partial \mathcal{A}(\tilde{\mathbf{u}}, \tilde{\phi})}{\partial \tilde{\phi}} - \tilde{\boldsymbol{\lambda}}^\top \frac{\partial \mathcal{G}(\tilde{\phi})}{\partial \tilde{\phi}} = \mathbf{0}^\top \quad (3.18)$$

を満たす。ここで $(\tilde{\phi}, \tilde{\mathbf{u}}) = (\phi_N, \mathbf{u}_N)$ として (3.17) と (3.18) の各辺を引くことで,

$$\frac{\partial \mathcal{B}(\mathbf{u}_{N-1}, \phi_N)}{\partial \phi_N} = - \left(\frac{\partial \mathcal{A}(\mathbf{u}_N, \phi_N)}{\partial \mathbf{u}_N} \right)^{-1} \frac{\partial \mathcal{A}(\mathbf{u}_N, \phi_N)}{\partial \phi_N} \quad (3.19)$$

が得られる。(3.19) が満たされる場合にだけ、iSA 法で得られる解が (P) の最適解であることがいえる。A と B の関係 $\mathcal{A}(\mathbf{u}_N, \phi_N) = \mathcal{B}(\mathbf{u}_N, \phi_N) - \mathbf{u}_N$ を用いると

$$\begin{aligned}\frac{\partial \mathcal{A}(\mathbf{u}_N, \phi_N)}{\partial \phi_N} &= \frac{\partial \mathcal{B}(\mathbf{u}_N, \phi_N)}{\partial \phi_N}, \\ \frac{\partial \mathcal{A}(\mathbf{u}_N, \phi_N)}{\partial \mathbf{u}_N} &= \frac{\partial \mathcal{B}(\mathbf{u}_N, \phi_N)}{\partial \mathbf{u}_N} - I\end{aligned}$$

が得られるから、(3.19) は

$$\frac{\partial \mathcal{B}(\mathbf{u}_N, \phi_N)}{\partial \phi_N} = \left(I - \frac{\partial \mathcal{B}(\mathbf{u}_N, \phi_N)}{\partial \mathbf{u}_N} \right)^{-1} \frac{\partial \mathcal{B}(\mathbf{u}_N, \phi_N)}{\partial \phi_N} \quad (3.20)$$

と等価である。以上のように、本節で構築した iSA 法は (P) を近似的に解く手法と言え、(3.20) を満たすような B を用いる場合には厳密な最適解が求まる。

3.3 時間周期的な場に関する最適化問題に対する iSA 法の定式化

本節では場の変数が明確な周期をもっている場合に対して、最適化問題を定義し、iSA 法を定式化する。これは例えば周期的な流れ場において、一周期分のエネルギー損失を積分することに相当する。第 3.2 節では (P) の近似解を求めていたが、本節で構築する手法は計算する区間について極限を取ることで (P) と等価な解を求めることができることを本節で示す。

目的関数は m ステップ分の区間で和をとり、 $\sum_{j=1}^m E(\mathbf{u}_j, \phi)$ とする。本節の問題設定では、一周期に相当する m ステップ分は、同じ ϕ を用いて \mathbf{u} を更新するのが自然である。従って、 ϕ_k を用いて更新する \mathbf{u} は $\mathbf{u}_1, \dots, \mathbf{u}_m$ までの m 個になる。すると制約条件は

$$\mathbf{u}_\ell = \mathcal{B}(\mathbf{u}_{\ell-1}, \phi) \quad (\ell = 1, \dots, m)$$

を考えればよい。以上より次の最適化問題を考える：

$$(P^m) \quad \text{find} \quad \phi \quad (3.21a)$$

$$\text{min.} \quad \sum_{j=1}^m E(\mathbf{u}_j, \phi), \quad (3.21b)$$

$$\text{s.t.} \quad \mathbf{u}_\ell = \mathcal{B}(\mathbf{u}_{\ell-1}, \phi), \ell = 1, \dots, m, \quad (3.21c)$$

$$\mathcal{G}(\phi) \leq \mathbf{0}. \quad (3.21d)$$

本節では (P^m) を解くことを考える。第 3.2 節で構築した iSA 法では、最適化問題 (P) の近似解を求めた。本節では、扱う場の方程式が非定常であることと随伴変数法を用いることにより、iSA 法でも (P^m) の最適解が求まる。iSA 法における Lagrange 関数 (3.3) は次のように離散化される：

$$\mathcal{L}_k = \sum_{j=km+1}^{km+m} E(\mathbf{u}_j, \phi_k) - \sum_{j=km+1}^{km+m} (\mathbf{u}_j - \mathcal{B}(\mathbf{u}_{j-1}, \phi_k))^\top \mathbf{v}_j - \mathcal{G}(\phi_k)^\top \boldsymbol{\lambda} \quad (3.22)$$

ここで定義した Lagrange 関数 (3.22) は, $k = 0$ とすると (P^m) の Lagrange 関数と一致する. 従って前述したように, iSA 法により (P^m) の最適解が求まる. この Lagrange 関数を用いて, 感度は

$$\frac{\partial \mathcal{L}_k^m}{\partial \phi_k} = \sum_{j=km+1}^{km+m} \frac{\partial E(\mathbf{u}_j, \phi_k)}{\partial \phi_k} + \sum_{j=km+1}^{km+m} \mathbf{v}_j^\top \frac{\partial \mathcal{B}(\mathbf{u}_{j-1}, \phi_k)}{\partial \phi_k} - \boldsymbol{\lambda}_k^\top \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k}, \quad (3.23a)$$

$$\frac{\partial \mathcal{L}_k^m}{\partial \mathbf{u}_j} = \frac{\partial E(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j} - \mathbf{v}_j^\top + \mathbf{v}_{j+1}^\top \frac{\partial \mathcal{B}(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j}, \quad (j = k, k+1, \dots, k+m-1), \quad (3.23b)$$

$$\frac{\partial \mathcal{L}_k^m}{\partial \mathbf{u}_j} = \frac{\partial E(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j} - \mathbf{v}_j^\top, \quad (j = k+m-1), \quad (3.23c)$$

と離散化される. ただし $m \geq 2$ とする. $m = 1$ の場合は (3.23b) を除いた (3.23a), (3.23c) を考えればよいが, これは第 3.3 節で扱った感度 (3.12) と等価であるから, 本節では取り扱わない. また (3.24), (3.25) から随伴方程式

$$\frac{\partial E(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j}^\top - \mathbf{v}_j + \frac{\partial \mathcal{B}(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j} \mathbf{v}_{j+1} = \mathbf{0}, \quad (j = km+1, km+2, \dots, km+m-1), \quad (3.24)$$

$$\frac{\partial E(\mathbf{u}_j, \phi_k)}{\partial \mathbf{u}_j}^\top - \mathbf{v}_j = \mathbf{0}, \quad (j = km+m) \quad (3.25)$$

が導かれる. 随伴変数 \mathbf{v}_j は \mathbf{v}_{km+m} から順に, $\mathbf{v}_{km+m-1}, \dots, \mathbf{v}_{km+1}$ へ時間を遡るように決定される.

(3.14) と同様に, 次の更新式を考える.

$$\mathbf{u}_{km+l} = \mathcal{B}(\mathbf{u}_{km+l-1}, \phi_k), \quad (3.26a)$$

$$\phi_{k+1} = \phi_k - \kappa \frac{\partial \mathcal{L}_{k+1}^m}{\partial \phi_{k+1}}^\top, \quad (3.26b)$$

$$\mathcal{G}(\phi_k)^\top \boldsymbol{\lambda}_k = 0, \mathcal{G}(\phi_k) \leq \mathbf{0}, \boldsymbol{\lambda}_k \leq \mathbf{0}. \quad (3.26c)$$

ただし, (3.24), (3.25) を用いて (3.23a) より \mathbf{v}_j を消去したものを感度 $\frac{\partial \mathcal{L}_{k+1}^m}{\partial \phi_{k+1}}$ として用いる. (3.26) を用いて変数を更新し, ϕ が定常解になり, \mathbf{u} が周期 m の定常解になったとき, $(\phi_N, \{\mathbf{u}_{Nm+1}, \mathbf{u}_{Nm+2}, \dots, \mathbf{u}_{Nm+m}\})$ は (P^m) の最適解である.

第 3.2 節と同様に, iSA 法により得られる解に解釈を加える. 本節で用いた関数 (3.22) を Lagrange 関数とする最適化問題は, 次のように記述できる:

$$(P_k^m) \quad \text{find} \quad \phi_k \quad (3.27a)$$

$$\text{min.} \quad \sum_{j=k}^{k+m-1} E(\mathbf{u}_j, \phi_k), \quad (3.27b)$$

$$\text{s.t.} \quad \mathbf{u}_{k+l} = \mathcal{B}(\mathbf{u}_{k+l-1}, \phi_k), \quad \ell = 0, 1, \dots, m-1, \quad (3.27c)$$

$$\mathcal{G}(\phi_k) \leq \mathbf{0}. \quad (3.27d)$$

(P_k^m) は第 3.2 節と同様にステップ k に依存する．第 3.2 節の場合と異なり，本節の問題では (P_N^m) は (P^m) と等価である．従って，iSA 法で (P^m) の最適解が求まる．

本節のこれまでの定式化では，流れ場が非定常な場合を想定して手法を構築した．この場合は，既存手法と提案手法の解は一致する．ここで定常状態にある場の変数 \mathbf{u} について考えたい場合，つまり (P) の最適解を求めたい場合に，感度 $\frac{\partial \mathcal{L}_{k+1}^m}{\partial \phi_{k+1}}$ の $m \rightarrow \infty$ における極限を取れば，iSA 法の感度が既存手法の感度と等価になることを示す．

まず，既存手法の感度 $\frac{\partial \mathcal{L}^{\text{conv}}}{\partial \phi}$ は (3.4a)，(3.4b) において， $\mathcal{A}(\mathbf{u}, \phi) = \mathbf{u} - \mathcal{B}(\mathbf{u}, \phi)$ とすることで，

$$\frac{\partial \mathcal{L}^{\text{conv}}}{\partial \phi} = \frac{\partial E}{\partial \phi} - \mathbf{v}^\top \frac{\partial \mathcal{B}}{\partial \phi} - \boldsymbol{\lambda}^\top \frac{\partial \mathcal{G}}{\partial \phi} = \mathbf{0}^\top, \quad (3.28a)$$

$$\frac{\partial E}{\partial \mathbf{u}} - \mathbf{v}^\top \left(I - \frac{\partial \mathcal{B}}{\partial \mathbf{u}} \right) = \mathbf{0}^\top, \quad (3.28b)$$

と定式化できる．ただし (3.28a) の \mathbf{v} は，(3.28b) を満たすような \mathbf{v} を用いる．次に， $0 < r < s < m$ を満たす， m に依存しない定数 r, s について，場の変数が時刻 r 以降で定常になると仮定し，さらに随伴変数 \mathbf{v} が時刻 s 以前で定常になると仮定する：

$$\begin{aligned} \mathbf{u}_{k+r+\ell} &= \mathbf{u}_{k+r} \quad (0 \leq \ell < m - r), \\ \mathbf{v}_{k+s-\ell} &= \mathbf{v}_{k+s} \quad (0 \leq \ell \leq s). \end{aligned}$$

ただし，随伴変数は時間を遡るように \mathbf{v}_{m-1} から順に決定されることに注意する．ここで， r と s が m に依存しないことは，所与の ϕ について m を変化させても場の方程式と随伴方程式の形が変わらないため，定常状態になるまでの計算ステップ数が変わらないことから妥当である．このとき感度について $m \rightarrow \infty$ の極限を取ると， s と r が m に依存しないことから，

$$\lim_{m \rightarrow \infty} \frac{1}{m} \frac{\partial \mathcal{L}_{k+1}^m}{\partial \phi_{k+1}} = \frac{s-r}{m} \frac{\partial E(\mathbf{u}_{k+r}, \phi_k)}{\partial \phi_k} + \mathbf{v}_{k+r+1}^\top \frac{\partial \mathcal{B}(\mathbf{u}_{k+r}, \phi_k)}{\partial \phi_{k+r+1}} - \boldsymbol{\lambda}_k^\top \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k} \quad (3.29)$$

となる．ただし仮定より $r \leq j \leq s$ において $\mathbf{u}_j = \mathbf{u}_r$ ， $\mathbf{v}_j = \mathbf{v}_s$ であることを用いた．ただし \mathbf{v}_{r+1} は (3.24) を $j = r + 1$ とした次の式を満たす：

$$\frac{\partial E(\mathbf{u}_{k+r+1}, \phi_k)}{\partial \mathbf{u}_{k+r+1}} - \mathbf{v}_{k+r+1}^\top + \mathbf{v}_{r+2}^\top \frac{\partial \mathcal{B}(\mathbf{u}_{k+r+1}, \phi_k)}{\partial \mathbf{u}_{k+r+1}} = \mathbf{0}^\top.$$

ここで仮定より， $\mathbf{v}_{r+1} = \mathbf{v}_{r+2}$ である．従って

$$\frac{\partial E(\mathbf{u}_{k+r+1}, \phi_k)}{\partial \mathbf{u}_{k+r+1}} - \mathbf{v}_{k+r+1}^\top \left(1 - \frac{\partial \mathcal{B}(\mathbf{u}_{k+r+1}, \phi_k)}{\partial \mathbf{u}_{k+r+1}} \right) = \mathbf{0}^\top \quad (3.30)$$

が成り立つ．従って，(3.29)，(3.30) より，

$$\lim_{m \rightarrow \infty} \frac{1}{m} \frac{\partial \mathcal{L}_{k+1}^m}{\partial \phi_{k+1}} = \frac{\partial \mathcal{L}^{\text{conv}}}{\partial \phi}$$

が成り立つ．

$m \rightarrow \infty$ は場の変数を無限に長く計算することを意味しており，十分に長い計算をした上で感度を求めることで，iSA 法の感度は既存手法と一致すると言える．ただし，実際の計算で

は、場の方程式が定常状態になるまでの時間、つまり $0 \leq k \leq r$ の区間を解けば他の部分は計算する必要がない。同様に随伴方程式は $s \leq k \leq m$ の間を解けばよい。このとき、既存手法の計算量と iSA 法の計算量が一致する場合がある。例えば場の方程式として格子ボルツマン方程式を用いる場合には、格子ボルツマン法の更新式が収束するまで \mathbf{u} を計算し、その後随伴変数が収束するまで随伴方程式を解き、最後に感度を計算する。これは前述の iSA 法と同じ計算量である。一方で第 3.2 節で述べたように m が短い場合の iSA 法の感度は既存手法の近似にすぎない。感度を近似することで計算量を減らしていると言うことができる。

3.4 未定乗数の決定方法

本節では (3.14) を、制約条件を明示した形で定式化する。まず制約条件が一つだけである場合を考える。制約条件 $\mathcal{G}(\phi) \leq \mathbf{0}$ は、対になる未定乗数 $\boldsymbol{\lambda}$ と相補性の関係にある。すなわち、

$$\mathcal{G}(\phi) \leq \mathbf{0}, \boldsymbol{\lambda} \leq \mathbf{0}, \mathcal{G}(\phi)^\top \boldsymbol{\lambda} = 0$$

を満たす。制約条件が非有効である場合、つまり $\mathcal{G}(\phi) < \mathbf{0}$ である場合、相補性条件から $\boldsymbol{\lambda} = \mathbf{0}$ になる。このとき、

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k} = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \phi_k} + \left\langle \frac{\partial \mathcal{B}(\mathbf{u}_k, \phi_k)}{\partial \phi_k}, \mathbf{v}_{k+1} \right\rangle$$

が成り立つ。逆に制約条件が有効である場合、つまり $\mathcal{G}(\phi) = \mathbf{0}$ である場合、相補性条件から $\boldsymbol{\lambda} \leq \mathbf{0}$ である。このとき、最適性の必要条件を満たすために

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k} = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \phi_k} + \left\langle \frac{\partial \mathcal{B}(\mathbf{u}_k, \phi_k)}{\partial \phi_k}, \mathbf{v}_{k+1} \right\rangle - \left\langle \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k}, \boldsymbol{\lambda} \right\rangle = 0$$

である必要がある。そこで

$$\left\langle \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k}, \boldsymbol{\lambda}_k \right\rangle = \frac{\partial E(\mathbf{u}_k, \phi_k)}{\partial \phi_k} + \left\langle \frac{\partial \mathcal{B}(\mathbf{u}_k, \phi_k)}{\partial \phi_k}, \mathbf{v}_{k+1} \right\rangle$$

を満たすように $\boldsymbol{\lambda}$ を決定すればよい。すると、(3.31) は次のように書き直される：

$$\mathbf{u}_k = \mathcal{B}(\mathbf{u}_{k-1}, \phi_k), \quad (3.31a)$$

$$\phi_k = \phi_{k-1} - \kappa \nabla_{\phi_k} E - \left\langle \frac{\partial \mathcal{G}(\phi_k)}{\partial \phi_k}, \boldsymbol{\lambda}_k \right\rangle. \quad (3.31b)$$

次に実用上頻出する制約条件について、対になる未定乗数の決定方法を示す。実用上重要な制約条件として ϕ の値域と体積制約があり、それぞれ

$$\begin{aligned} \mathbf{0} &\leq \phi_k, \\ \phi_k &\leq \mathbf{1}, \\ \phi_k^\top \mathbf{1} &= V_k \end{aligned}$$

で定式化される。これらの制約条件に対し、iSA 法において解くべき方程式は次のように書くことができる：

$$\mathbf{u}_k = \mathcal{B}(\mathbf{u}_{k-1}, \phi_k), \quad (3.32a)$$

$$\phi_k = \phi_{k-1} - \kappa \nabla_{\phi_k} E - \lambda_k^1 + \lambda_k^2 - M_k. \quad (3.32b)$$

ここで未定乗数 M_k , λ_k^1 , λ_k^2 は次のように決定する。まず M_k は

$$\left(\phi_{k-1} - \kappa \frac{\partial \mathcal{L}_k}{\partial \phi_k} \right)^\top \mathbf{1} \delta r = V_k \quad (3.33)$$

を満たすように決定する。次に λ_k^1 , λ_k^2 について述べる。任意のベクトル \mathbf{a} の j 番目の要素を $(\mathbf{a})_j$ で表す。このとき λ_k^1 および λ_k^2 は

$$(\lambda_k^1)_j = \begin{cases} \kappa (\nabla_{\phi_k} E)_j - M_k & \text{if } (\phi_k)_j - \kappa (\nabla_{\phi_k} E)_j - M_k \leq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (3.34a)$$

$$(\lambda_k^2)_j = \begin{cases} \kappa (\nabla_{\phi_k} E)_j - M_k & \text{if } (\phi_k)_j - \kappa (\nabla_{\phi_k} E)_j - M_k \geq 1, \\ 0 & \text{otherwise} \end{cases} \quad (3.34b)$$

を満たすように決定する。(3.34a) および (3.34b) は、 $(\phi_{k+1})_j$ が 0 以下または 1 以上にならないよう制限していると考えてもよい。ここで (3.33) は非線形で滑らかでない連立方程式であり、即座に解を求めることは難しい。しかし次のように簡単な反復により M_k を決定することができる。まず $\phi' = \phi_k - \kappa \nabla_{\phi} E$ を計算し、その成分を昇順に並べ替えて ϕ'_i ($\phi'_0 \leq \phi'_1 \leq \dots \leq \phi'_N$) とする。 $X(\phi'_i) = i \delta r$ と定義すると、 $X(\phi'_i)$ は $\phi(\mathbf{x}) < \phi'_i$ であるような \mathbf{x} の総和であり、 i に関して単調非減少である。すると体積は $X(\phi'_i)$ を用いて

$$\begin{aligned} \phi_k^\top \mathbf{1} \delta r &= \int X(\phi') d\phi' \\ &= \sum_{-M_k \leq \phi'_i \leq 1-M_k} (\phi'_{i+1} - \phi'_i) X(\phi'_i) \end{aligned} \quad (3.35)$$

と変形できる。これをもとに M_k を探索すればよい。以上をアルゴリズムとして記述するとアルゴリズム 1 になる。

アルゴリズム 1.

Step 1. $\phi' = \phi_{k-1} - \kappa \nabla_{\phi_k} E$ を計算する。

Step 2. ϕ' を昇順に並べ、順に番号を振る。

Step 3. $(\phi'_{i+1} - \phi'_i) X(\phi'_i)$ を計算する。

Step 4. 適当な ℓ を用いて $M_k = \phi'_\ell$ とし、(3.35) を満たすまで ℓ を増加あるいは減少させる。

(3.35) は M_k に関して単調非減少であるため、アルゴリズム 1 は高々 N 回の反復で停止する。

アルゴリズム 1 により得られる M_k は体積制約を厳密に満たす。一方で近似的に次のアルゴリズムを使用することもできる。

アルゴリズム 2.

Step 1. $\phi' = \phi_{k-1} - \kappa \nabla_{\phi_k} E$ を計算する.

Step 2. $M_k = V_k - \phi'^T \mathbf{1} \delta r$ とする.

アルゴリズム 2 は計算が簡易であるものの, これを用いて計算した M_k は必ずしも体積制約を満たさない. ただし経験的にその誤差は小さく, 実用上問題にならない程度である.

3.5 iSA法の典型的なアルゴリズム

本節では第3.2節, 第3.3節および第3.4節の結果を元に, iSA法を用いる場合の典型的なアルゴリズムを記述する. まず, (3.32) は次のように書き直される.

$$\mathbf{u}_k = \mathcal{B}(\mathbf{u}_{k-1}, \phi_k), \quad (3.36a)$$

$$(\phi_k)_j = \begin{cases} 0 & \text{if } (\phi_{k-1})_j - \kappa (\nabla_{\phi_k} E)_j - M_k \leq 0, \\ (\phi_{k-1})_j - \kappa (\nabla_{\phi_k} E)_j - M_k & \text{if } 0 < (\phi_{k-1})_j - \kappa (\nabla_{\phi_k} E)_j - M_k < 1, \\ 1 & \text{if } (\phi_{k-1})_j - \kappa (\nabla_{\phi_k} E)_j - M_k \geq 1. \end{cases} \quad (3.36b)$$

(3.36) は, \mathcal{B} と $\nabla_{\phi_k} E$ を問題に適した形で定式化することで多くの最適設計に使用できる式である. 第4章以降では, 基本的に (3.36) を用いて最適化を行う.

アルゴリズムの収束の判定は次の式を用いる:

$$\begin{aligned} \mathbf{u} \text{ の収束判定: } & \|\mathbf{u}_k - \mathbf{u}_{k-1}\| \leq \delta_u, \\ \phi \text{ の収束判定: } & \|\phi_k - \phi_{k-1}\| \leq \delta_\phi. \end{aligned}$$

ただしノルムは2-ノルムか最大値ノルムが適当である.

以上をアルゴリズムとして記述すると次のようになる.

アルゴリズム 3.

Step 1. $k = 0$ とし, \mathbf{u}_0 と ϕ_0 を与える.

Step 2. (3.36a) により \mathbf{u}_k を計算する.

Step 3. \mathbf{u}_k と ϕ_k を用いて $\nabla_{\phi_{k+1}} E$ を計算する.

Step 4. アルゴリズム 1 又はアルゴリズム 2 を用いて M_{k+1} を計算する.

Step 5. $(\phi')_j = (\phi_k)_j - \kappa (\nabla_{\phi_{k+1}} E)_j - M_{k+1}$ を計算する.

Step 6. $(\phi')_j < 0$ の場合は $(\phi_{k+1})_j = 0$, $(\phi')_j > 1$ の場合は $(\phi_{k+1})_j = 1$ とし, それ以外の場合は $(\phi_{k+1})_j = (\phi')_j$ とする.

Step 7. 収束条件を満たしていなければ k に $k+1$ を代入して Step 2 に戻る.

Step 8. 停止する.

アルゴリズム 3 の収束条件が満たされるかどうかは自明ではなく, 停止しないことがあり得る. 特に, 第4章以降で取り扱う流れ場の方程式については, Navier–Stokes 方程式の滑らか

な解の大域的な存在性と唯一性が未解決問題であり、停止性は不明である。必ずしも停止しないという意味で、アルゴリズム 3 は健全なアルゴリズムではない。ただし、あるクラスの問題に対してはアルゴリズムが停止することが保証される。第 3.6 節では、解の存在とアルゴリズムの停止に対する十分条件を述べる。

3.6 iSA 法の解の存在とアルゴリズムの停止に関する十分条件

本節ではあるクラスの最適化問題に対して、連立偏微分方程式 (3.8) の解の存在条件を述べ、それをもとにアルゴリズム 3 が停止することを示す。まず (3.1) における \mathcal{A} が、適当な関数 $q \in C^1$ を用いて

$$\mathcal{A}(\mathbf{u}, \phi) = -\nabla_{\mathbf{u}} q(\mathbf{u}, \phi) \quad (3.38)$$

と表せる場合を考える。(3.38) は勾配系と呼ばれる力学系を表す。勾配系は Hamilton 系を含む多くの力学系の問題を包含する系である。(3.38) を用いて (3.8) は次のように書き直すことができる：

$$\frac{\partial}{\partial t} \begin{pmatrix} \mathbf{u} \\ \phi \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{u}} & O \\ O & \kappa \nabla_{\phi} \end{pmatrix} \begin{pmatrix} q \\ \mathcal{L} \end{pmatrix}. \quad (3.39)$$

ただし相補性条件 (3.8c) については、(3.8c) を満たすような未定乗数を第 3.4 節の結果にしたがって与えることとし、記述しない。(3.39) の右辺は q と \mathcal{L} の降下方向である。ここで (3.39) が定常解を持つ必要十分条件は q と \mathcal{L} が \mathbf{u} と ϕ に関して C^1 級で、 $\frac{\partial q}{\partial \mathbf{u}} = 0$, $\frac{\partial \mathcal{L}}{\partial \phi} = 0$ を満たすような \mathbf{u} と ϕ が存在することである。特に \mathcal{L} が下に有界で、実行可能領域がコンパクトであれば局所最適解が存在する。

次に \mathcal{L}_k も \mathcal{B} も ϕ_k と \mathbf{u}_k に関して滑らかで下に有界である場合について、アルゴリズム 3 の停止性について述べる。あるステップ k における ϕ の更新方向は最適化問題 (P_k) における Lagrange 関数 \mathcal{L}_k の降下方向である。 \mathcal{L}_k も \mathcal{B} も滑らかで下に有界であることから、ステップ幅を適切に管理すれば、アルゴリズム 3 は収束する。

3.7 iSA 法の応用可能性

本節では iSA 法を適用できる対象を述べる。まず、解の存在とアルゴリズムの停止性を問わず、(3.8) や (3.36) の形式で記述できる対象について述べる。場の変数が時間発展方程式に従うか、何らかの反復解法で解が得られれば形式上は適用が可能である。このような対象は幅広く、例えば次の方程式が挙げられる。

- 熱伝導方程式： $\mathcal{A}(\mathbf{u}) = -\kappa \Delta \mathbf{u}$.
- Stokes 方程式： $\mathcal{A}(\mathbf{u}) = -\text{grad} p + \nu \Delta \mathbf{u}$.

- 非圧縮性 Navier–Stokes 方程式：

$$\operatorname{div} \mathbf{u} = 0,$$

$$\mathcal{A}(\mathbf{u}) = -\mathbf{u} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} - \operatorname{grad} p + \nu \Delta \mathbf{u}.$$

このうち特に熱伝導方程式と Stokes 方程式は線形方程式であり、(3.19) を満たすような定式化が可能であるため、iSA 法で得られる解が既存手法による最適解と一致することが期待できる。一方で非圧縮性 Navier–Stokes 方程式は (3.19) を満たすような定式化ができず、iSA 法の解と既存手法により得られる最適解は一致しない。ただし、Reynolds 数が小さい場合には非圧縮性 Navier–Stokes 方程式は Stokes 方程式で近似できるので、iSA 法の解が良い近似になると期待できる。また、反復解法で求まるような場の変数とその解法としては、次のようなものがあげられる。なお、時間発展方程式で記述できるものは以下では省略する。

- 弾性体のひずみと応力：線形方程式の CG 法などによる解法。
- 定常非圧縮性 Navier–Stokes 方程式：非線形方程式の CG 法などによる解法。

このうち弾性体のひずみと応力を制約条件とする問題については第 3.7.1 節で例を示す。このように iSA 法で記述できる対象は幅広い。しかしながら、それらが必ずしも収束するとは限らない。時間発展方程式で記述できるようなクラスの問題は、iSA 法で用いる感度が力学的に意味づけ可能な感度になっていることが多いと期待されるため、比較的収束性が良いと思われる。一方で反復解法でのみ記述できるクラスの問題は、iSA 法の感度が力学的に意味づけ不可能な場合があり得るため、比較的収束しにくいと思われる。

3.7.1 構造トポロジー最適化に対する iSA 法アルゴリズム

本節では単純な対象について、実際の定式化を行う。本節の定式化は iSA 法に基づいてなされるが、線形方程式を CG 法で解くため、実装の上では SAND 法 [24] と等価である。第 3.6 節ではアルゴリズムが停止する十分条件として \mathcal{B} が (3.38) を満たすことを挙げた。まず、この十分条件を満たす対象について考察する。(3.38) より、時間発展方程式は

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla_{\mathbf{u}} q \quad (3.40)$$

と書ける。これは、 \mathbf{u} を変数とし、 q を Lagrange 関数とするような最適化問題の、最急降下法による解法とみることができる。このようにして求めることができる場の変数とその支配方程式として、変分原理で記述できる力学系が挙げられる。以下では弾性体の支配方程式を変分原理に基づいて記述し、iSA 法が適用できることを示す。

二次元弾性体におけるコンプライアンス最小化問題を考える。変数を有限要素法で離散化し、要素数を ℓ 、節点自由度の数を n とする。弾性体はフックの法則に従うが、これは荷重 $\mathbf{f} \in \mathbb{R}^n$ と変位 $\mathbf{u} \in \mathbb{R}^n$ 、全体剛性行列 $K \in \mathbb{R}^{n \times n}$ 、要素ごとに定義される密度 $\phi = (\phi_1, \phi_2, \dots, \phi_\ell)^\top$ を用いて

$$\mathbf{f} = K(\phi)\mathbf{u} \quad (3.41)$$

と記述される．ただし $K(\phi)$ は i 番目の要素に対する要素剛性行列 K_i に， i 番目の要素における ϕ_i を乗じたもの $\phi_i K_i$ を足し合わせて作られる．文献 [63] 等では $\phi_i K_i$ ではなく $\phi_i^p K_i$ ($p \geq 1$) とするが，本節では $p = 1$ とした．ここで $K(\phi)$ の正則性を保証するために， ϕ を 0.01 などの十分小さな定数として $\phi_i \in [\phi, 1]$ とする．以下では簡単のため，(3.41) の解法として共役勾配法 (CG 法) を用いる．

目的関数であるコンプライアンスは構造物の柔性を表す指標であり， $\mathbf{f} = K(\phi)\mathbf{u}$ の下で，

$$E(\mathbf{u}; \phi) = \mathbf{f}^\top \mathbf{u} - \frac{1}{2} \mathbf{u}^\top K(\phi) \mathbf{u}$$

で定義される．剛性行列は半正定値行列であるため， $E(\mathbf{u}; \phi)$ は \mathbf{u} に関する凸関数である．以上よりコンプライアンス最小化問題を定式化すると，

$$(P_e) \quad \min. \quad E(\mathbf{u}; \phi), \quad (3.42a)$$

$$\text{s.t.} \quad \mathbf{f} = K(\phi)\mathbf{u}, \quad (3.42b)$$

$$\phi \leq \phi_i \leq 1 \quad (1 \leq i \leq \ell) \quad (3.42c)$$

となる．最適化問題 (P_e) の感度は

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \phi_i} &= \frac{\partial E}{\partial \phi_i} - \gamma_1 - \gamma_2 \\ &= \mathbf{f}^\top \frac{\partial \mathbf{u}}{\partial \phi_i} - \frac{1}{2} \mathbf{u}^\top \frac{\partial K(\phi)}{\partial \phi_i} \mathbf{u} - \mathbf{u}^\top K(\phi) \frac{\partial \mathbf{u}}{\partial \phi_i} - \gamma_1 - \gamma_2 \end{aligned}$$

で定義される． $\frac{\partial \mathbf{u}}{\partial \phi_i}$ に関しては，(3.41) の両辺を ϕ_i で微分して左から K^{-1} を乗じて整理することで

$$\frac{\partial \mathbf{u}}{\partial \phi_i} = -K(\phi)^{-1} \frac{\partial K(\phi)}{\partial \phi_i} \mathbf{u}$$

を得る．従って感度は

$$\frac{\partial \mathcal{L}}{\partial \phi_i} = -\frac{1}{2} \mathbf{u}^\top \frac{\partial K(\phi)}{\partial \phi_i} \mathbf{u} - \gamma_1 - \gamma_2$$

となる．

最適化アルゴリズムは [63] を元にし，連立一次方程式の求解の部分のみ変更した．なお，文献 [63] 等で一般的に用いられるペナルティやフィルタは，本節では使用しない．求解には CG 法を用いた．また既存手法の実装では，単純な計算時間短縮の工夫として，CG 法の初期解として前の最適化ステップにおける CG 法の解を利用する．実装は MATLAB [71] で行った．

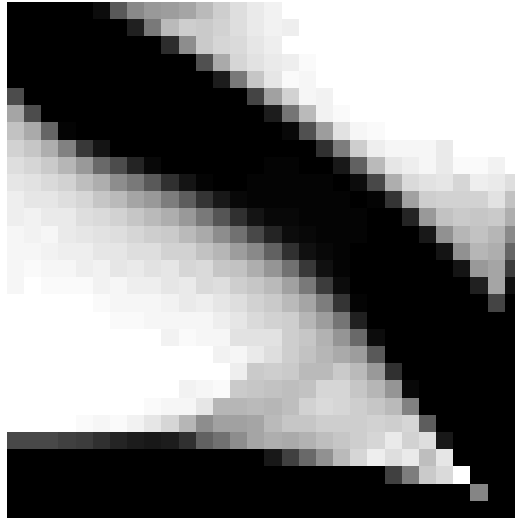
iSA 法と既存手法の反復回数を比較する．最適解と最適値を図 3.1 に示し，最適解に至るまでの最適化の反復回数と CG 法の反復回数を表 3.1 に示す．最適解は iSA 法の結果の方がグレーな領域が多く，目的関数値は iSA 法の方が小さい．また iSA 法が既存手法の約半分の反復回数で停止している．iSA 法では最適化の反復回数と CG 法の合計反復回数が一致している．これは，CG 法を一回反復するたびに最適化の反復を行っているためである．従って，最適化の反復は iSA 法の方が多い．以上のように，本節で扱った問題では iSA 法の方が CG 法

の反復回数が少なく済む。これにより計算時間が短縮されることが期待できる。ただし、線形方程式の場合、行列サイズが特別大きい場合を除いて、方程式の求解の計算コストが小さく、計算時間で言えばiSA法が必ずしも早いとは限らない。

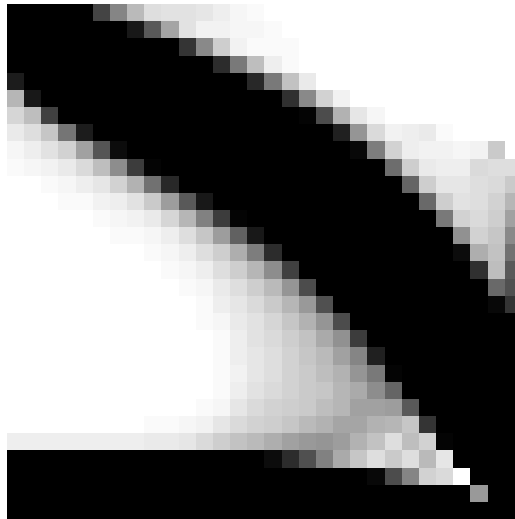
以上の方法は弾性体問題の変分原理に基づく解法であるが、同様の定式化が弾性体以外にも可能である。例えば[82, 84]では、弾塑性変形をする固体に対してポテンシャルエネルギーを定義し、歪みがポテンシャルエネルギーの最小化問題の解として得られることを示した。文献[82, 84]では主双対内点法を用いて歪みを求めていたが、この結果を用いれば載果履歴に除荷を含まない場合は、弾塑性変形をする構造物もiSA法の対象になりうる。

表 3.1. コンプライアンス最小化問題における iSA 法と既存手法の反復回数

	最適化の反復回数	CG法の合計反復回数
iSA法	456	456
既存手法	47	1051



(a) iSA 法による最適化結果 ($E = 25.600$).



(b) 既存手法による最適化の結果 ($E = 27.919$).

図 3.1. 弾性体のトポロジー最適化問題の最適解.

第 4 章

多孔質媒体モデルによる最適化

本章では多孔質媒体モデルを用いた流れ場のトポロジー最適化問題を定式化し，iSA 法に基づいて最適解を求める．流れ場の解法として LBM を利用し，LBM に対する iSA 法による感度の計算方法を提案する．また数値計算例により手法の妥当性を検証する．

4.1 多孔質媒体モデルによる流れ場のトポロジー最適化

一般的に流れ場はカオス的な挙動を示すことが知られている．しかしながら Reynolds 数 Re の低い流れ，つまりごく遅い流れや粘性の大きな流れは，挙動が比較的単純で議論がしやすいという特徴がある．身の回りの流れの多くは Reynolds 数が高いことは事実であるが，例えば医療器具等への活用が期待されているマイクロチャンネル流れ [5] など，低 Reynolds 数流れにも工学的に興味のある流れがある．本章では，低 Reynolds 数流れに適用することを前提として，流れ場のトポロジー最適化問題を定式化し，iSA 法を用いて解く．

低 Reynolds 数流れの中でも， $Re \rightarrow 0$ の極限の流れは Stokes 流れと呼ばれる．Stokes 流れは定常な方程式である Stokes 方程式に従うことが知られている．Stokes 方程式は線形であり，滑らかで一意的な解が存在することが知られている．Pironneau [59] は Stokes 流れ中に物体が置かれた場合のエネルギー損失最小化問題を解析的に解いた．しかしながらこのように解析的に解が求まる問題は限られている．その後，計算機の登場と発展により，数値的に最適形状を求める手法が盛んに研究された．Stokes 流れに対するトポロジー最適化手法には [1, 7, 10, 14, 21, 59] などがある．Borrvall and Peterson [7] の多孔質媒体モデルは多くのトポロジー最適化手法で採用されており，基本的なモデルである．そこで本章でも多孔質媒体モデルを用いて最適化手法を構築する．

本章は次のように構成される．第 4.2 節では最適化問題を定式化し，最適化手法を構築する．第 4.3 節では数値計算例を示して手法の妥当性を検証する．最後に第 4.4 節で結論を述べる．

4.2 エネルギー損失最小化問題と iSA 法による解法

多孔質媒体モデル [7] は、領域が多孔質媒体で充たされていると仮定し、その空隙率を変数として最適化を行う。したがって空隙率を設計変数 ϕ とする。このとき、2.4 節で導いた LBM を用いて流れ場を数値的に解くことができる。

目的関数をエネルギー損失として、その最小化問題を考える。多孔質媒体中の流れにおけるエネルギー損失は粘性散逸関数 D と摩擦散逸関数 C の和で定義される：

$$D(\mathbf{u}, \phi) = \frac{1}{2} \int_{\Omega} \phi^p \nu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\|^2 dx, \quad (4.1)$$

$$\begin{aligned} C(\mathbf{u}, \phi) &= - \int_{\Omega} \mathbf{F} \cdot \mathbf{u} dx \\ &= \int_{\Omega} \alpha(\phi) \|\mathbf{u}\|^2 dx. \end{aligned} \quad (4.2)$$

すなわち 3 章の定式化における目的関数 E は、本章では $D + C$ である。(4.1) における係数 p は物理的には零であるべきである。しかしながら、Borrval and Peterson [7] によると $2 < p < 10$ 程度の値を使用することで、収束性が向上する。そこで本節でも同様に $2 < p < 10$ とする。なお、 $p \neq 0$ であっても、 ϕ が 0 又は 1 であれば $\phi^p = \phi$ であるから、完全に流体又は固体の領域には影響を及ぼさない。

次に最適化問題を書き下す。定常 Navier–Stokes 方程式に従う流れ場におけるエネルギー損失の最小化問題は、次のように書き下せる。

$$(P_{NS}) : \min_{\phi, \mathbf{u}} D(\mathbf{u}, \phi) + C(\mathbf{u}, \phi) \quad (4.3a)$$

$$\text{s. t. } \mathbf{u} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = -\text{grad} p + \nu \Delta \mathbf{u}, \quad (4.3b)$$

$$0 \leq \phi(\mathbf{x}, t) \leq 1, \quad (4.3c)$$

$$\int_{\Omega} \phi(\mathbf{x}, t) dx = \bar{V}. \quad (4.3d)$$

ただし (4.3b) が定常 Navier–Stokes 方程式であり、 ν は動粘性係数である。(4.3c) は ϕ の定義より与えられる制約である。また、(4.3d) は流体領域の体積を指定する体積制約である。構造力学におけるトポロジー最適化では、体積制約は構造物の重量を一定に保つという意味がある。しかし流路のトポロジー最適化においては、そのような意味付けは難しい。一方で体積制約を設けない場合、最適解として設計領域全体が流体領域である解が得られる場合がある。この結果は工学的に望ましい結果ではないため、体積制約を設けてそのような解に陥らないような工夫をする。

第 3 章では偏微分方程式の解が時間発展方程式を解くことで得られることを仮定した。本章以降では、第 3 章で定式化した LBM を解くことで近似的に Navier–Stokes 方程式の解を得る。以降では時間と空間について離散化し、 \mathbf{u} と ϕ について (1.1) の定義を用いる。すなわ

ち、第3章における関数 \mathcal{A} および \mathcal{B} はそれぞれ

$$\begin{aligned}\mathcal{A}(\mathbf{u}, \phi) &= -\mathbf{u} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} - \text{glad}p + \nu \Delta \mathbf{u}, \\ \mathcal{B}(\mathbf{u}_k, \phi_k) &= \mathbf{u}_{\text{LBM}}(\mathbf{f}_k, \phi_k)\end{aligned}$$

に対応する。ここで、連続関数を用いて定式化された最適化問題 (3.2) は (4.3) に対応している。第3.2節では、手法の解釈のために最適化問題 (3.16) を導いた。本節において (3.16) に対応する最適化問題を書き下すと、次のようになる：

$$(\text{P}_{\text{porosity}}) : \min_{\phi_k, \mathbf{u}_k} D(\mathbf{u}_k, \phi_k) + C(\mathbf{u}_k, \phi_k) \quad (4.4a)$$

$$\text{s. t. } \mathbf{u}_k = \mathbf{u}_{\text{LBM}}(\mathbf{f}_{k-1}, \phi_k), \quad (4.4b)$$

$$0 \leq \phi(\mathbf{x}, t_k) \leq 1, \quad (4.4c)$$

$$\int_{\Omega} \phi(\mathbf{x}, t_k) d\mathbf{x} = V_k. \quad (4.4d)$$

ここで (4.4b) は第2.1節で定義したように、LBM の手続きにしたがって流れ場を得ることを示している。以降では時間と空間について離散化したものを考え、(1.1) の定義にしたがって記述する。

以降では、iSA 法を用いて $(\text{P}_{\text{porosity}})$ を近似的に解く。(4.4c) は ϕ の定義より与えられる制約であり、(4.4d) は流体領域の体積を指定する体積制約である。本来、体積制約は最適解で満たされていればよく、(4.4d) は $k = N$ でのみ満たされればよい。一般的な構造ないし流路の最適化問題の実装においては、 V_k を $k \geq 1$ において定数とする実装を行うことが多い。このとき一般に、初期値である V_0 と、制約条件とする V_1 に差がある。iSA 法では設計変数と場の変数を並行して更新するため、 V_k の変化が大きい場合に \mathbf{u} が大きく変化し、望ましい結果が得られない場合がある。そこで V_k を連続的に変化させ、 \mathbf{u} に不必要な擾乱を与えないような工夫をする。たとえば、 V_k を次のように与える：

$$V_k = \bar{V} - (\bar{V} - V_0) \exp^{-\beta k}. \quad (4.5)$$

ただし β は V_k の \bar{V} への漸近の速さを制御する定数である。

(3.11) で導いた、ステップ数に依存する Lagrange 関数は、 $(\text{P}_{\text{porous}})$ に対しては、

$$\begin{aligned}\mathcal{L}_k &= D(\mathbf{u}_k, \phi_k) + C(\mathbf{u}_k, \phi_k) - \langle \mathbf{u}_k - \mathbf{u}_{\text{LBM}}(\mathbf{u}_{k-1}, \phi_k), \mathbf{v} \rangle - \langle \phi_k, \boldsymbol{\lambda}^1 \rangle \\ &\quad - \langle \mathbf{1} - \phi_k, \boldsymbol{\lambda}^2 \rangle - \left(\int \phi_k d\Omega - V_k \right) M_k\end{aligned}$$

と書き表される。ただし $\boldsymbol{\lambda}_k^1$ および $\boldsymbol{\lambda}_k^2$ はともに未定乗数である。直接法により感度を求めると、感度は次式で定式化される：

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k(\mathbf{x})} = \frac{\partial (D + C)_k}{\partial \phi_k(\mathbf{x})} + \frac{\partial \mathbf{u}_{\text{LBM}}(\mathbf{u}_{k-1}, \phi_k)}{\partial \phi_k(\mathbf{x})} \mathbf{v} - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k. \quad (4.6)$$

ただし $\phi_k(\mathbf{x})$ に対応する未定乗数 λ_k^1, λ_k^2 をそれぞれ $\lambda_k^1(\mathbf{x}), \lambda_k^2(\mathbf{x})$ と表記する. なお M_k はスカラーであり位置 \mathbf{x} に依存しない. ここで $\frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k} = \mathbf{0}^\top$ より

$$\begin{aligned} \frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k} &= \frac{\partial(D+C)_k}{\partial \mathbf{u}_k} - \mathbf{v}^\top = \mathbf{0}^\top \\ \Leftrightarrow \mathbf{v} &= \frac{\partial(D+C)_k^\top}{\partial \mathbf{u}_k} \end{aligned} \quad (4.7)$$

が得られる. (4.7) を (4.6) に代入することで

$$\begin{aligned} \frac{\partial \mathcal{L}_k}{\partial \phi_k(\mathbf{x})} &= \frac{\partial(D+C)_k}{\partial \phi_k(\mathbf{x})} + \frac{\partial(D+C)_k}{\partial \mathbf{u}_k} \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k \\ &= \nabla_{\phi_k(\mathbf{x})}(D+C) - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k \end{aligned}$$

を得る.

iSA 法にしたがって, (3.36) を今回の場合に適用すると, 次の更新式が得られる,

$$\mathbf{u}_{k+1} = \mathbf{u}_{\text{LBM}}(\mathbf{u}_k, \phi_{k+1}), \quad (4.8a)$$

$$\phi_{k+1}(\mathbf{x}) = \phi_k(\mathbf{x}) + \delta\phi_{k+1}(\mathbf{x}), \quad (4.8b)$$

$$\delta\phi_{k+1}(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})}(D+C) - M_{k+1} \leq 0, \\ -\kappa \nabla_{\phi_{k+1}(\mathbf{x})}(D+C) - M_{k+1} & \text{if } 0 < \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})}(D+C) \\ & - M_{k+1} < 1, \\ 1 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})}(D+C) - M_{k+1} \geq 1. \end{cases} \quad (4.8c)$$

(4.8) を解くために $\nabla_{\phi_k}(D+C)$ を導く. 最初に, ϕ_{k+1} は \mathbf{u}_{k+1} に影響を与えるが \mathbf{u}_k には影響を与えないことに注意して微分を行う. まず定義より,

$$\nabla_{\phi_{k+1}(\mathbf{x})}(D+C) = \frac{\partial(D+C)_{k+1}}{\partial \mathbf{u}_{k+1}} \frac{\partial \mathbf{u}_{k+1}}{\partial \phi_{k+1}(\mathbf{x})} + \frac{\partial(D+C)_{k+1}}{\partial \phi_{k+1}(\mathbf{x})}. \quad (4.9)$$

と分解される. 右辺の各項は (4.1), (4.2) より,

$$\frac{\partial(D+C)_{k+1}}{\partial \mathbf{u}_{k+1}} = (-\nu \Phi_k^p \Delta \mathbf{u}_{k+1}(\phi_k) \delta r + 2\alpha \mathbf{u}_{k+1}(\phi_k) \delta r)^\top, \quad (4.10)$$

$$\frac{\partial(D+C)_{k+1}}{\partial \phi_{k+1}(\mathbf{x})} = \frac{p}{2} \phi_{k+1}(\mathbf{x})^{p-1} \nu \|\nabla \mathbf{u}_k(\mathbf{x}) + (\nabla \mathbf{u}_k(\mathbf{x}))^\top\|^2 \delta r + \frac{\partial \alpha}{\partial \phi_k} \|\mathbf{u}_k(\mathbf{x})\|^2 \delta r \quad (4.11)$$

で与えられる. ただし Φ_k^p は $\Phi_k^p = \text{diag}(\phi_k(\mathbf{x}_1)^p, \phi_k(\mathbf{x}_1)^p, \phi_k(\mathbf{x}_2)^p, \dots, \phi_k(\mathbf{x}_N)^p)$ であり, $\phi_k(\mathbf{x}_i)$ を, 次元の数に相当する 2 個ずつ並べた対角行列である. (4.10) の右辺に含まれる $\mathbf{u}_{k+1}(\phi_k)$ は, $\phi_{k+1} = \phi_k$ として LBM により計算した場の変数を示す. すなわち $\mathbf{u}_{k+1}(\phi_k) = \mathbf{u}_{\text{LBM}}(\mathbf{u}_k, \phi_k)$ である. しかし, (4.10) を計算している時刻は $t = t_k$ であり, \mathbf{u}_{k+1} の値は未知である上に, t_{k+1} での流れ場は $\mathbf{u}_{k+1}(\phi_{k+1})$ であり, $\mathbf{u}_{k+1}(\phi_k)$ は (4.10) でしか使用されない. $\mathbf{u}_{k+1}(\phi_k)$ の計算には LBM の手続きが必要であり, 計算コストが大きい. そこで $\mathbf{u}_{k+1}(\phi_k) \simeq \mathbf{u}_{k+1}(\phi_{k+1})$ という近似を用いる. この近似は最適解においては \mathbf{u} も ϕ

も定常であり, $\mathbf{u}_{k+1}(\phi_{k+1}) = \mathbf{u}_{k+1}(\phi_k) = \mathbf{u}_k(\phi_k)$ を満たすことから正当化される. この近似を用いることで, (4.10) は

$$\frac{\partial(D+C)_{k+1}}{\partial\mathbf{u}_{k+1}} \simeq (-\nu\Phi_k^p\Delta\mathbf{u}_k\delta r + 2\alpha\mathbf{u}_k\delta r)^\top \quad (4.12)$$

と近似される. 次に (4.9) の右辺の $\frac{\partial\mathbf{u}_{k+1}}{\partial\phi_{k+1}}$ は, 再び連鎖側により次のように分解される:

$$\frac{\partial\mathbf{u}_{k+1}}{\partial\phi_{k+1}(\mathbf{x})} = -\frac{0.5}{1+0.5\alpha}\frac{\partial\alpha}{\partial\phi_{k+1}(\mathbf{x})}\hat{\mathbf{u}} + \frac{1}{1+0.5\alpha}\sum_{i=0}^6\frac{\partial\hat{\mathbf{u}}_{k+1}}{\partial f_i(\mathbf{x}, t_{k+1})}\frac{\partial f_i(\mathbf{x}, t_{k+1})}{\partial\phi_{k+1}(\mathbf{x})}. \quad (4.13)$$

定義 (2.3) より

$$\frac{\partial\hat{\mathbf{u}}}{\partial f_i} = \frac{\mathbf{c}_i - \hat{\mathbf{u}}}{\rho} \quad (4.14)$$

が得られる. また, BGK モデルにおける格子 Boltzmann 方程式 (2.4), (2.10), (2.11) より次の式が得られる:

$$\begin{aligned} \frac{\partial f_i(\mathbf{x}, t_{k+1})}{\partial\phi_{k+1}(\mathbf{x})} &= \frac{\partial f_i^{\text{col}}(\mathbf{x} - \mathbf{e}_i, t_{k+1})}{\partial\phi_{k+1}(\mathbf{x})} \\ &= \frac{1}{\tau}\frac{\partial f_i^{\text{eq}}(\mathbf{x}, t_k)}{\partial\phi_{k+1}(\mathbf{x})} + \delta t\frac{\partial F_i(\mathbf{x}, t_k)}{\partial\phi_{k+1}(\mathbf{x})}, \end{aligned} \quad (4.15)$$

$$\frac{\partial f_i^{\text{eq}}(\mathbf{x}, t_k)}{\partial\phi_{k+1}(\mathbf{x})} = -\omega_i\rho\frac{(\mathbf{u}_k \otimes \mathbf{u}_{k+1}) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2\mathbf{I})}{2c_s^4}\frac{1}{\phi_{k+1}(\mathbf{x})^2}, \quad (4.16)$$

$$\frac{\partial F_i(\mathbf{x}, t_k)}{\partial\phi_{k+1}(\mathbf{x})} = -\omega_i\rho\left(1 - \frac{1}{2\tau}\right)\left[\frac{\mathbf{c}_i \cdot \mathbf{u}_k}{c_s^2}\frac{\partial\alpha}{\partial\phi_{k+1}(\mathbf{x})}\right. \quad (4.17)$$

$$\left. + \frac{(\mathbf{u}_k \otimes \mathbf{u}_k) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2\mathbf{I})}{2c_s^4}\frac{\partial}{\partial\phi_{k+1}(\mathbf{x})}\left(\frac{\alpha}{\phi_{k+1}(\mathbf{x})}\right)\right]. \quad (4.18)$$

(4.9), (4.11)–(4.18) により, $\nabla_{\phi_{k+1}}(D+C)$ が計算できる. 以上により得られる $\nabla_{\phi_{k+1}}(D+C)$ を (4.8) に代入して計算することで, 最適解を得る. 以上の一連の計算は要的な計算であって反復がない.

4.3 数値計算例

(4.8) を用いた数値計算例を示す. 数値計算には XEON X5690 (3.33GHz, 8GB メモリ) を使用し, C++ を用いて実装した.

4.3.1 縮流管問題

本節では縮流管問題を解く. 図 4.1 に計領域と境界条件を示す. 入口と出口の流速は次の通り定義する:

$$\begin{aligned} U_{\text{in}}(y) &= -\bar{U}y(y-L), \\ U_{\text{out}}(y) &= \begin{cases} -3\bar{U}(y - \frac{L}{3})(y - \frac{2L}{3}) & \text{for } y \in \{y \mid \frac{L}{3} \leq y \leq \frac{2L}{3}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{L}{3}, \frac{2L}{3} \leq y\}. \end{cases} \end{aligned}$$

ただし Reynolds 数は $Re = 10.0$ とする．また体積制約は $\bar{V} = 0.5|\Omega|$ とし，格子点数は x 方向に 52 点， y 方向に 61 点とする．

最適化の過程における形状の変化の様子を図 4.2 に示す．また目的関数の履歴を D と C の値も含めて図 4.3(a) に示し，各ステップにおける変数の変化量を 図 4.3(b) に示す．最適化アルゴリズムは 876 ステップで停止した．計算時間は 3.61 秒であった．図 4.3(b) に示すように，収束履歴はおおよそ単調である．初期状態 (図 4.2(a)) では入口と出口は接続されておらず，最適化の初期段階 (図 4.2(b)) でも接続されていない．その後，100 ステップ前後 (図 4.2(c)) で入口と出口が接続され始めた．それとともに図 4.3(a) に示すように目的関数の値も急に減少した．さらに 150 ステップ (図 4.2(d)) 以降では流路が完全に接続され，目的関数は徐々に減少した．完全に流体でも固体でもない領域をグレーな領域と呼ぶ．150 ステップ目までにかけてはグレーな領域が広く存在するが，その後，そのような領域は徐々に縮小する．最適解においては，個体と流体の境界部分にだけグレーな領域が存在する．実用上はグレーな部分を勘案しながら，設計者が適切な境界の位置を判断することになる．

Borrvall and Peterson [7] は，本例題と同じ問題を有限要素法を用いて解いた．本最適化結果は入口境界の近傍を除いて，その結果とほぼ一致する．入口境界条件の付近では，本結果の方がグレーな領域が少なく，より明確な境界が得られている．この違いは境界条件による違いと考えることができる．

次に iSA 法により得られる感度と，定常状態の流速を使用した感度を比較した．定常状態における感度を計算するため，100 ステップ目で ϕ を固定し， \mathbf{u} が定常になるまで LBM を計算し，感度を計算した．二種類の感度を 図 4.4 に示す．両者はわずかに異なる．iSA 法による感度 (図 4.4(a)) では，領域の中央付近の感度が定常状態の流れを利用した感度 (図 4.4(b)) に比べて大きい．この違いは流速の違いに起因している．iSA 法では \mathbf{u} は非定常性を持っており，感度を計算した時点で， ϕ が 0 に近い部分で \mathbf{u} が小さいために感度も小さかった．しかしながら定常状態まで計算すると， \mathbf{u} の分布が変わるために感度も異なる．

また，同じ縮流管問題を細かな格子を使用して最適化した．格子点数は x 方向に 104 点， y 方向に 122 点である．これ以外の条件は同様である．最適解と収束の履歴を図 4.5 に示す．最適解は粗い格子を使用した場合と変わらない．収束前に要したステップ数は 1900 ステップで，粗い格子を用いた場合のおよそ二倍であった．計算時間は 37.62 秒であった．これは一ステップあたり 0.0198 秒であり，粗い格子の場合に比べて 4.75 倍大きい．これは格子点数が 4 倍になったことを考慮すると納得できる数値である．結果として，格子の密度は最適化の結果に影響しないことがわかる．計算に要する時間は格子点数と，時間ステップ幅の逆数に比例する．

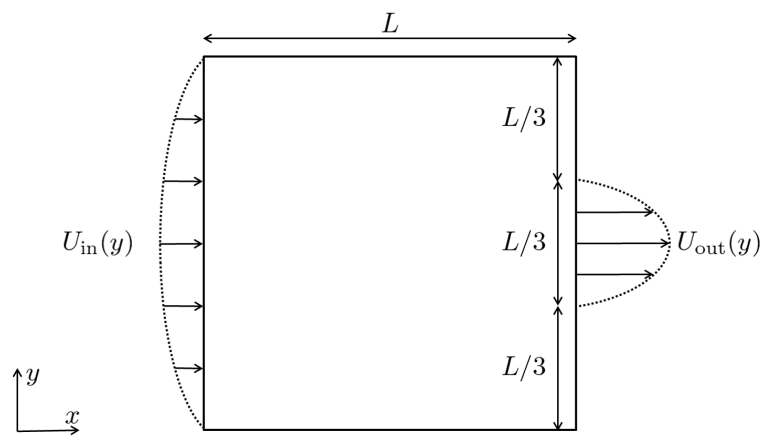


図 4.1. 縮流管問題の設計領域と境界条件.

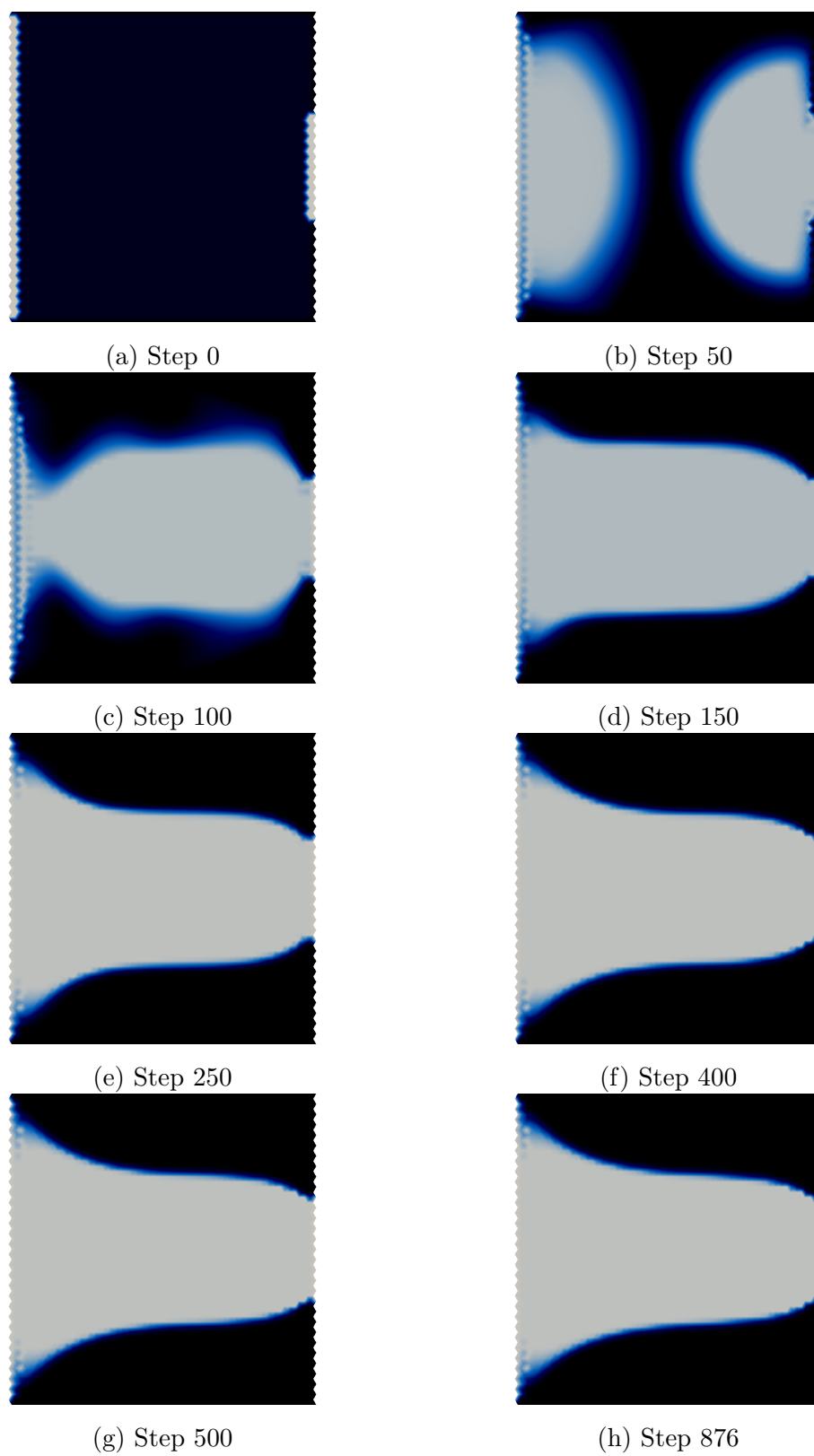
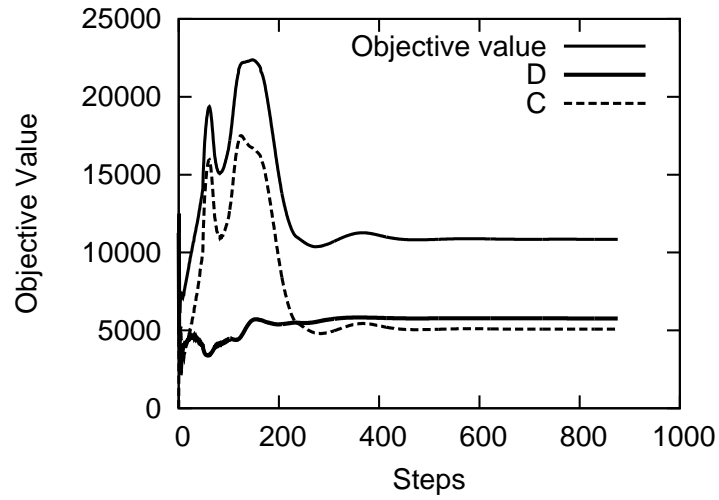
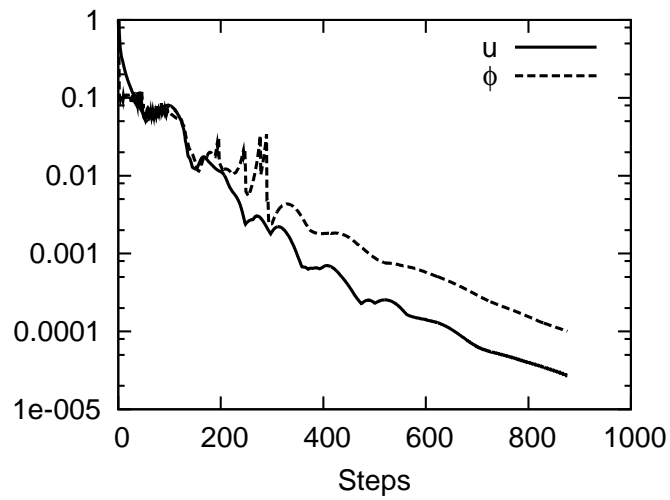


図 4.2. 縮流管問題の設計変数の変化.

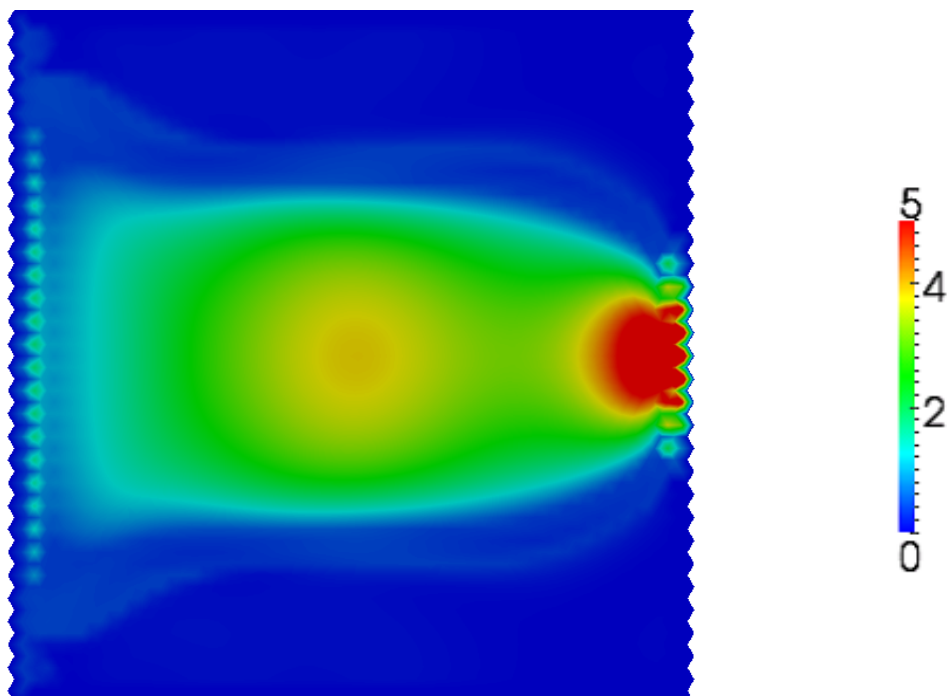


(a) 目的関数値.

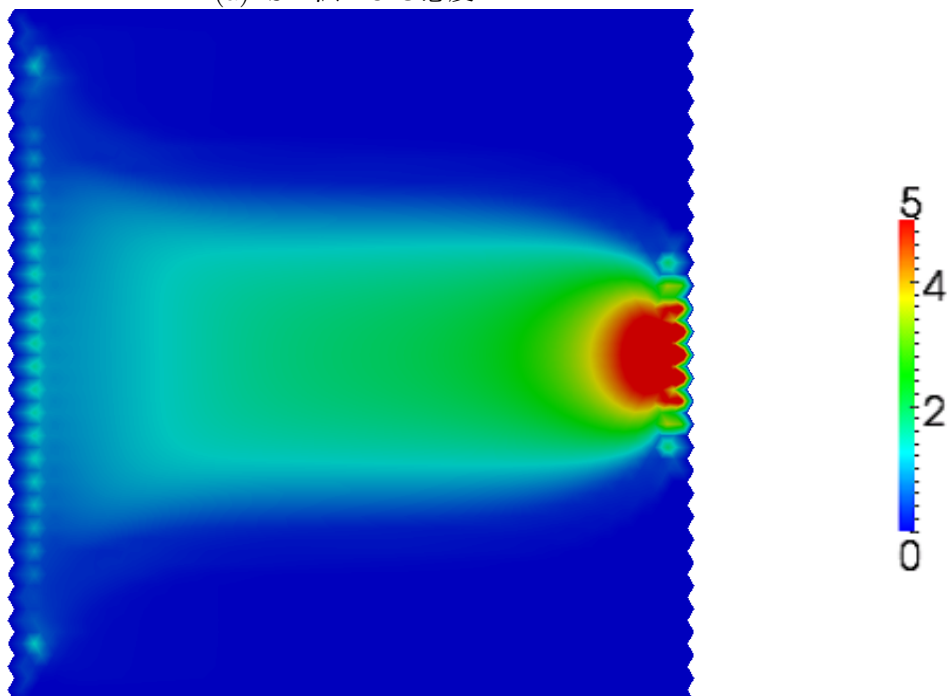


(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$.

図 4.3. 縮流管問題の収束履歴

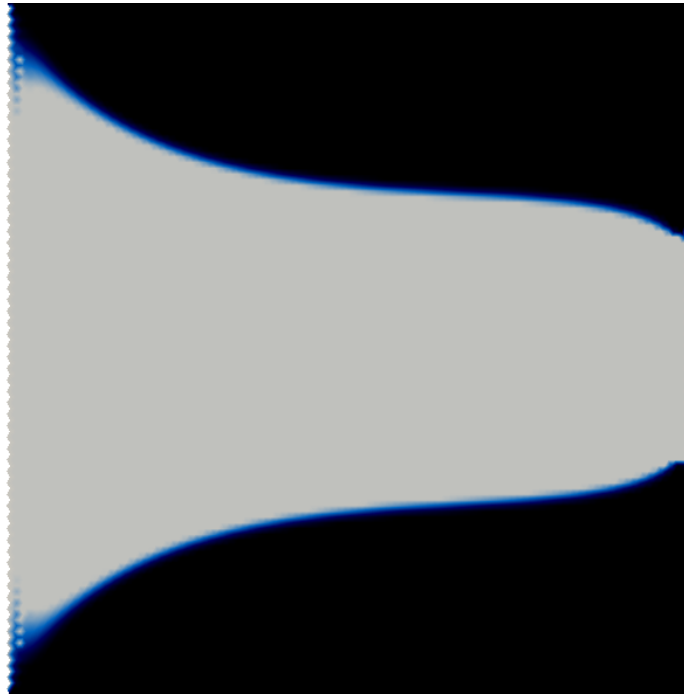


(a) iSA 法による感度.

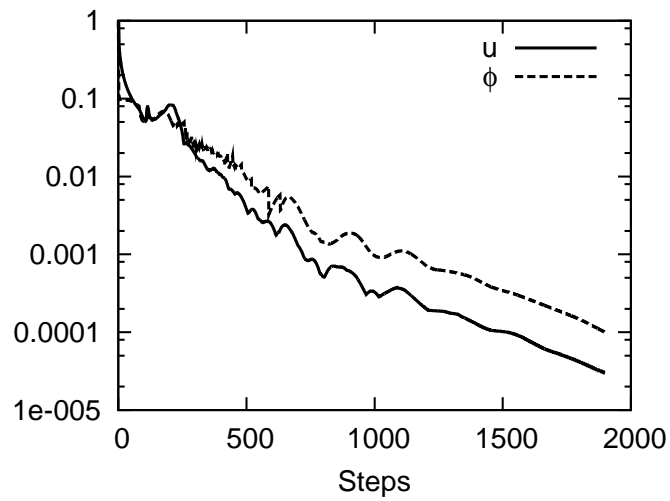


(b) 定常状態の u を用いた場合の感度.

図 4.4. Step 100 における感度.



(a) 最適解.



(b) 収束履歴.

図 4.5. 細かな格子を用いた場合の結果.

4.3.2 合流管問題

本節では合流管問題を解く．合流管問題は二箇所の入口から流入する流れが合流し，一箇所の出口から流出する問題である．設計領域と境界条件を図 4.6 に示す．入口流速と出口流速は次のように定義する：

$$\begin{aligned}
 U_{\text{in1}}(y) &= \begin{cases} -\bar{U}(y - \frac{3L}{8})(y - \frac{5L}{8}) & \text{for } y \in \{y \mid \frac{3L}{8} \leq y \leq \frac{5L}{8}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{3L}{8}, \frac{5L}{8} \leq y\}, \end{cases} \\
 U_{\text{in2}}(x) &= \begin{cases} -0.5\bar{U}(x - \frac{L}{2})(x - \frac{3L}{4}) & \text{for } x \in \{x \mid \frac{L}{2} \leq x \leq \frac{3L}{4}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{L}{2}, \frac{3L}{4} \leq y\}, \end{cases} \\
 U_{\text{out}}(x) &= \begin{cases} -1.5\bar{U}(y - \frac{L}{2})(y - \frac{3L}{4}) & \text{for } y \in \{y \mid \frac{L}{2} \leq y \leq \frac{3L}{4}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{L}{2}, \frac{3L}{4} \leq y\}. \end{cases}
 \end{aligned}$$

ただし Reynolds 数は $Re = 1.0$ とする．格子点数は x 方向に 52 点， y 方向に 61 点とする．体積制約は $\bar{V} = 0.35|\Omega|$ とする．最適化の過程における形状の変化の履歴を図 4.7 に示し，目的関数と変数の変化分の履歴を図 4.8 に示す．最適化アルゴリズムは 2515 ステップで停止した．計算時間は 11.10 秒であった．

最適化の初期（図 4.7(c), (d)）では，左側の入口と出口が接続されている一方で，下側の入口はその流れに接続されていない．下側からの流れはグレーな領域を流れている．その後，図 4.7(e) において，下側からの流れが主流に接続された．それに従って，摩擦損失 C が大幅に下がった．これは，グレーな領域が少なくなったことに起因している．一方で散逸関数 D はあまり変化しない．結果として目的関数は全体的に減少している．500 ステップ以降，流れは大きく変化することはないが徐々に変化し，目的関数も漸減した．図 4.8(b) を見ると， ϕ と \mathbf{u} の変分は振動しながらも減少する．特に 200 ステップ近辺で ϕ と \mathbf{u} の変分が増加しているが，この時刻は前述の流れのトポロジーが変化した時刻に一致しており，トポロジーが変化するときに非単調性が現れると言える．

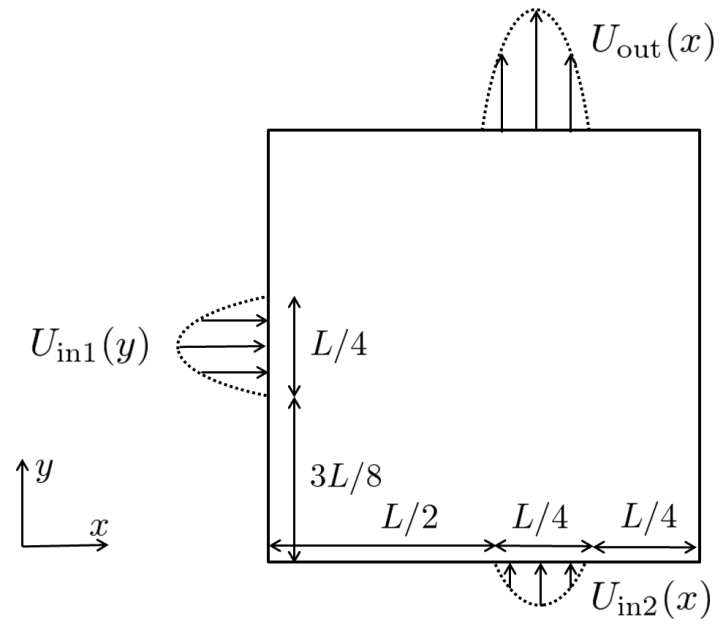


図 4.6. 合流管問題の設計領域と境界条件.

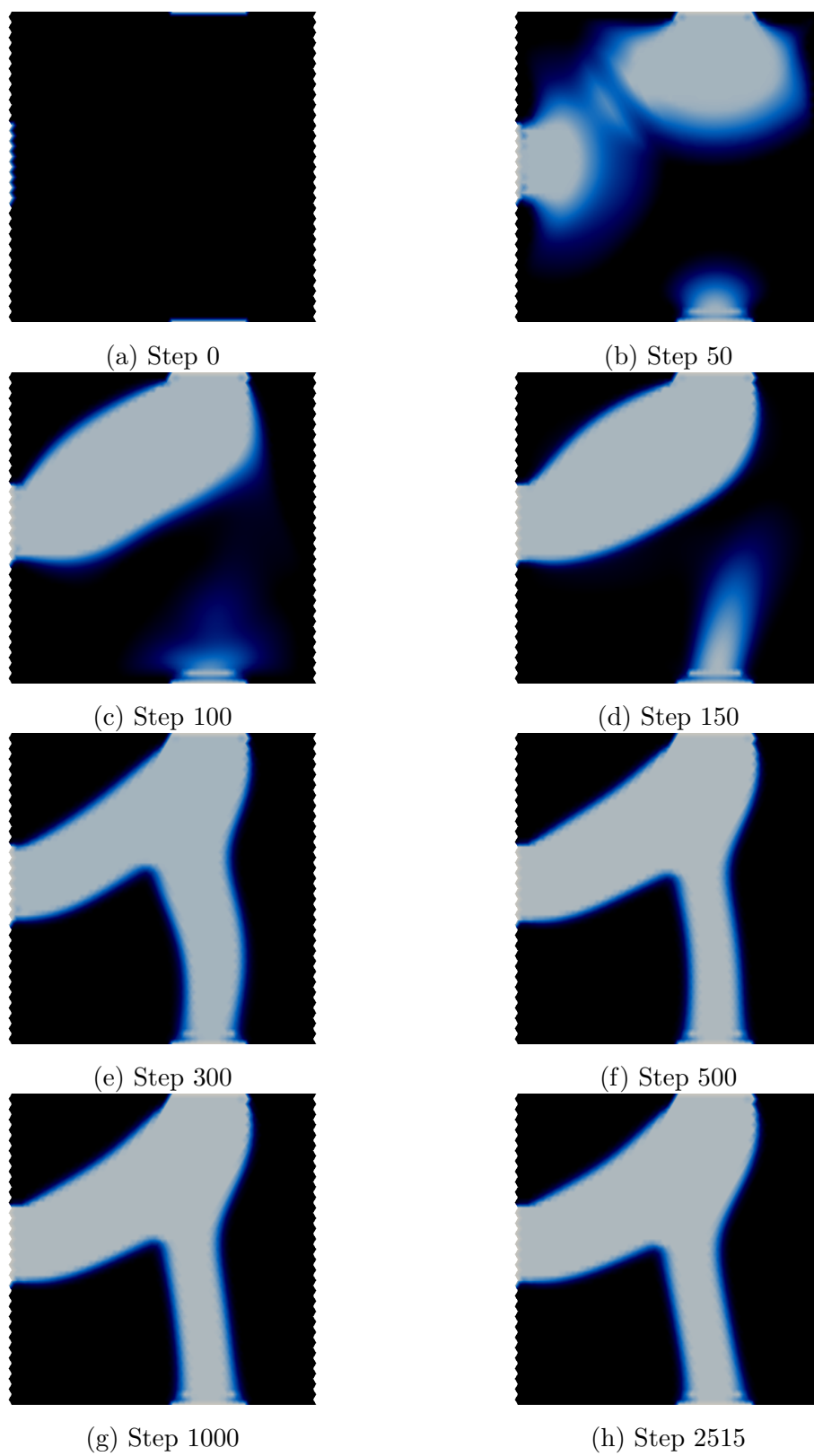
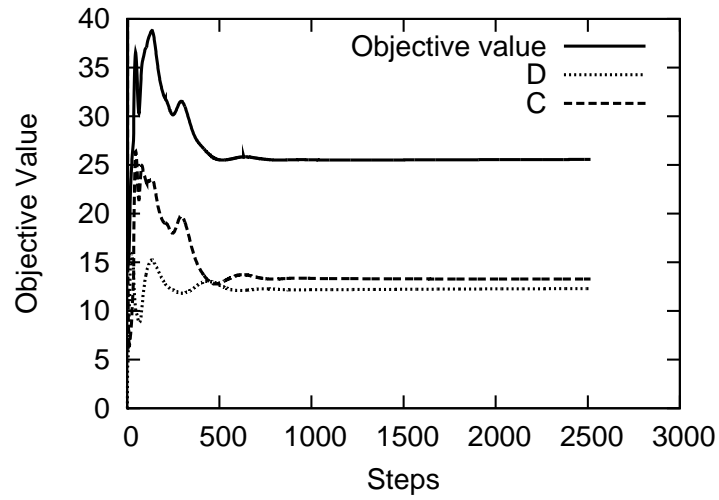
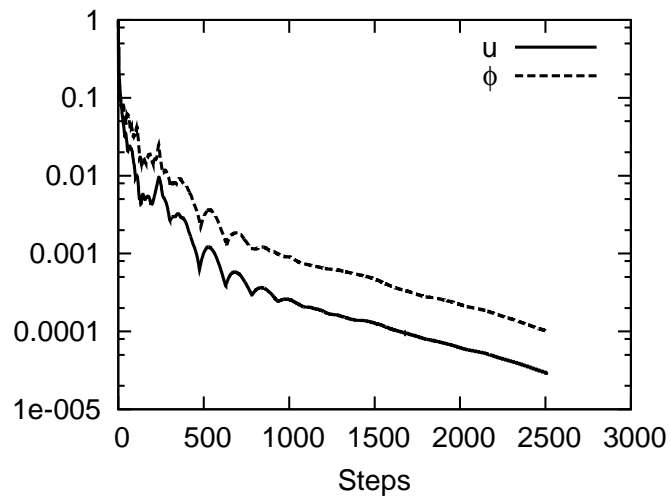


図 4.7. 合流管問題の設計変数の変化.



(a) 目的関数値.



(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$.

図 4.8. 合流管問題の収束の履歴.

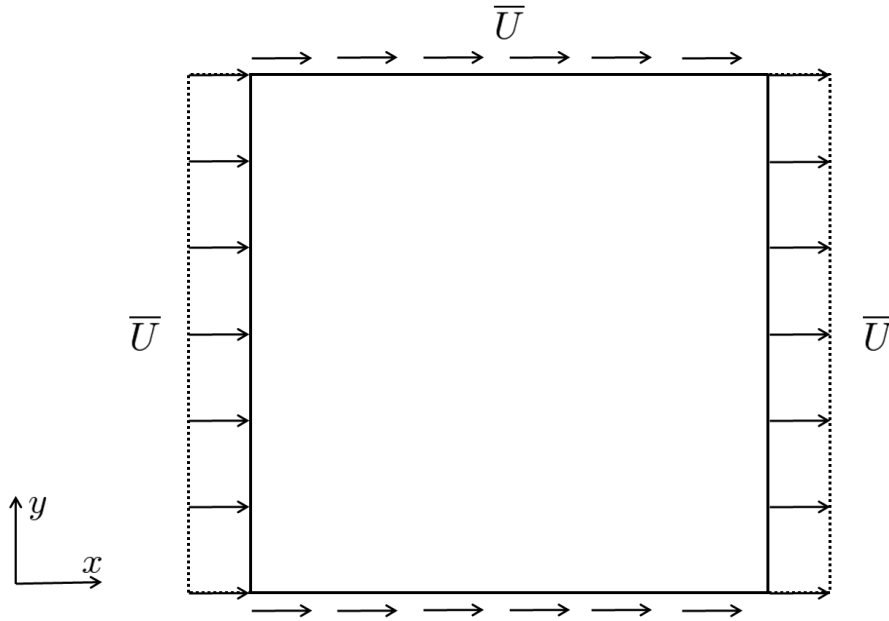


図 4.9. ラグビーボール問題の設計空間と境界条件.

4.3.3 ラグビーボール問題

本節では外部流れ問題を解く．設計領域と境界条件を図 4.9 に示す．Reynolds 数は $Re = 1.0$ とする．本例題は，Stokes 流れについては Pironneau [59] により解析的に解かれ，最適解が頂角 120 度のラグビーボール型の形状になることが知られている．格子点数は x 方向に 52 点， y 方向に 61 点とする．体積制約は $\bar{V} = 0.7|\Omega|$ とする．最適化の過程における形状変化の様子を図 4.10 に示し，収束の履歴を図 4.11 に示す．最適化アルゴリズムは 491 ステップで停止した．計算時間は 1.79 秒であった．

最適化の初期の段階である図 4.10(b) では，流れを邪魔するような固体領域が存在する．この領域はその後すぐに消失する． 150 ステップより後では，形状は僅かにしか変化しない．最終的に，図 4.10(e) に示すラグビーボール型の最適解を得た．この点で，本手法は妥当な解を得られると言える．ただし，物体の頂角は Pironneau [59] により報告されている 120 度よりも小さく，尖った形状をしている．これは，境界付近では ϕ の値が 0 と 1 の間の値を取っており設計変数の扱いが Pironneau [59] の問題設定と厳密には異なることと，格子サイズが大きいために原因と考えられる．

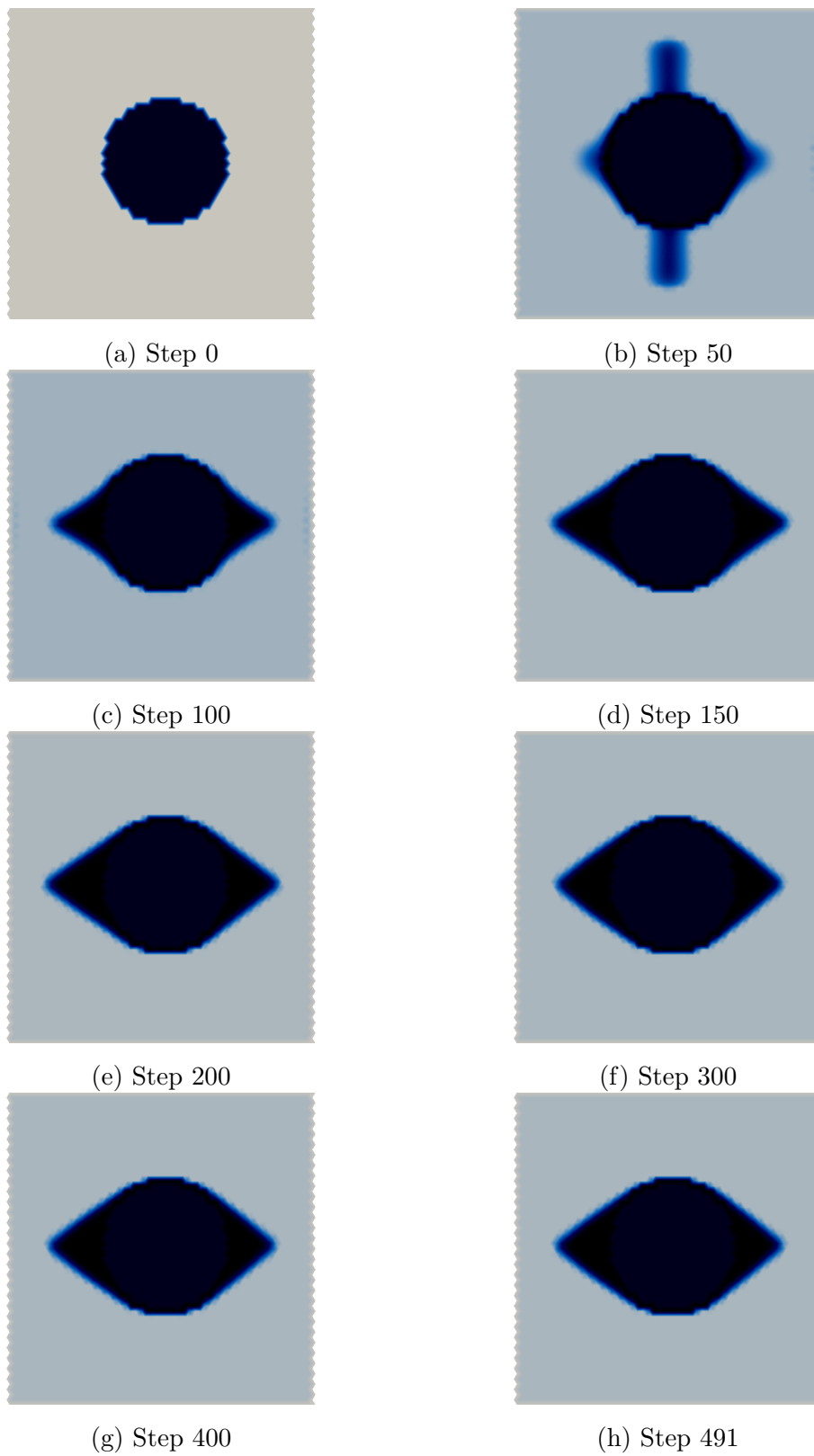
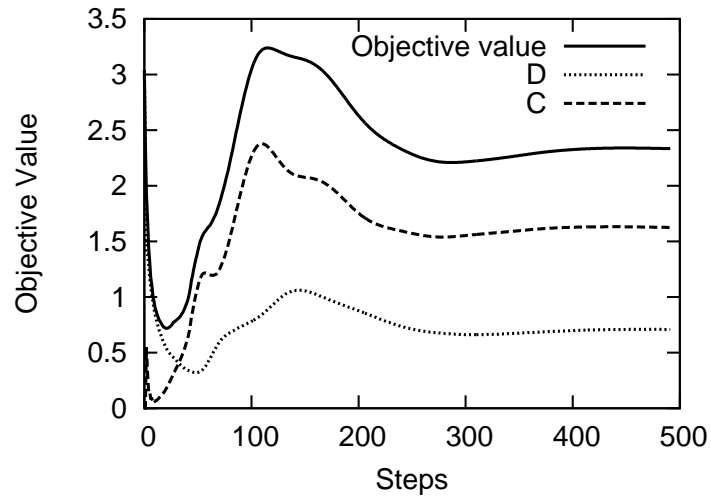
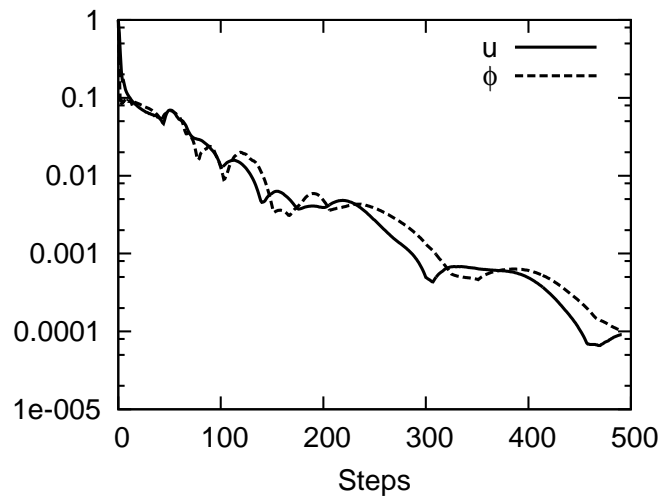


図 4.10. ラグビーボール問題の設計変数の変化.



(a) 目的関数値



(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$.

図 4.11. ラグビーボール問題の収束履歴.

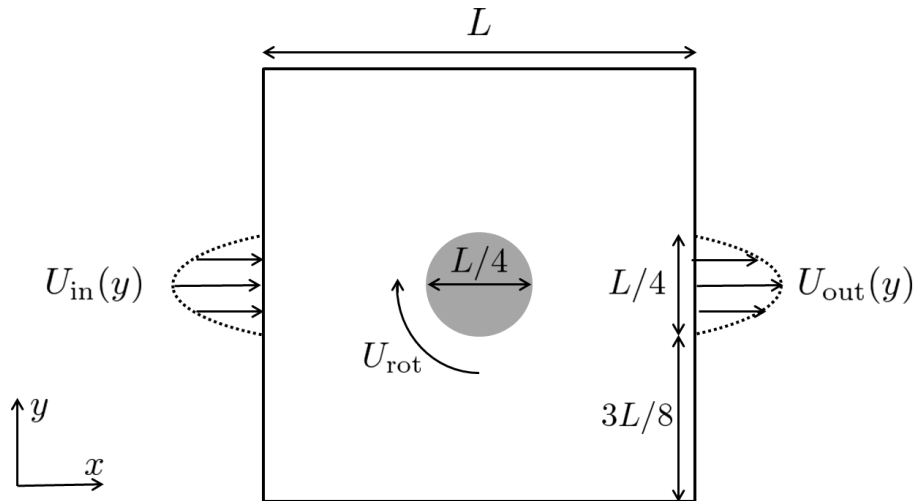


図 4.12. 回転体問題の設計領域と境界条件

4.3.4 回転体周りの問題

本節では、中央に回転する円柱がある場合の流れ場を最適化する．設計領域と境界条件を図 4.12 に示す．入口と出口は速度境界として与えられ、

$$U_{\text{in}}(y) = U_{\text{out}}(y) \begin{cases} -\bar{U}(y - \frac{3L}{8})(y - \frac{5L}{8}) & \text{for } y \in \{y \mid \frac{3L}{8} \leq y \leq \frac{5L}{8}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{3L}{8}, \frac{5L}{8} \leq y\} \end{cases}$$

で定義される．Reynolds 数は $\text{Re} = 100.0$ とする．格子点数は x 方向に 52 点、 y 方向に 61 点の計 3172 点である．体積制約は $\bar{V} = 0.3|\Omega|$ とする．設計領域の中央には円柱があり、固定領域とされている．円柱は表面での回転速度が $U_{\text{rot}} = \bar{U}$ であるように時計回りに回転している．流れ場には境界条件として、前記の速度を速度境界条件として与えた．

最適化の各ステップでの ϕ の分布を図 4.13 に示し、収束の履歴を図 4.14 と図 4.15 に示す．目的関数については D と C の値と合わせて図 4.14(a) に示すとともに、 D と C の値を図 4.14(b), (c) にそれぞれ個別に示した．最適化アルゴリズムは 710 ステップで停止し、計算時間は 3.01 秒であった．最適化の初期条件を表す図 4.13(a) では、領域全体が流体領域である．その後周辺領域で ϕ が小さくなって固体領域に近づき、回転体の上側に流路が形成される．また回転体の下側にも、回転体に引きずられるように流れ場が形成される．流体領域とも固体領域とも判別が難しい領域は、最初は存在するものの、徐々に減少し、最適解では固体領域と流体領域がおおよそ判別できる．収束の履歴に関しては図 4.15 に示すように、目的関数地がほぼ単調に減少する．最適解は回転体に沿った流路が形成されており、力学的にも妥当である．このように回転体が存在して流れを攪拌するような場合にも、本手法は妥当な解を得ることができる．

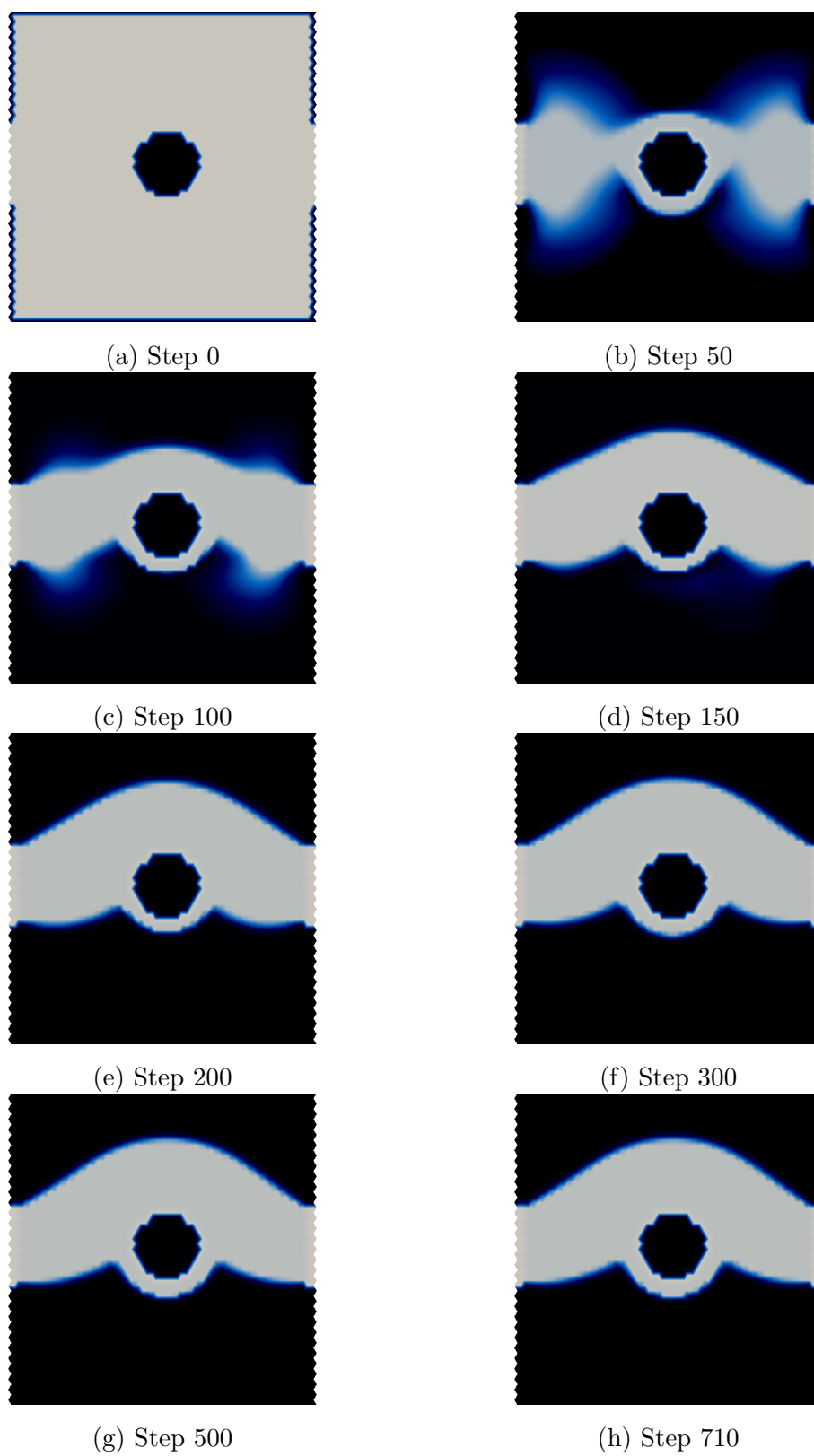
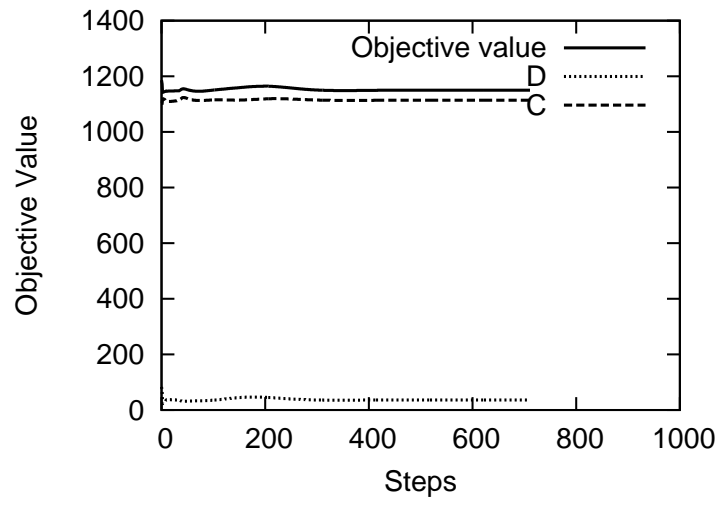
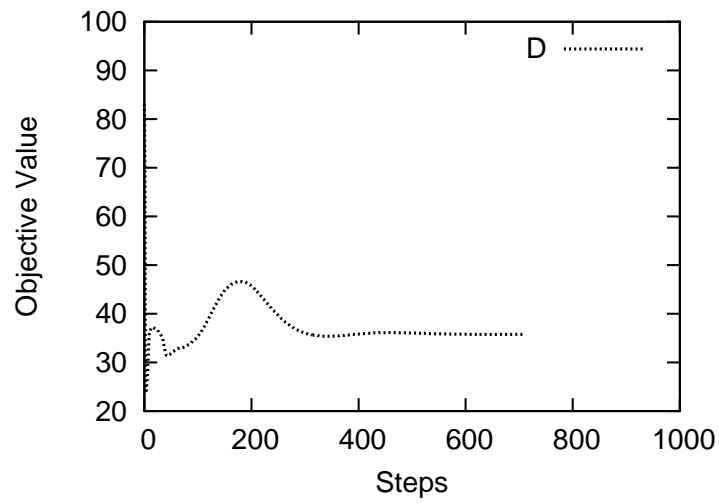


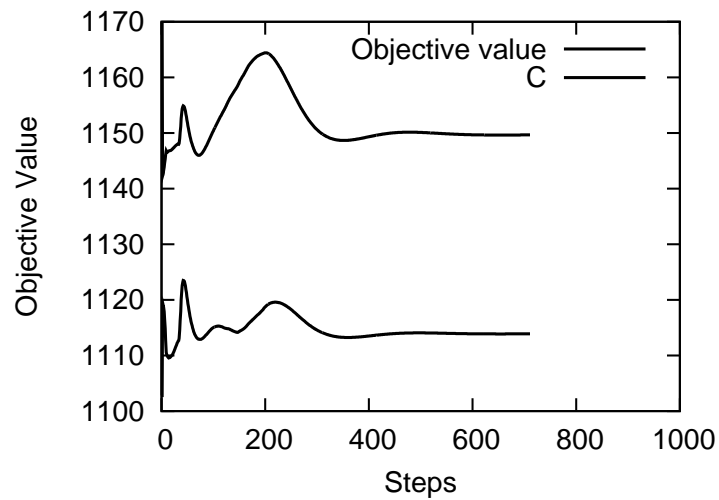
図 4.13. 回転体問題の設計変数の変化.



(a) 目的関数値.



(b) D の値の拡大図.



(c) C の値と目的関数値の拡大図.

図 4.14. 回転体問題の目的関数の収束履歴

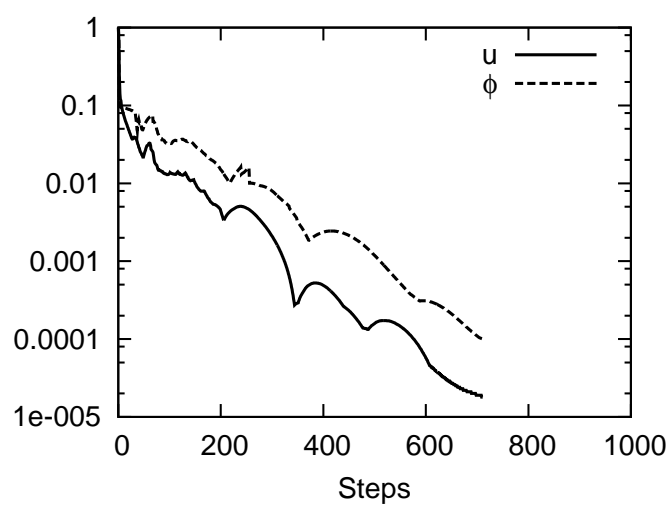


図 4.15. 回転体問題の $\|u_k - u_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$ の収束履歴

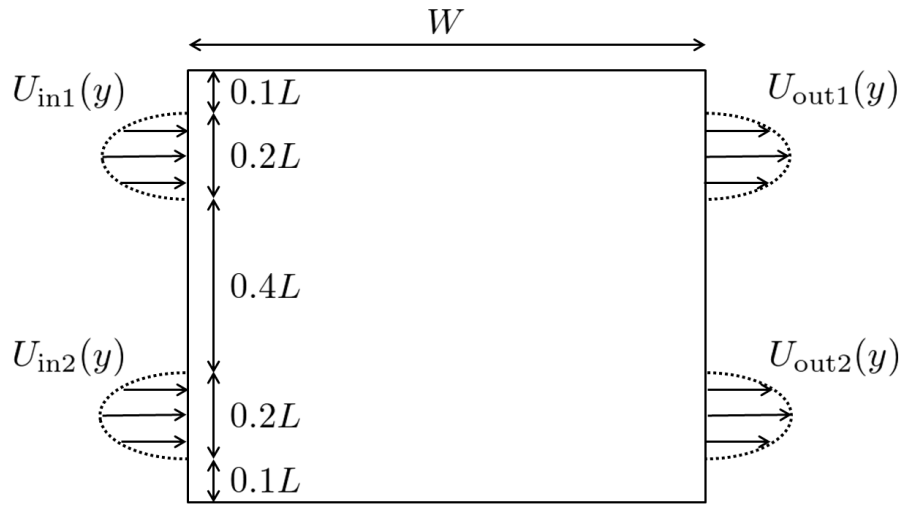


図 4.16. 二管問題の設計領域と境界条件.

4.3.5 二管問題

本節では、設計領域のパラメータによって最適解のトポロジーが変化する例を示す。問題は図 4.16 に示すように、二か所の入口と出口を備える領域の最適化とする。図 4.16 に示す x 方向の距離 W を二通りに変更して計算することとし、 $W = L$ の場合をケース A、 $W = 1.5L$ の場合をケース B とする。入口と出口は速度境界として、

$$\begin{aligned} U_{in1}(y) = U_{out1}(y) &= -\bar{U}(y - 0.1L)(y - 0.3L) & \text{for } y \in \{y \mid 0.1L \leq y \leq 0.3L\}, \\ U_{in2}(y) = U_{out2}(y) &= -\bar{U}(y - 0.7L)(y - 0.9L) & \text{for } y \in \{y \mid 0.7L \leq y \leq 0.9L\} \end{aligned}$$

で与える。Reynolds 数は $Re = 1.0$ とする。格子点数はケース A では x 方向に 52 点、ケース B では 78 点とし、 y 方向はどちらの場合も 61 点とする。体積制約は $\bar{V} = 0.4|\Omega|$ とする。

最適化の各ステップでの ϕ の分布を図 4.17, 図 4.18 に示し、収束の履歴を図 4.19, 図 4.20 に示す。ケース A の最適形状ではまっすぐな流路が二本存在し、ケース B では中央で交差する流路が存在する。特にケース A では計算の初期の段階では中央で交差している流路が、計算が進むにつれて二本に分離する。このようにパラメータの違いにより最適なトポロジーが異なる場合がある。計算ステップ数と計算時間はケース A が 2679 ステップで 9.48 秒、ケース B は 3240 ステップで 18.41 秒であった。

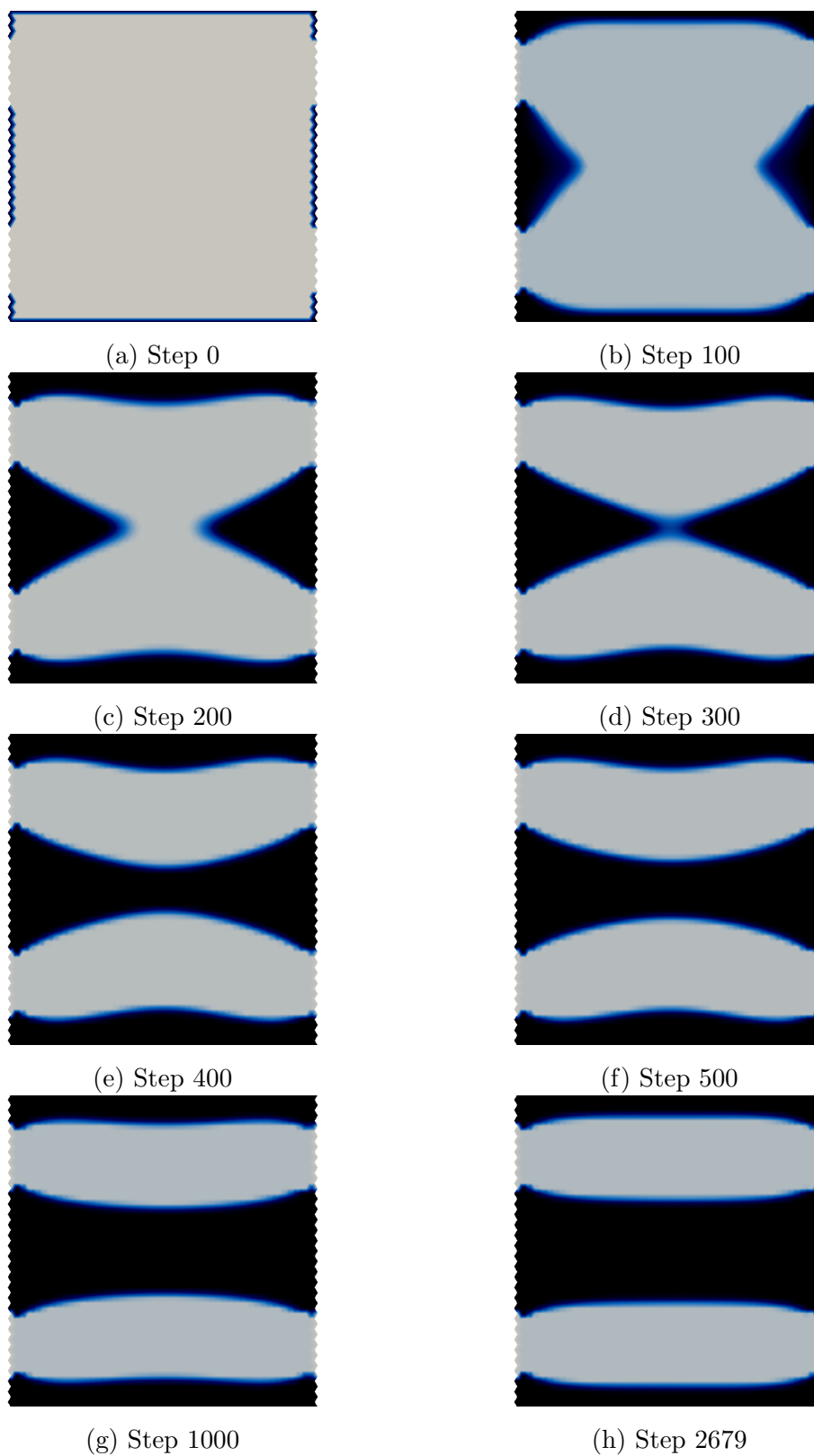


図 4.17. 二管問題ケース A の設計変数の変化.

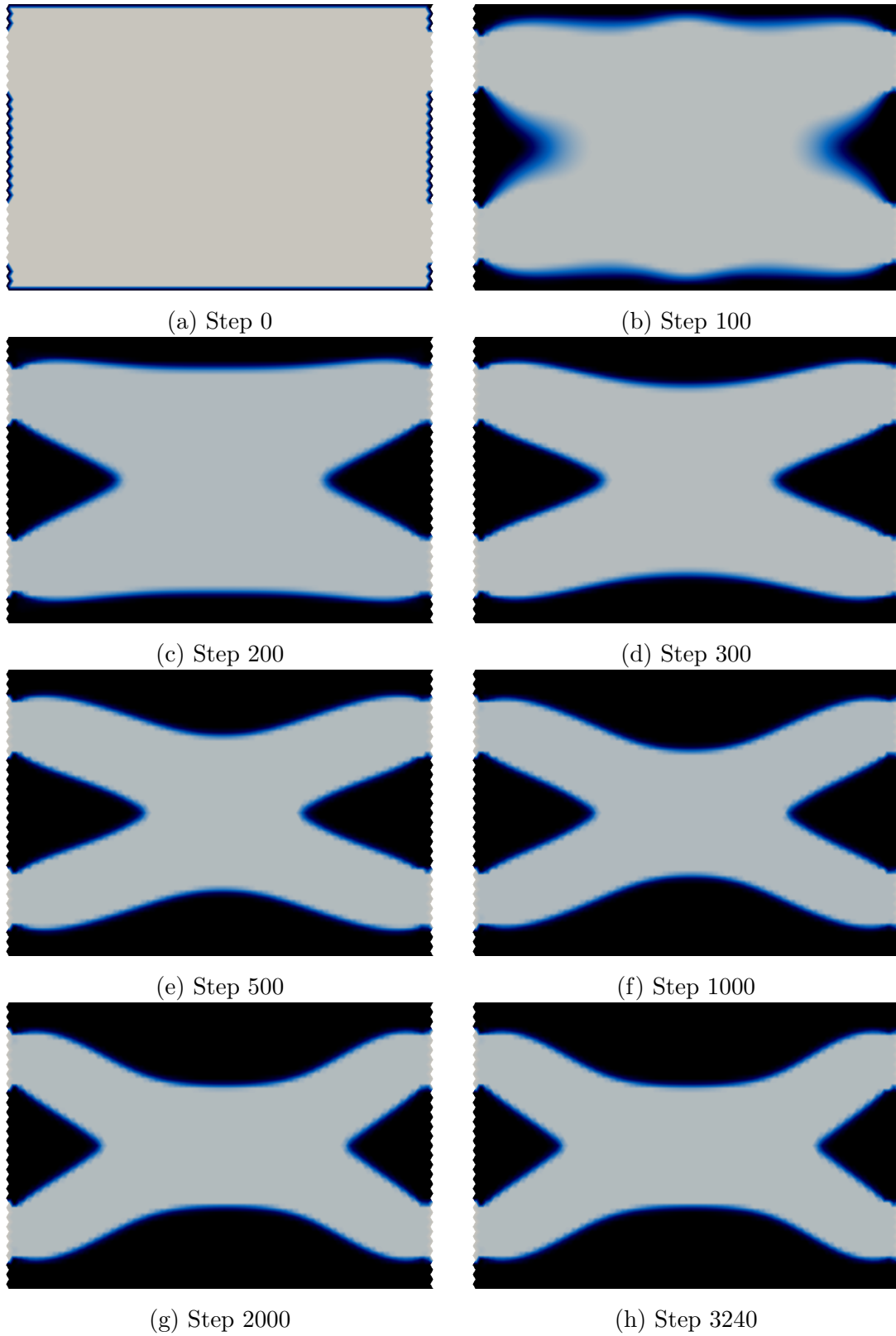
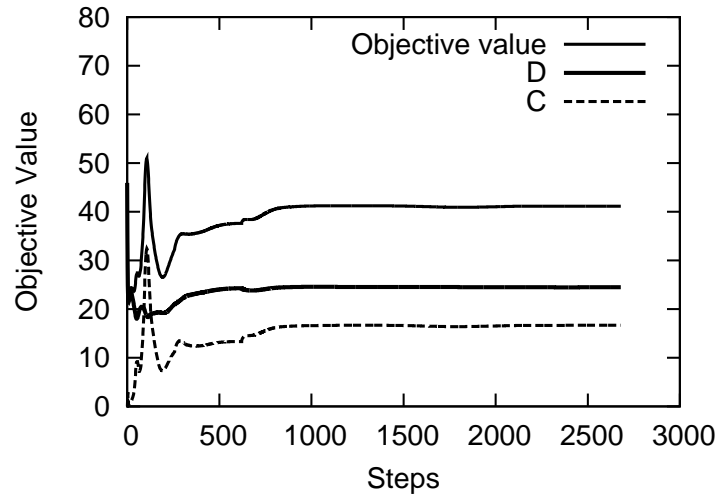
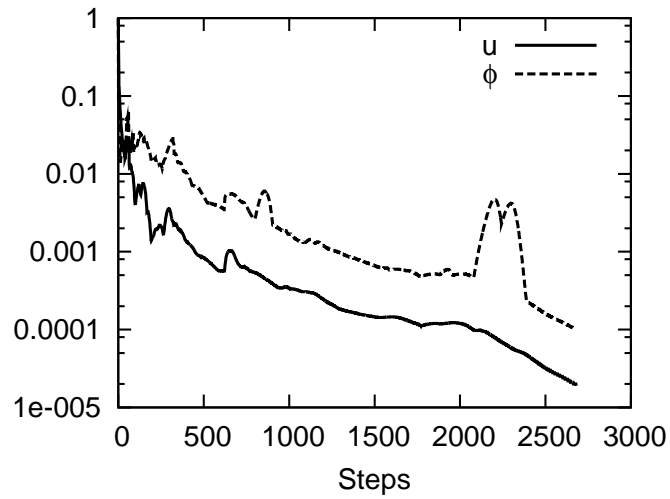


図 4.18. 二管問題ケース B の設計変数の変化.

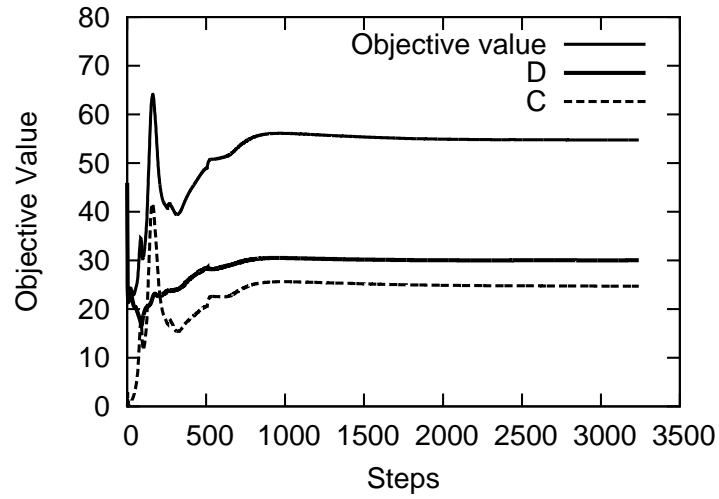


(a) 目的関数値.

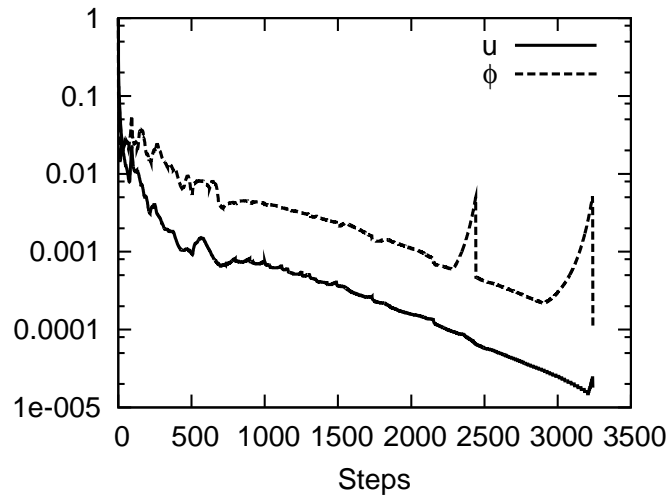


(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$.

図 4.19. 二管問題ケース A のの収束履歴



(a) 目的関数値.



(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ と $\|\phi_k - \phi_{k-1}\|_\infty$.

図 4.20. 二管問題ケース B の収束履歴

4.4 iSA 法と多孔質モデルを用いた場合のトポロジー最適化の結論

本章では多孔質媒体を用いて流体領域と固体領域を合わせて表現し、空隙率を設計変数として最適化を行った。まず Navier–Stokes 方程式を用いて最適化問題を記述し、その最適化問題を近似的に解くための手法を iSA 法に基づいて定式化した。そのうえで、手法の妥当性の検討のために五例の数値計算を行った。まず内部流れのベンチマーク問題として縮流管問題を対象とし、外部流れのベンチマークとしてラグビーボール問題を対象として最適化を行って既存手法との比較を行ったところ、結果は既存手法による結果とよく一致した。さらに手法のロバスト性を検討するために、トポロジーが明確に変化する合流管問題と速度境界を含む問題を対象として最適化を行い、いずれの例においても安定して最適解が得られることを確認した。以上により、手法自体の妥当性を確認することができた。

iSA 法の特徴の一つが、計算時間が短縮されることである。今回の数値計算では、格子点数 3172 点に対して計算時間は、2 秒から 12 秒程度であった。また、縮流管問題について格子点数を x, y の各方向にそれぞれ 2 倍に増やした場合、計算時間は約 8 倍に増加した。これは格子点数が 4 倍に増えたことに加え、格子解像度が 2 倍に増加したことに伴って時間ステップ幅が $1/2$ に細くなったため、計算ステップ数が 2 倍に増加したことで説明がつく。この結果より、 d 次元の流れ場の最適化では、幾何的なスケールが s 倍に大きくなった場合、計算時間は s^{d+1} に比例して長くなると言える。

一方で、数値計算の結果、流体領域と固体領域の境界が不鮮明であるという特徴も明らかになった。この特徴は先行研究 [7] でも見られ、多孔質媒体モデルを使用した場合には常に報告されている問題である。多孔質媒体を使用できれば、最適解に忠実な物体を製造できる可能性はある。しかしながら一般的な機械設計においては、金属やプラスチックなどの固体を使用するため、最適解を忠実に再現することは困難である。そのため、最適解の空隙率分布を元に、設計者の目分量や三次元の場合にはマーチングキューブス法 [42] を用いて形状を決定することになる。本章で扱った対象のように Reynolds 数が低い場合には、境界が不鮮明であることの流れ場に対する影響は小さいと考えられ、前述のように境界を決定しても流体力学的には流れ場には大きな問題にはならないと考えられる。しかしながら結果は異なることは自明であり、何らかの手段で解決する必要がある。

第 5 章

Newton 法を利用した高速化

第 3 章で提案した最適化手法では設計変数の更新方法として素朴な勾配法を用いており、収束速度が遅かった。本章では変数の更新方法の一部に Hesse 行列を利用して収束速度を早くする手法を提案する。数値計算例により収束速度の違いを示し、解の妥当性について述べる。

5.1 多孔質モデルの欠点と感度法の収束性

第 4 章では、多孔質媒体モデルに基づいた流れ場の最適化問題に iSA 法を適用した。その結果、短時間で最適解を求めることができた。しかしながら効率良く収束したとは言えず、特に最適解の近傍に近づいてからの収束が遅かった。最適解の近傍での収束性の悪さは固定ステップ幅を用いた最急降下法に一般的に言われることである。数値最適化手法では、直線探索を用いてステップ幅を随時変更する手法もあるが、この場合も収束性は一次収束である。しかし直線探索は目的関数を繰り返し計算してステップ幅を決定する必要がある。このとき目的関数の計算量が大きければ、繰り返し計算による計算量の増加が大きく、全体として計算量が増加する可能性がある。特に流れ場の最適化においては流れ場を繰り返し計算する必要があるため、計算量の負担が大きい。特に iSA 法では場の変数と設計変数を同時並行して更新するため、設計変数だけを優先して更新することになる直線探索は適さないと考えられる。

直線探索以外の局所的な収束速度を向上させる手法としては Newton 法が挙げられる。Newton 法は目的関数の二階微分を使用して変数を更新する方法で、二次収束することが知られている。純粋な Newton 法は制約のない凸関数最小化問題の解法として整備されている。第 4 章の記号をそのまま用いることとし、目的関数を $D + C$ 、変数を ϕ とすると、次の式で定義される Newton 方向に変数を更新する。

$$\mathbf{d}^n = -H_{D+C}^{-1} \nabla_{\phi} (D + C). \quad (5.1)$$

ただし H_{D+C} は $D + C$ の Hesse 行列である。Newton 方向は、目的関数を局所的に二次関数で近似し、その頂点に向けて変数を更新する手法であると言える。そのため非凸関数の最小化問題に適用する場合、Newton 方向が降下方向であることが保証されず、非凸関数に対しては Newton 法はそのままでは使用できない。そこで修正 Newton 法や準 Newton 法を用いる。

Newton 方向が少なくとも局所的に降下方向であるかどうかは、Hesse 行列の正定値性によって判別できる。修正 Newton 法では Hesse 行列が不定値である場合に、単位行列に Hesse 行列の最小固有値を乗じた行列を Hesse 行列から減算して、Hesse 行列を正定値化する。この操作は次式で定義される：

$$\mathbf{d}^{\text{modified}} = -H_D^{\text{modified}}{}^{-1} \nabla_{\phi}(D), \quad (5.2)$$

$$H_D^{\text{modified}} = H_D - \underline{\lambda}I, \quad (5.3)$$

$$\underline{\lambda} = \min \{0, \lambda_1, \lambda_2, \dots\}. \quad (5.4)$$

ただし λ_i は H_D の固有値である。修正 Newton 法は単純な手法であり、Hesse 行列の固有値をシフトするだけであるため、もとの Hesse 行列が疎行列であれば、修正後の Hesse 行列も疎行列であるという特徴がある。前述したもう一つの手法である準 Newton 法では、Hesse 行列の近似行列を作成する。近似行列を作る方法は BFGS 公式など [69] があるが、いずれも Hesse 行列の疎性を壊し、近似行列は一般に密行列になる。以上のように、Hesse 行列が疎行列である場合は、修正 Newton 法を用いた方が疎行列の良い性質を利用できる。

第4章で解いた流れ場の最適化問題は一般に非凸最適化問題である。そのため Hesse 行列は正定値行列とは限らない。また、後述するように iSA 法を用いた場合には Hesse 行列は対角行列になるという著しい特徴がある。本節では Newton 方向を利用した上で、修正 Newton 法のように、Hesse 行列の疎性を保存したまま更新方向を定義する。

また、第4章で得られた最適解では固体領域と流体領域の境界が曖昧で、グレーな領域が存在する。実用上は適当な位置、例えば $\phi = 0.5$ で流体領域と個体領域を分けることで、明確な境界を設定することができる。また、level-set 関数を用いて最適化問題を書き下し、境界を設定することもできる [2, 39, 79, 80]。しかしこのように人為的に境界を設定した場合には、流れ場の数値計算上は明確な境界がないにもかかわらず、最適化の結果としてはあたかも明確な境界が得られているように見える。これは実用上は問題ないことが多いものの、厳密には不正確である。そこで、特に収束の最終盤において、グレーな領域を減らすことが有用である。これは最適解の近傍での収束性に関係している。前述のように Newton 方向を利用すると、なかば強制的に ϕ を 0 か 1 に移動させ、グレーな領域が縮小し、明確な境界が得られる。このとき、流れ場の数値計算上の境界と視覚的な境界が一致するという利点がある。

本章は次のように構成されている。まず第5.2節で最適化手法を構築する。次に第5.3節で数値計算例を示す。その後第5.4節で解の最適性と収束性に関して考察する。最後に第5.5節で結論を述べる。

5.2 部分的に Newton 法を用いた最適化手法

5.2.1 部分的に Newton 法を用いた手法の定式化

第4章の手法では勾配法により最適化を行っている。このように勾配法を用いることは他手法でも多く行われている。一般に勾配法による ϕ の更新量 \mathbf{d}^g は (5.5) で定義される。

$$\mathbf{d}^g = -\kappa \nabla_{\phi}(D + C) \quad (5.5)$$

得られた最適解で ϕ が各点で 0 あるいは 1 に十分近ければ、領域を固体と流体に区別することが容易である。しかしながら、第5.3節で示すように、勾配法を用いた場合は ϕ が 0 にも 1 にも近くない点、すなわちグレーな点が存在する。これは、その点での勾配が小さく、また κ が固定であるために $\kappa \frac{\partial(D+C)}{\partial\phi}$ が小さいことに起因する。そこでグレーな点について、勾配法ではなく Newton 法に基づいて最適化を行う方法を提案する。まず、Newton 法では ϕ の更新量は次の式で定義される：

$$\mathbf{d}^n = -H_{D+C}^{-1} \nabla_{\phi}(D + C). \quad (5.6)$$

ただし、 H_{D+C} は次式で定義される Hesse 行列である：

$$(H_{D+C})_{ij} = \frac{\partial^2(D + C)}{\partial\phi(\mathbf{x}_i, t_k) \partial\phi(\mathbf{x}_j, t_k)}.$$

以降では \mathbf{u} と ϕ については時間と空間に関して離散化した (1.1) の定義を用いる。Newton 法は目的関数を局所的に二次関数として近似し、その二次関数の停留点へ移動する手法である。そのため Newton 法は Hesse 行列が正定値であれば降下方向を与えるが、そうでない場合には目的関数が増加する方向を与えることがあり得る。本稿の目的関数は非凸な関数であるため、Hesse 行列は必ずしも正定値とは限らない。非線形計画問題におけるこの種の問題を解決するために、準 Newton 法が用いられる。準 Newton 法では Hesse 行列の近似行列を作成するが、この行列が一般に密行列になる。そのため、 $\mathcal{O}(N^2)$ のメモリと $\mathcal{O}(N^2)$ の計算量が要求され、計算量が多い。そこで準 Newton 法は使用しない。

iSA 法では、後述するように、Hesse 行列は対角行列になる。以下では、この特徴を利用して少ない計算量と記憶容量で早い収束を実現する手法を提案する。つまり、 $0 < \phi(\mathbf{x}) < 1$ であるような $\phi(\mathbf{x})$ に対して、その点に対応する Hesse 行列の対角要素、すなわち固有値が正の場合はそのままにし、負である場合は勾配法に基づいて変数を更新する。(5.5) と (5.6) を比較すると、両者は勾配 $\nabla_{\phi}(D + C)$ に Hesse 行列の逆行列 H_{D+C}^{-1} を乗じるか、 κI を乗じるかの違いであることがわかる。先述の操作は、Hesse 行列の逆行列 H_{D+C}^{-1} の固有値が負の場合

に, その負固有値を κ に置き換えることに相当する. 以上の操作は次のように表される:

$$\phi_k(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi_{k-1}(\mathbf{x}) + d_k(\mathbf{x}) + M_K \leq 0, \\ \phi_{k-1}(\mathbf{x}) + d_k(\mathbf{x}) + M_K & \text{if } 0 < \phi_{k-1}(\mathbf{x}) + d_k(\mathbf{x}) + M_K < 1, \\ 1 & \text{if } 1 \leq \phi_{k-1}(\mathbf{x}) + d_k(\mathbf{x}) + M_K, \end{cases} \quad (5.7a)$$

$$d_k(\mathbf{x}) = -\widehat{H}_{D+C}^{-1} \nabla_{\phi_k}(D+C), \quad (5.7b)$$

$$\widehat{H}_{D+C} = \text{diag}(\widehat{\lambda}_1, \widehat{\lambda}_2, \dots), \quad (5.7c)$$

$$\widehat{\lambda}_i = \begin{cases} \lambda_i & \text{if } \lambda_i > 0, 0 < \phi_k(\mathbf{x}_i) - \frac{1}{\lambda_i} \nabla_{\phi_k}(\mathbf{x}_i)(D+C) < 1, \\ \frac{1}{\kappa} & \text{otherwise.} \end{cases} \quad (5.7d)$$

ただし $\text{diag}(\lambda_1, \lambda_2, \dots)$ は λ_i を対角成分に持つ対角行列とする. $-\widehat{H}_{D+C}^{-1} \nabla_{\phi}(D+C)$ は $D+C$ の降下方向であり, d_k は降下方向を与える. 前述のように, 最初は勾配法で最適化を行い, 途中から (5.7) を用いる. この切り替えの判断は, ϕ の増分 $\|\phi_k - \phi_{k-1}\|_{\infty}$ がある閾値以下になって大まかな形が決まった時点とする. 閾値は問題によって異なるが, 例えば 1.0×10^{-3} などの値を用いる.

5.2.2 H_{D+C} の計算方法

本節では (5.7) で使用する H_{D+C} と $\nabla_{\phi}(D+C)$ を導く. 最初に, 格子 Boltzmann 方程式における流速の定義を

$$\rho(\mathbf{x}, t_k) = \sum_i f_i^{\text{col}}(\mathbf{x}, t_k), \quad (5.8a)$$

$$\mathbf{u}(\mathbf{x}, t_k) = \frac{\widehat{\mathbf{u}}}{1 + 0.5\alpha}, \quad \widehat{\mathbf{u}} = \frac{1}{\rho} \sum_i f_i^{\text{col}}(\mathbf{x}, t_k) \mathbf{c}_i \quad (5.8b)$$

に変更する. (2.12) と (5.8) の違いは, 移流操作の後の流速を採用するか, 衝突操作の後の流速を採用するかの違いである. まず $\nabla_{\phi(\mathbf{x})}(D+C)$ は, 連鎖側より

$$\nabla_{\phi_k(\mathbf{x})}(D+C) = \frac{\partial(D+C)_k}{\partial \mathbf{u}_k} \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} + \frac{\partial(D+C)_k}{\partial \phi_k(\mathbf{x})} \quad (5.9)$$

と分解される. 右辺の各項は (4.1), (4.2) より,

$$\frac{\partial(D+C)_k}{\partial \mathbf{u}_k} = \left(-\nu \Phi_k^p \Delta \mathbf{u}_k(\mathbf{u}_{k-1}, \phi_{k-1}) \delta r + 2\alpha \mathbf{u}_k(\mathbf{u}_{k-1}, \phi_{k-1}) \delta r \right)^{\top}, \quad (5.10)$$

$$\frac{\partial(D+C)_k}{\partial \phi_k(\mathbf{x})} = \frac{p}{2} \phi_k(\mathbf{x})^{p-1} \nu \|\nabla \mathbf{u}_k(\mathbf{x}) + (\nabla \mathbf{u}_k(\mathbf{x}))^{\top}\|^2 \delta r + \frac{\partial \alpha}{\partial \phi_k(\mathbf{x})} \|\mathbf{u}_k(\mathbf{x})\|^2 \delta r \quad (5.11)$$

で与えられる. ただし Φ_k^p は $\Phi_k^p = \text{diag}(\phi_k(\mathbf{x}_1)^p, \phi_k(\mathbf{x}_1)^p, \phi_k(\mathbf{x}_2)^p, \dots, \phi_k(\mathbf{x}_N)^p)$ であり, $\phi_k(\mathbf{x}_i)$ を, 次元の数である 2 個ずつ並べた対角行列である. ここで, (5.10) において次の近似を用いる:

$$\frac{\partial(D+C)_k}{\partial \mathbf{u}_k} \simeq (-\nu \Phi_k^p \Delta \mathbf{u}_{k-1} + 2\alpha \mathbf{u}_{k-1})^{\top}. \quad (5.12)$$

この近似は最適解が $\mathbf{u}_k(\mathbf{u}_{k-1}, \phi_k) = \mathbf{u}_{k-1}(\mathbf{u}_{k-2}, \phi_{k-1})$ を満たすことから正当化される.

次に (5.9) の右辺の $\frac{\partial \mathbf{u}_k}{\partial \phi_k}$ は, 再び連鎖側により次のように分解される:

$$\frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} = -\frac{0.5}{1 + 0.5\alpha} \frac{\partial \alpha}{\partial \phi_k(\mathbf{x})} \hat{\mathbf{u}} + \frac{1}{1 + 0.5\alpha} \sum_{i=0}^6 \frac{\partial \hat{\mathbf{u}}_k}{\partial f_i^{\text{col}}(\mathbf{x}, t_k)} \frac{\partial f_i^{\text{col}}(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})}. \quad (5.13)$$

定義 (5.8b) より

$$\frac{\partial \hat{\mathbf{u}}}{\partial f_i^{\text{col}}} = \frac{\mathbf{c}_i - \hat{\mathbf{u}}}{\rho} \quad (5.14)$$

が得られる. (5.8) と格子 Boltzmann 方程式 (2.4), (2.10), (2.11) より次の式が得られる:

$$\frac{\partial f_i^{\text{col}}(\mathbf{x}, t_{k+1})}{\partial \phi_k(\mathbf{x})} = \frac{1}{\tau} \frac{\partial f_i^{\text{eq}}(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})} + \delta t \frac{\partial F_i(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})}, \quad (5.15)$$

$$\frac{\partial f_i^{\text{eq}}(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})} = -\omega_i \rho \frac{(\mathbf{u}_k \otimes \mathbf{u}_k) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \frac{1}{\phi_k(\mathbf{x})^2}, \quad (5.16)$$

$$\frac{\partial F_i(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})} = -\omega_i \rho \left(1 - \frac{1}{2\tau}\right) \left[\frac{\mathbf{c}_i \cdot \mathbf{u}_k}{c_s^2} \frac{\partial \alpha_k}{\partial \phi_k(\mathbf{x})} \right. \quad (5.17)$$

$$\left. + \frac{(\mathbf{u}_k \otimes \mathbf{u}_k) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \frac{\partial}{\partial \phi_k(\mathbf{x})} \left(\frac{\alpha_k}{\phi_k(\mathbf{x})} \right) \right]. \quad (5.18)$$

以上で $D + C$ の一階微分 $\nabla_\phi(D + C)$ が得られた. 次に二階微分項を導く. 速度ベクトルに関して, $\phi_k(\mathbf{x})$ が影響を与えるのは $\mathbf{u}_k(\mathbf{x})$ だけであり, 他の格子点には影響を与えない. 実際, (5.13)–(5.18) により $\frac{\partial \mathbf{u}_k(\mathbf{x})}{\partial \phi_k(\mathbf{x})}$ が与えられるが, この一連の式の中には $\phi_k(\mathbf{x})$ しか現れず, 他の格子点の $\phi_k(\mathbf{x} + \mathbf{e}_i)$ で微分しても結果は零である. 目的関数についても, 定義より, $(D + C)_k(\mathbf{x})$ 中には $\phi_k(\mathbf{x})$ しか現れず, 他の格子点の $\phi_k(\mathbf{x} + \mathbf{e}_i)$ で微分しても結果は零である. したがって, Hesse 行列は対角行列である. そこで, Hesse 行列の非対角項については考慮せず, 対角項に関してのみ計算を実施すると, 次のようになる:

$$\begin{aligned} \frac{\partial^2 (D + C)_k}{\partial \phi_k(\mathbf{x})^2} &= \left(\frac{\partial^2 (D + C)_k}{\partial \mathbf{u}_k^2} \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} + \frac{\partial}{\partial \phi_k(\mathbf{x})} \frac{\partial (D + C)_k}{\partial \mathbf{u}_k} \right)^\top \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} \\ &+ \left(\frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} \right)^\top \left(\frac{\partial}{\partial \mathbf{u}_k} \frac{\partial (D + C)_k}{\partial \phi_k(\mathbf{x})} \right) \\ &+ \frac{\partial^2 (D + C)_k}{\partial \phi_k(\mathbf{x})^2} + \frac{\partial (D + C)_k}{\partial \mathbf{u}_k} \frac{\partial^2 \mathbf{u}_k}{\partial \phi_k(\mathbf{x})^2}. \end{aligned}$$

続いて各項は次のようになる．ここで，(5.12)の近似を用いる．

$$\begin{aligned} \left(\frac{\partial^2(D+C)_k}{\partial \mathbf{u}_k^2} \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} \right)^\top \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} &= -\phi_k(\mathbf{x})^p \nu \left\langle \nabla \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})}, \nabla \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} \right\rangle + 2\alpha \left\| \frac{\partial \mathbf{u}_k}{\partial \phi_k(\mathbf{x})} \right\|^2, \\ \frac{\partial}{\partial \phi_k(\mathbf{x})} \frac{\partial(D+C)_k}{\partial \mathbf{u}_k} &\simeq \left(-p\phi_k(\mathbf{x})^{p-1} \nu \Delta \mathbf{u}_k + 2 \frac{\partial \alpha}{\partial \phi_k(\mathbf{x})} \mathbf{u}_k \right)^\top, \\ \frac{\partial^2(D+C)_k}{\partial \phi_k(\mathbf{x})^2} &\simeq \frac{p}{2} (p-1) \phi_k(\mathbf{x})^{p-2} \nu \|\nabla \mathbf{u}_k\|^2 + \frac{\partial^2 \alpha}{\partial \phi_k(\mathbf{x})^2} \|\mathbf{u}_k\|^2, \\ \frac{\partial}{\partial \mathbf{u}_k} \frac{\partial(D+C)_k}{\partial \phi_k(\mathbf{x})} &\simeq \left(-p\phi_k(\mathbf{x})^{p-1} \nu \Delta \mathbf{u}_k + 2 \frac{\partial \alpha}{\partial \phi_k(\mathbf{x})} \mathbf{u}_k \right)^\top. \end{aligned}$$

最後に，以下の式を用いて $\frac{\partial^2 \mathbf{u}}{\partial \phi_k^2}$ を求めることができる．

$$\begin{aligned} \frac{\partial^2 \mathbf{u}}{\partial \phi_k(\mathbf{x})^2} &= \left(\frac{\partial^2}{\partial \phi_k(\mathbf{x})^2} \frac{1}{1+0.5\alpha} \right) \hat{\mathbf{u}} + 2 \left(\frac{\partial}{\partial \phi_k(\mathbf{x})} \frac{1}{1+0.5\alpha} \right) \frac{\partial \hat{\mathbf{u}}}{\partial \phi_k(\mathbf{x})} + \frac{1}{1+0.5\alpha} \frac{\partial^2 \hat{\mathbf{u}}}{\partial \phi_k(\mathbf{x})^2}, \\ \frac{\partial^2 \hat{\mathbf{u}}}{\partial \phi_k(\mathbf{x})^2} &= \sum_{i=0}^6 \left(\frac{\partial \mathbf{c}_i - \mathbf{u}}{\rho} \frac{\partial f_i^{\text{col}}}{\partial \phi_k(\mathbf{x})} + \frac{\mathbf{c}_i - \mathbf{u}}{\rho} \frac{\partial^2 f_i^{\text{col}}}{\partial \phi_k(\mathbf{x})^2} \right), \\ \frac{\partial \mathbf{c}_i - \mathbf{u}}{\rho} &= -\frac{1}{\rho} \frac{\partial \mathbf{u}}{\partial \phi_k(\mathbf{x})} - \frac{\mathbf{c}_i - \mathbf{u}}{\rho^2} \sum_{i=0}^6 \frac{\partial f_i^{\text{col}}}{\partial \phi_k(\mathbf{x})}, \\ \frac{\partial^2 f_i^{\text{col}}}{\partial \phi_k(\mathbf{x})^2} &= \frac{1}{\tau} \frac{\partial^2 f_i^{\text{eq}}}{\partial \phi_k(\mathbf{x})^2} + \delta t \frac{\partial^2 F_i}{\partial \phi_k(\mathbf{x})^2}, \\ \frac{\partial^2 f_i^{\text{eq}}(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})^2} &= -\omega_i \frac{(\mathbf{u}_k \otimes \mathbf{u}_k) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I}) \sum \frac{\partial f_i}{\partial \phi_k(\mathbf{x})} \phi_k(\mathbf{x}) - 2\rho}{2c_s^4 \phi_k(\mathbf{x})^3}, \\ \frac{\partial^2 F_i(\mathbf{x}, t_k)}{\partial \phi_k(\mathbf{x})^2} &= -\omega_i \left(1 - \frac{1}{2\tau} \right) \left[\frac{\mathbf{c}_i \cdot \mathbf{u}_k}{c_s^2} \frac{\partial}{\partial \phi_k(\mathbf{x})} \left(\rho \frac{\partial \alpha_k}{\partial \phi_k(\mathbf{x})} \right) \right. \\ &\quad \left. + \frac{(\mathbf{u}_k \otimes \mathbf{u}_k) : (\mathbf{c}_i \otimes \mathbf{c}_i - c_s^2 \mathbf{I})}{2c_s^4} \frac{\partial}{\partial \phi_k(\mathbf{x})} \left(\rho \frac{\partial}{\partial \phi_k(\mathbf{x})} \left(\frac{\alpha_k}{\phi_k(\mathbf{x})} \right) \right) \right]. \end{aligned}$$

以上を用いて $\nabla_{\phi(\mathbf{x})}(D+C)$ および H_{D+C} を計算し，(5.7)を用いて最適化を行う． $\nabla_{\phi(\mathbf{x})}(D+C)$ と H_{D+C} のどちらも代入計算だけで計算が可能であり，計算コストは小さい．最後に，以上の計算方法をアルゴリズム4にアルゴリズムとして記述する．

アルゴリズム 4.

- Step 1. $k=0$ とし，適当な値を用いて \mathbf{u}_0 と ϕ_0 を初期化する．
- Step 2. $\phi = \phi_k$ として \mathbf{u}_k を計算する．
- Step 3. 勾配法による更新量 \mathbf{d}^{S} を (5.5) により計算する
- Step 4. k ステップ目までに $\|\phi_k - \phi_{k-1}\|_\infty \leq 1.0 \times 10^{-3}$ を満たしていなければ Step 5(a) を，そうでなければ Step 5(b) を実行する．
- Step 5(a). (5.7a) で $\mathbf{d}_k = \mathbf{d}^{\text{S}}$ として ϕ_{k+1} を計算する．
- Step 5(b). ϕ_{k+1} を (5.7) により計算する．

Step 6. 収束判定を満たしていれば停止.

Step 7. $k = k + 1$ として Step 2 へ.

アルゴリズムの収束判定に用いる閾値は問題によって異なるが, 本論文の例題では $\|\delta\phi\|_\infty \leq 1.0 \times 10^{-5}$ を用いる.

5.3 数値計算例

本節では提案手法を用いた数値計算例を示す. 使用した計算機は XEON X5690 (3.33GHz with 8GB memory) であり, C++ により実装した. 同じ問題を提案手法で解き, それに加えて勾配法のみでも解いて両者を比較した.

5.3.1 縮流管の最適化

本節では縮流管のトポロジー最適化を行う. 設計空間と境界条件を図 5.1 に示す. 入口と出口は流速を次のように指定する:

$$U_{\text{in}}(y) = -\bar{U}y(y - L),$$

$$U_{\text{out}}(y) = \begin{cases} -3\bar{U}(y - \frac{L}{3})(y - \frac{2L}{3}) & \text{for } y \in \{y \mid \frac{L}{3} \leq y \leq \frac{2L}{3}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{L}{3}, \frac{2L}{3} \leq y\}. \end{cases}$$

Reynolds 数は 10.0 とした. 格子数は x 方向に 52 点, y 方向に 61 点とし, 格子点数は計 3162 点である. 体積制約は $\bar{V} = 0.5|\Omega|$ とした.

収束の履歴を図 5.2 および図 5.3 に示す. 提案手法を用いる場合は, 最初は勾配法で解き 250 ステップ目で $\|\phi_k - \phi_{k-1}\|_\infty \leq 1.0 \times 10^{-3}$ を満たしたため, それ以降で提案手法を適用した. 図 5.3(a) を見ると, 提案手法に切り替えた瞬間は, それまで一定であった更新幅が大きく変わるため, ϕ の更新量が増えることがわかる. しかしすぐに更新幅は小さくなり, 約 389 ステップで ϕ の更新幅が 1.0×10^{-5} を下回った. これに対し勾配法を使い続けた場合は, ϕ の更新量は順調に小さくなるものの, 提案手法よりは収束が遅い. 勾配法のみの場合には計算を 1000 ステップで打ち切った. 一方で図 5.3(b) に示す \mathbf{u} の収束速度は, 提案手法の方が遅い. これは, 勾配法のみの場合には $0 < \phi < 1$ の領域が広く, その領域での摩擦散逸がダンパーの役割を果たして収束性を向上させているためと考えられる. 次に図 5.2(a) に示す目的関数の値を見ると, 提案手法に切り替えたところで大幅に下がることがわかる. 図 5.2(b) に, D および C のそれぞれの値を示す. C は大幅に小さくなるものの, D は逆に大きくなっていることがわかる. これは次に示すようにグレースケールの領域が小さくなることから説明できる.

図 5.4 と図 5.5 に提案手法と勾配法を用いた最適化の過程における ϕ の分布をそれぞれ示す. ただし, 図 5.4(a)–(d) では勾配法だけを用いている. 図 5.4(f) が提案手法の最適解であり, (h) が勾配法を用いた場合の最適解である. 提案手法に切り替えた場合にはグレースケールの部分が少なくなり, 境界がくっきりと分かっている様が見て取れる. グレースケールの面積が大きい場合には C が大きくなるため, 提案手法によりグレースケールの領域が減った分だけ, 目的関数の値も小さくなる. 逆に, グレースケールの領域の付近では速度勾配が小さ

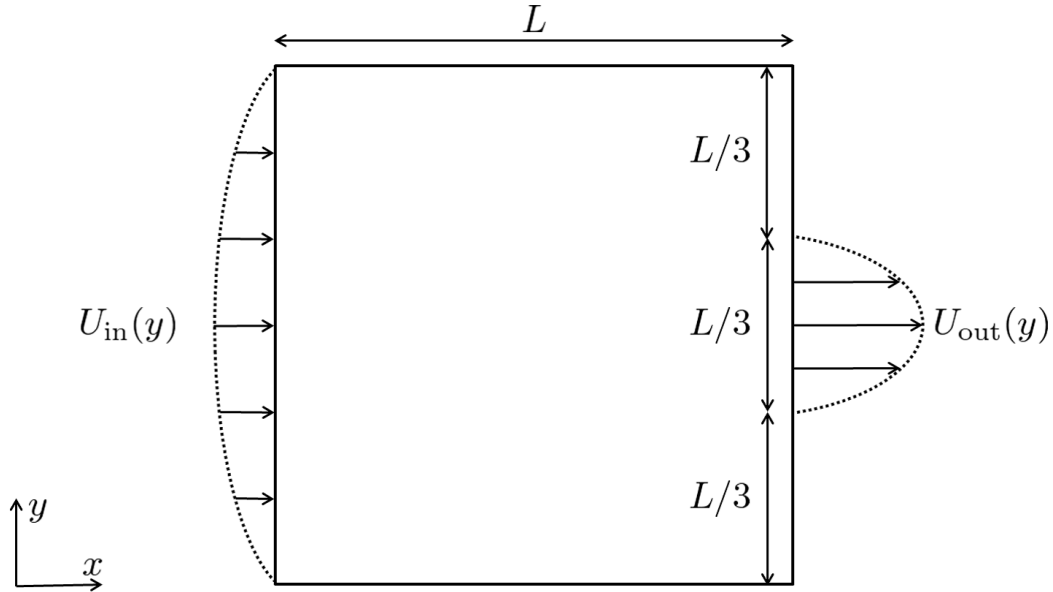


図 5.1. 縮流管問題の設計領域と境界条件.

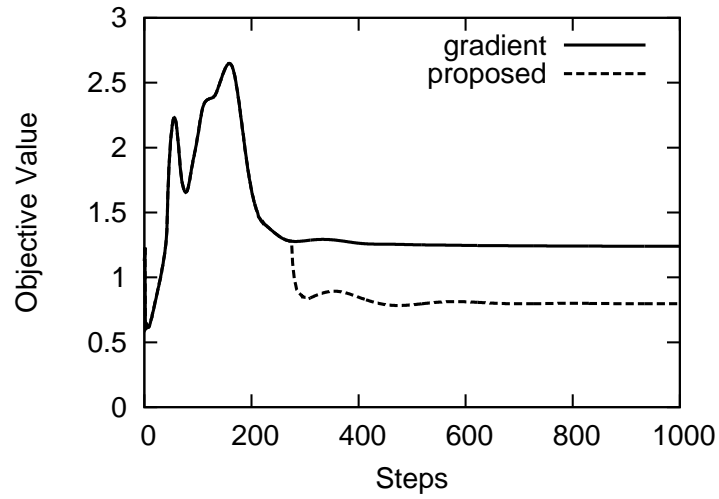
い。これに対し，グレースケールの領域が小さくなると， ϕ の勾配が大きくなるため速度勾配も大きくなる．その結果として D の値が大きくなる．

提案手法と勾配法的最適解（図 5.4(f) と (h)）を比べると，両者の概形は同じである．これらの最適解は先行文献 [7] とおよそ一致しており，手法の妥当性が確認できる．提案手法を用いた場合は，流体と固体の境界が明確になる反面，格子に起因する凹凸が明確に見られるようになる．数値計算上の流れ場の境界面は，この視覚的な境界と一致している．しかしながら，製品を設計する際には，通常は滑らかな形状で設計するため，凹凸が目立つ結果は感覚的に受け入れにくい可能性がある．そのため，別途境界面を滑らかにするなどの操作を行う必要がある．一方で，最後まで勾配法を用いた場合には境界は滑らかであり，およその形状の見当を付けることはできるため，設計者にとって感覚的に受け入れやすい可能性がある．しかしながら，実際に形状を決める際には，境界面をどこにおいてよいか明確でなく，決定した形状に数理的な保証を付けることが困難になる．

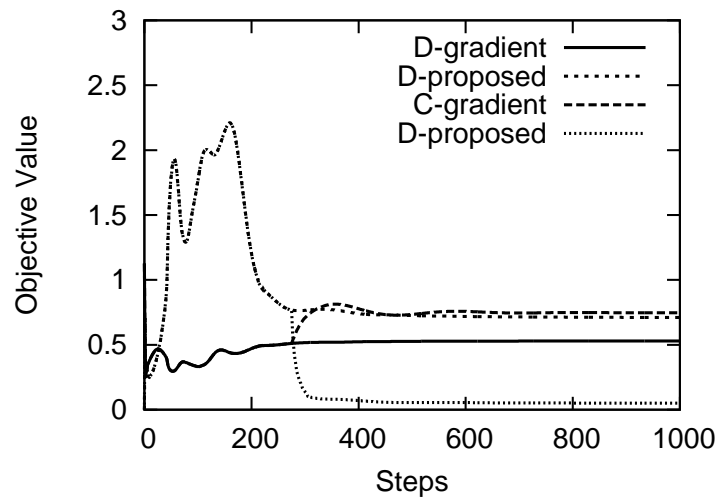
計算を止めた時点のステップ数と計算時間を表 5.1 に示す．提案手法を用いることで収束速度が向上し，計算時間が短縮される．

表 5.1. 縮流管問題の計算時間の比較.

	提案手法	勾配法
ステップ数	389	1000
計算時間 (秒)	1.94	3.66

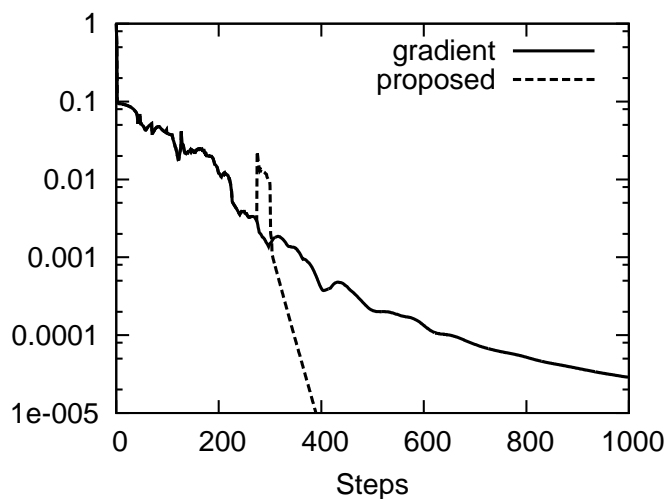


(a) 目的関数値.

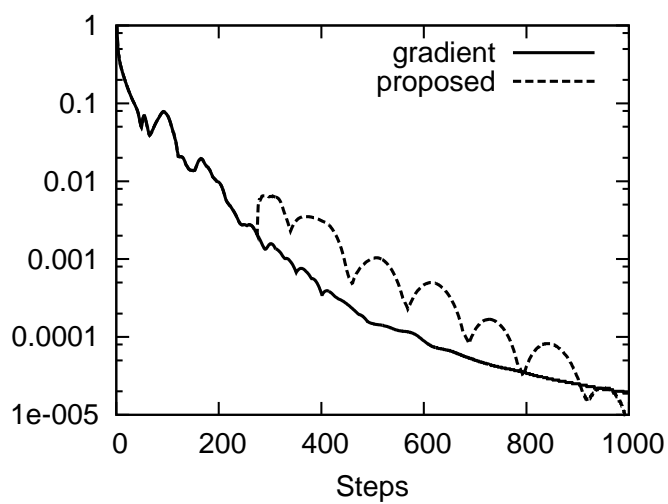


(b) D と C の値.

図 5.2. 縮流管問題の収束の履歴.



(a) $\|\phi_k - \phi_{k-1}\|_\infty$ の履歴.



(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ の履歴.

図 5.3. 縮流管問題の変数の収束の履歴.

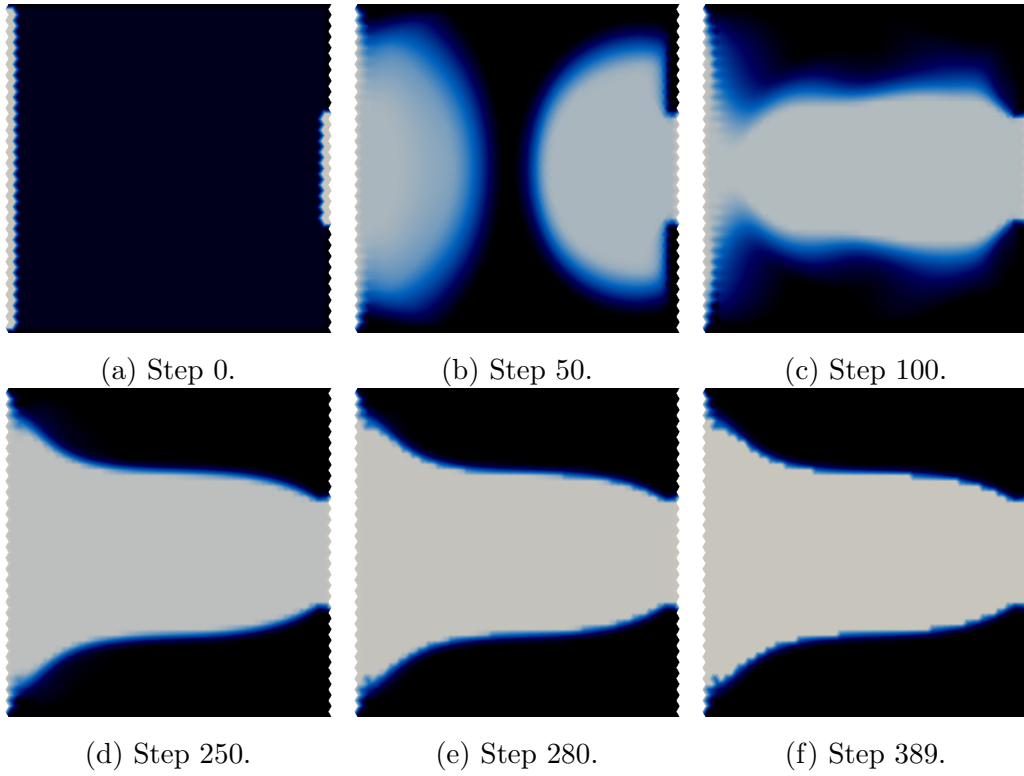


図 5.4. 縮流管問題の設計変数の変化の様子.

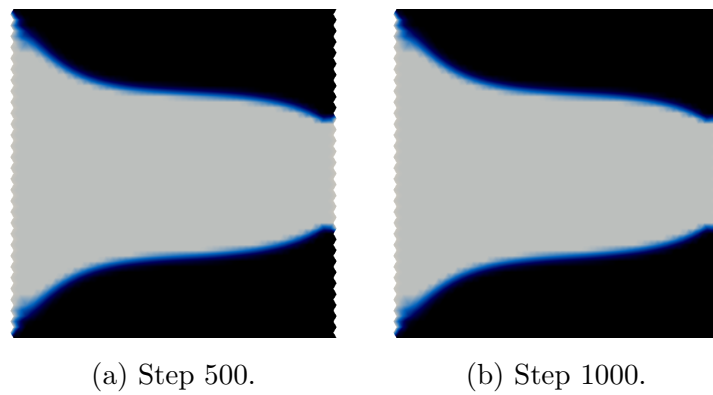


図 5.5. 勾配法を用いた場合の縮流管問題の設計変数の変化の様子.

5.3.2 合流管の最適化

本節では二か所の入口と一か所の出口を持つ合流管の最適化問題を解く．設計空間と境界条件は図 5.6 に示す．入口と出口の速度は次のように指定した：

$$\begin{aligned}
 U_{\text{in1}}(y) &= \begin{cases} -\bar{U}(y - \frac{3L}{8})(y - \frac{5L}{8}) & \text{for } y \in \{y \mid \frac{3L}{8} \leq y \leq \frac{5L}{8}\}, \\ 0 & \text{for } y \in \{y \mid y \leq \frac{3L}{8}, \frac{5L}{8} \leq y\}, \end{cases} \\
 U_{\text{in2}}(x) &= \begin{cases} -0.5\bar{U}(x - \frac{L}{2})(x - \frac{3L}{4}) & \text{for } x \in \{x \mid \frac{L}{2} \leq x \leq \frac{3L}{4}\}, \\ 0 & \text{for } x \in \{x \mid x \leq \frac{L}{2}, \frac{3L}{4} \leq x\}, \end{cases} \\
 U_{\text{out}}(x) &= \begin{cases} -1.5\bar{U}(x - \frac{L}{2})(x - \frac{3L}{4}) & \text{for } x \in \{x \mid \frac{L}{2} \leq x \leq \frac{3L}{4}\}, \\ 0 & \text{for } x \in \{x \mid x \leq \frac{L}{2}, \frac{3L}{4} \leq x\}. \end{cases}
 \end{aligned}$$

Reynolds 数は 1.0 とした．格子数は x 方向に 52 点， y 方向に 61 点の合計 3162 点である．体積制約は $\bar{V} = 0.35|\Omega|$ とした．

収束の履歴を図 5.7 および図 5.8 に示す．提案手法を用いる場合は，最初は勾配法で解き 581 ステップ目で $\|\phi_k - \phi_{k-1}\|_\infty \leq 1.0 \times 10^{-3}$ を満たしたため，それ以降で提案手法を適用した．収束の履歴は縮流管の場合と同様の傾向を示す．図 5.8(a) を見ると，提案手法に切り替えた瞬間は， ϕ の更新量が増え， ϕ の分布が大きく変わるものの，すぐに更新幅は小さくなり，約 544 ステップで ϕ の更新幅が 1.0×10^{-6} を下回った．これに対し勾配法を使い続けた場合は， ϕ の収束速度は遅い．また，図 5.8(b) に示す \mathbf{u} の収束は，提案手法の方が早い．これは ϕ の収束が早いために形状が固定され， \mathbf{u} が収束しやすくなったためと考えることができる．次に図 5.7(a) に示す目的関数の値も，提案手法に切り替えたところで大きく小さくなる．図 5.7(b) に， D および C のそれぞれの値を示す．また，図 5.9 と図 5.9 に提案手法と勾配法を用いた最適化の過程における ϕ の分布をそれぞれ示す．ただし図 5.9(a)–(d) は勾配法を用いた部分の結果である．縮流管の場合と同様，提案手法に切り替えるとグレースケールの領域が小さくなるのがわかる．それに伴って C が大幅に小さくなり， D は逆に大きくなる．

計算を止めた時点のステップ数と計算時間を表 5.2 に示す．提案手法では，勾配法のみの場合の約半分の時間で収束することがわかる．以上のように，提案手法を用いることで収束速度が向上し，境界も明確になることがわかる．

表 5.2. 合流管問題の計算時間の比較

	提案手法	勾配法
ステップ数	544	2000
計算時間 (秒)	2.72	10.17

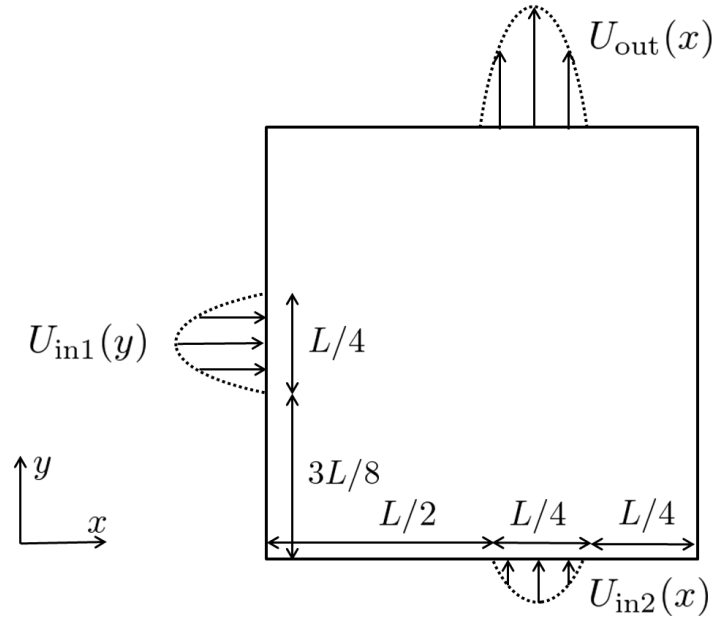
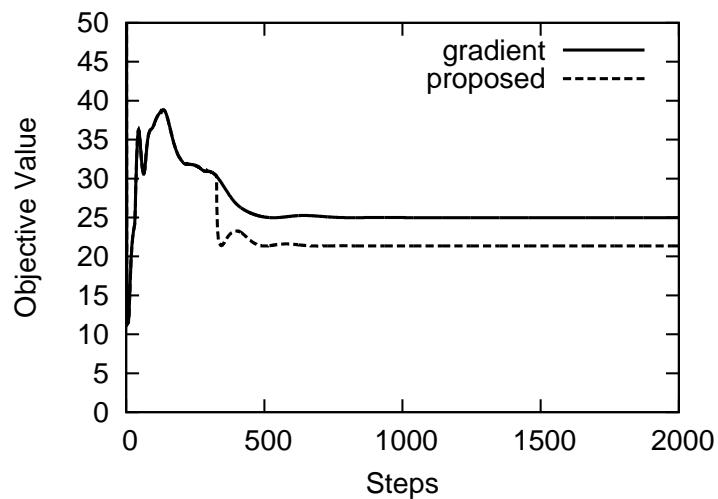
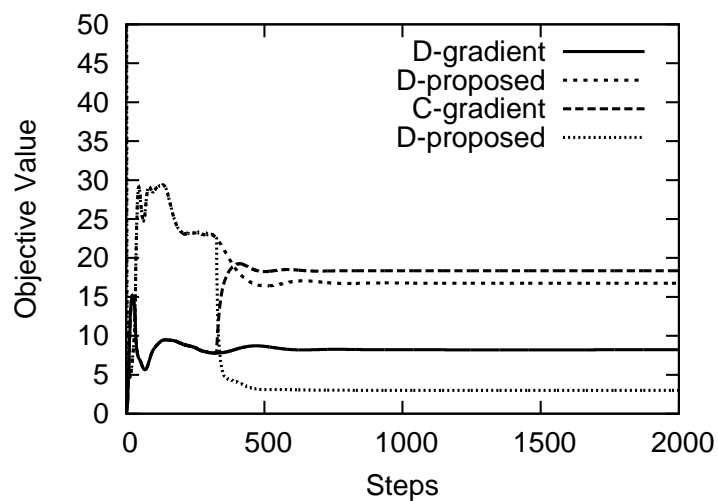


図 5.6. 合流管問題の設計領域と境界条件.



(a) 目的関数値.



(b) D と C の値.

図 5.7. 集合管問題の目的関数の収束の履歴.

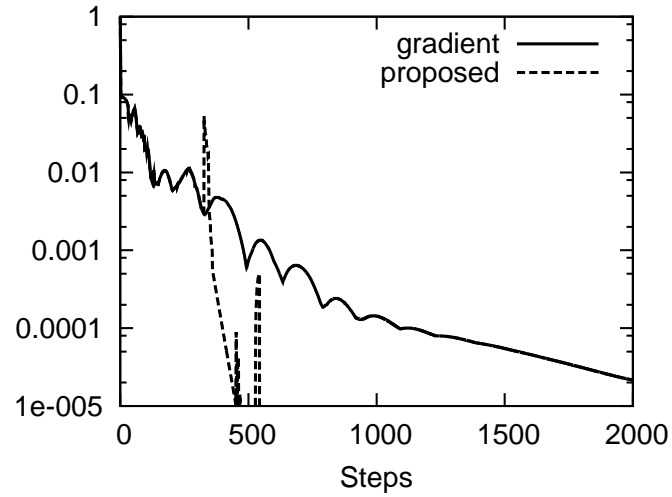
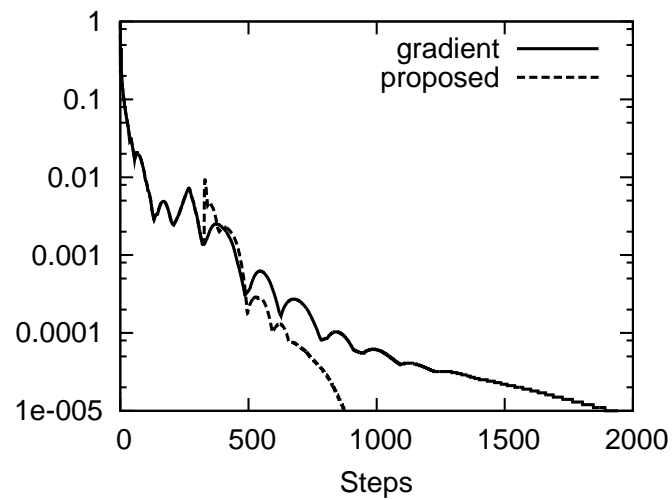
(a) $\|\phi_k - \phi_{k-1}\|_\infty$ の収束の履歴.(b) $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_\infty$ の収束の履歴.

図 5.8. 合流管問題の変数の収束の履歴.

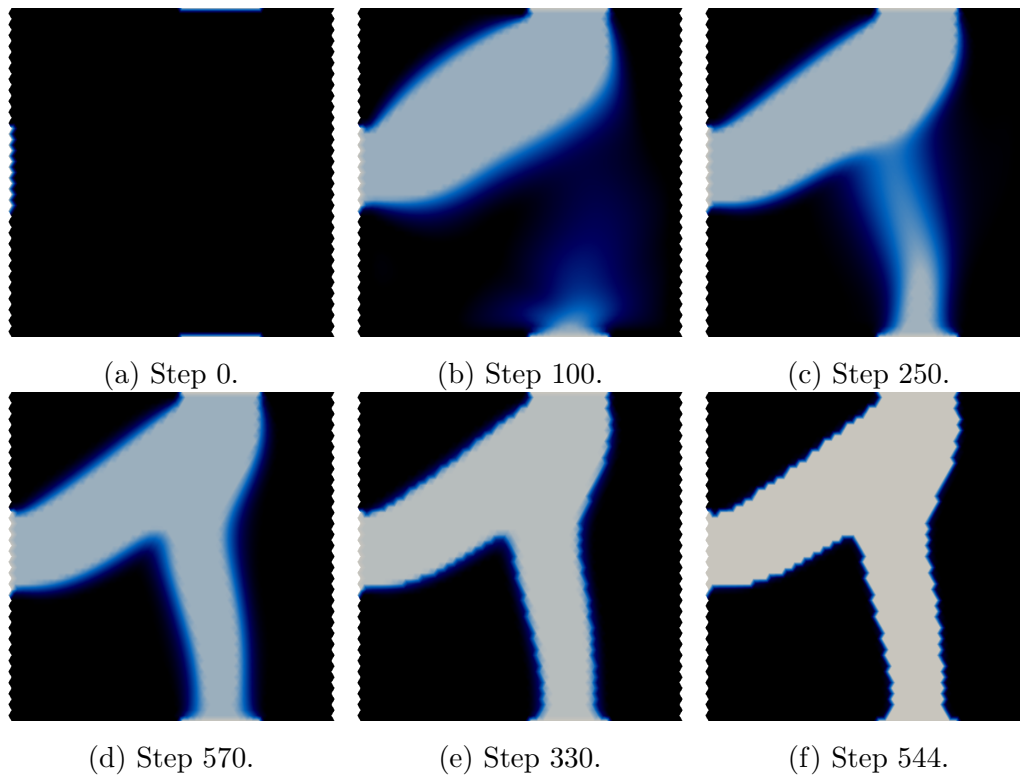


図 5.9. 合流管問題の設計変数の変化の様子.

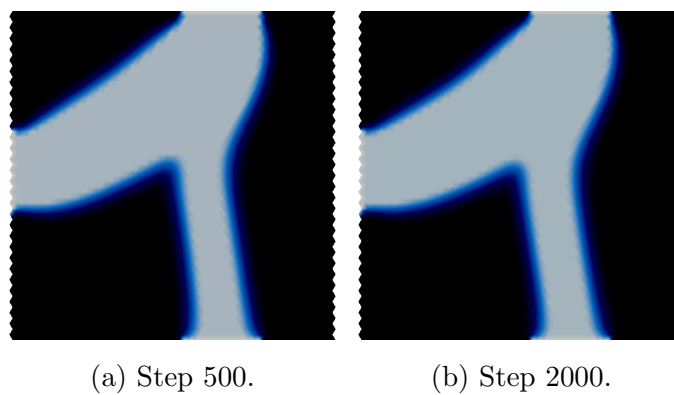


図 5.10. 勾配法を用いた場合の合流管問題の設計変数の変化の様子.

5.4 解の最適性と収束速度に関する議論

本節では、解の最適性の観点から、グレースケール問題と提案手法の関係を述べる。まず、提案手法の ϕ の更新方向は降下方向であり、第 5.3 節の数値実験では、提案手法の方が勾配法の場合よりも目的関数の値が小さくなった。この結果からは、勾配法でも時間をかければ提案手法と同じ解に到達できるとは言えない。iSA 法で使用する感度はその時刻での瞬時的な感度である。この感度は、その時刻だけを見れば降下方向であるが、 \mathbf{u} が変化したときに降下方向であることは保証されない。従って、 ϕ の更新方向を計算した時にそれが降下方向であることが保証されていたとしても、次の時刻では降下方向ではない可能性がある。実際、図 5.2(a) や図 5.7(a) の 200 ステップ目以前を見ると、降下方向に進んでいるにもかかわらず、目的関数は増大している。ここで改めて数値計算結果を見ると、提案手法に切り替えた途端に、それまでグレーであった領域で ϕ が 0 や 1 に変化する領域がある。ニュートン方向を利用することで、勾配法に比べて更新幅を大きくとることができ、 ϕ が大きく更新されている。特にこの領域で \mathbf{u} が大きく変化するため、前述のように ϕ の更新方向は、時間が経って \mathbf{u} が大きく変化した場合に降下方向であることは保証されない。従って、今回の数値計算では目的関数が下がったものの、実際には目的関数が小さくならない場合があり得る。今回の数値実験結果も、目的関数が大きく減少していることを考えると、勾配法での収束先とは別の局所最適解に到達している可能性がある。

このように、提案手法はニュートン方向を利用して ϕ を半ば強制的に 0 または 1 に変化させ、別の局所最適解に到達させている可能性があるため、目的関数が増加する可能性がある。最適化アルゴリズムの観点からは、収束に近づいたある局所解を外れて、別の局所解に収束することは不自然である。しかしながら、工学問題としては明確な部品形状を設計するために、目的関数の値よりも形状を明確にすることが優先される場合がある。例えば既存の構造最適化におけるフィルタリング法 [4] は、フィルターを用いてチェッカーボードパターンと呼ばれる、形状として成立させることが難しい形状を回避する手法である。この場合は結果として目的関数が増加することがある。

また、数値計算の結果を見ると、いずれの例でも勾配法の場合に比べて収束速度が向上している。勾配法でグレーな点は勾配が小さく、 κ で制御される更新幅が小さいために、収束が遅い。これは勾配法が一次収束することから妥当な結果である。これに対し、二次収束することが知られている Newton 法を用いることで、収束速度が向上したと解釈することができる。

iSA 法では ϕ と \mathbf{u} を同時に収束させていくため、 ϕ だけを早く収束させることは得策ではない。提案手法では、最適解のおよその形が決まってから Newton 方向を利用する。ここで例えば最初から Newton 方向を利用した場合、 \mathbf{u} に比べて ϕ の収束が早いために意味のある解が得られない場合がある。

以上のように、本手法はグレースケール問題を解決するとともに収束速度を向上させる手法であると言える。グレースケール問題の解決策としては level-set 関数を用いた手法が広く利

用されている。Level-set 関数を用いた場合には、境界の位置を流体計算の格子点に依存せずに表現することができるため滑らかな境界を得ることができるものの、流れ場の計算上は明確な境界がない部分にも視覚的には明確な境界が引かれる場合がある。その一方で、本手法では滑らかな境界表現ができず、格子点に依存した階段状の境界が得られるものの、視覚的な境界と流体計算上の境界が一致するという特徴がある。

5.5 部分的に Newton 法を利用した手法の結論

本章では部分的に Newton 法を利用することで第4章の手法の収束速度を向上させる手法を構築した。まず第4章の直接法を用いた定式化を元に目的関数の Hesse 行列を定義し、その対角成分の正負をもとに、対角成分が正の場合は Hesse 行列を使用し、負の場合は第4章と同じく勾配法を用いる。この新しく定義された更新方向は、その瞬間は降下方向であることが保証される。一方で解は iSA 法に基づいて更新され、更新のたびに場の変数が増えるため、ある時刻に効果方向であった方向が、次の時刻にも降下方向であるとは限らない。素朴な勾配法は一次収束することが知られており、最適解近傍での収束速度が遅い。一方で Newton 法は二次収束し、最適解近傍での収束速度が早い。本章では、この特徴を活かし、勾配法で最適解を探索したのちに、収束速度が遅くなった段階で Hesse 行列に切り替えて最適化を行った。

数値計算を実施し、収束速度が向上することを確認した。一方で、変数を半ば強制的に 0 や 1 に変更するという特徴があり、局所最適解にとらわれやすいということも明らかになった。そのため、勾配法である程度形状が決まってから、勾配法による更新幅が小さくなった段階で本手法に切り替えて最適化を行うことが有効である。また、この特徴のためにグレースケールが現れにくいという特徴も得られた。

第 6 章

境界条件を適切に考慮した定常流れの最適化

本章では流体と固体の境界条件を考慮した最適化手法を構築する。境界条件は bounce-back 条件を用い、bounce-back 条件を考慮した感度を計算する手法を提案する。また空間差分の方法として素朴な両側差分を用いた場合に適切な解が得られないことを示し、この問題を解決するための差分法を提案する。また数値計算例により手法の妥当性を検討する。

6.1 トポロジー最適化問題における境界条件の取り扱い

流れの数値計算では、境界条件を適切に設定することが重要である。特に早い流れにおいて、固体壁における境界条件は流れの挙動を大きく左右する。非滑り境界条件は代表的な固体壁の境界条件の一つである。非滑り境界条件では、流れと固体壁の境界面上では流れが固体壁に固着しているという仮定を置く [77]。このとき、境界 $\Gamma = \Omega_f \cap \Omega_s$ において

$$\mathbf{u}(\mathbf{x}) = \mathbf{0} \text{ if } \mathbf{x} \in \Gamma$$

という境界条件を加えることになる。

第 4 章で用いた多孔質媒体を用いた定式化では、流体領域と固体領域の境界が曖昧であった。このような境界は例えば細かなビーズ状の物体を利用することで物理的に実現できる可能性はあるものの、一般の機械設計ではこのようなことは行わない。機械設計に用いるためには、流体領域と固体領域を明確に区別した上で、その境界上に適切な境界条件を適用して数値計算を行うことが求められる。第 5 章で部分的に Newton 法を用いることで流体領域と固体領域を明確に区別することができたが、この場合も境界は多孔質媒体で満たされており、自然に適用される境界条件は前述のような非滑り境界条件とは異なる。そこで本章では、境界を明確に表現した上で、その境界上に非滑り境界条件を適用した上で最適化を行う手法を構築する。

境界を明確に表現する手法には、構造最適化の手法を応用することができる。初期の構造トポロジー最適化における問題として、領域ごとの密度を変数とした時に、密度が 0 でも 1 でも

ないグレーな領域が広く存在するグレースケール問題と、密度が1に近い要素と0に近い要素が千鳥格子上に並ぶチェッカーボード問題という二種類の問題があった。いずれの問題においても、固体領域の境界を定義することが難しい。これらを解決する一つの方法として level-set 関数を用いて固体領域の境界を明確に表現する手法が提案された [2]。この手法では level-set 関数を用いて密度が閾値以上の領域に固体が存在するとし、level-set 関数を更新することで形状を変更する。

流れ場の最適化において level-set 関数を用いて境界を明確に表現する試みも行われている [79]。一方で境界条件を適切に適用する試みは少ない。LBM において非滑り境界条件を適用する場合、bounce-back 条件が広く用いられる [17, 89]。素朴な bounce-back 条件では、固体壁は隣接する格子点の midpoint で実現される。ここで場の変数の微分を離散化するとき、素朴な両側差分を用いると bounce-back 条件による境界の位置を適切に考慮できない。実際、素朴な両側差分法を用いた場合に最適化計算が収束しないことを数値計算例を用いて紹介する。そこで本章では bounce-back 条件を考慮した差分法を構築し、適切な最適解が得られるような工夫を施す。Bounce-back 条件を導入した LBM については、第 2.5 節で定式化したとおりである。このとき、乱流モデルとしてコヒーレント構造 Smagorinsky モデル [33] を用いる。

また iSA 法を用いる場合は、一連の LBM の計算の最中に壁面境界の位置が移動する。そのために、ある計算ステップでは固体領域であった格子点が、次のステップでは流体領域に変更されたり、またその逆が起こる。このとき、特に固体領域が流体領域に変化する際は場の変数を適切に初期化する必要がある。これは iSA 法に独特な点である。本章では以上の工夫を用いて最適化アルゴリズムを構築する。

本章は次のように構成される。まず第 6.2 節で、境界位置を明確に考慮した最適化問題を定式化し、iSA 法に基づいた最適化法を構築する。次に第 6.3 節で各 LBM の反復における格子点の初期化方法を構築する。第 6.4 節では最適化法を実装する際の具体的な離散化手法を構築する。また bounce-back 法を用いた場合の LBM の境界の位置を考慮した差分法を構築し、感度の計算のための $\nabla_{\phi} D$ の計算方法を定式化する。以上で構築された手法を整理してアルゴリズムとして記述する。続いて第 6.5 節で数値計算例を示す。数値計算例では、本節で構築する手法に加えて、差分法として素朴な両側差分法を用いた場合に得られる結果も示し、第 6.4.1 節で構築した差分法の妥当性を示す。最後に第 6.6 節で結論を述べる。

6.2 最適化問題の定式化

本節では最適化問題を定式化する。第 4 章と第 5 章では多孔質媒体モデルを使用したが、前述のとおり、多孔質媒体では境界上で適切な境界条件を適用することができない。そこで、設計変数 ϕ は変えず、 ϕ が閾値よりも大きい場合には流体領域とし、小さい場合には固体領域として、最適化問題を記述する。

まず設計変数 ϕ は、第 4 章、第 5 章と変わらず、設計領域 Ω 全体で定義される $0 \leq \phi \leq 1$

を満たす連続変数とする．次に ϕ の関数で，0-1 の離散的な値を取る χ を次式で定義する：

$$\chi(\phi; \mathbf{x}, t) = \begin{cases} 1 & \text{if } \frac{1}{2} \leq \phi(\mathbf{x}, t), \\ 0 & \text{if } \phi(\mathbf{x}, t) < \frac{1}{2}. \end{cases} \quad (6.1)$$

境界条件を適用するためには Ω を流体領域 Ω_f と固体領域 Ω_s に分ける必要がある．そこで χ を用いて次のように Ω_f と Ω_s を判別する：

$$\begin{cases} \mathbf{x} \in \Omega_f & \text{if } \chi(\mathbf{x}, t) = 1, \\ \mathbf{x} \in \Omega_s & \text{if } \chi(\mathbf{x}, t) = 0. \end{cases}$$

ここで χ は ϕ に関する非連続関数である．従って $\frac{\partial \chi}{\partial \phi}$ は超関数の意味で定義する必要がある．本論文では次式を用いて χ の ϕ による微分を定義する：

$$\frac{\partial \chi}{\partial \phi} = \delta \left(\phi - \frac{1}{2} \right), \quad (6.2)$$

ただし δ は Dirac のデルタ関数である．数値計算上は (6.2) を次のように離散化する：

$$\frac{\partial \chi}{\partial \phi}(\mathbf{x}) = \begin{cases} \frac{1}{2\delta r} & \text{if } \exists i \text{ such that } (\phi(\mathbf{x}) - \frac{1}{2})(\phi(\mathbf{x} + \mathbf{e}_i) - \frac{1}{2}) \leq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (6.3)$$

(6.3) は，ある格子点において，隣接する格子点との間に固体壁があった場合に，その格子点で $\frac{\partial \chi}{\partial \phi} = \frac{1}{2\delta r}$ とすることを意味している．このように，本章および第7章では多孔質媒体モデルを用いないため，(2.8) で定義されていた外力 $\hat{\mathbf{F}}$ は，本章以降では $\hat{\mathbf{F}} = 0$ としてよい．

以上で定義した χ を用いて流れ場を解く．目的関数は第4章および第5章と同じくエネルギー損失関数とし，これを最小化する．ただし，本章および第7章では多孔質媒体モデルを用いないため外力項は零であり，(4.2) で定義された摩擦散逸関数 C は用いない．改めて粘性散逸関数 D は次式で定義される：

$$D(\mathbf{u}; \phi, t) = \int_{\Omega_f} \frac{1}{2} \nu \|\varepsilon(\mathbf{u}(\mathbf{x}, t))\|^2 \, d\mathbf{x} \quad (6.4a)$$

$$= \int_{\Omega} \frac{1}{2} \nu \|\varepsilon(\mathbf{u}(\mathbf{x}, t))\|^2 \, d\mathbf{x} \quad (6.4b)$$

$$= \int_{\Omega} -\frac{1}{2} \nu \left(\langle \Delta \mathbf{u}, \mathbf{u} \rangle + \frac{1}{2} \|\text{rot} \mathbf{u}\|^2 \right) \, d\mathbf{x}. \quad (6.4c)$$

ただし $\varepsilon(\mathbf{u}) \in \mathbb{R}^{3 \times 3}$ は $\varepsilon(\mathbf{u}) = \nabla \mathbf{u} + (\nabla \mathbf{u})^\top$ で定義される歪み速度テンソルとし， $\text{rot} \mathbf{u}$ は $\text{rot} \mathbf{u} = \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^\top$ で定義される渦度ベクトルである．(6.4a) と (6.4b) は， Ω_s において \mathbf{u} と $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ が零であることから成り立つ．(6.4b) は第4章と第5章で用いた定義 (6.4c) において $p = 0$ とした場合と一致する．第4章で述べたように， $p \neq 0$ としたのは多孔質媒体における最適化の収束性を向上させることが目的であり，力学的には $p = 0$ が正しい．第4章および第5章と記号が重複するが，本節以降では D の定義として (6.4c) を用いる．

以上で定義した目的関数を用いて、次の最適化問題が導かれる。

$$(P'_{NS}) : \min_{\phi, \mathbf{u}} D(\mathbf{u}, \phi) \quad (6.5a)$$

$$\text{s. t. } \mathbf{u} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = -\text{grad} p + \nu \Delta \mathbf{u}, \quad (6.5b)$$

$$0 \leq \phi(\mathbf{x}, t) \leq 1, \quad (6.5c)$$

$$\int_{\Omega} \phi(\mathbf{x}, t) d\mathbf{x} = \bar{V}. \quad (6.5d)$$

最適化問題 (P'_{NS}) を、iSA 法を用いて近似的に解く。以降では時間と空間について離散化したものを考え、(1.1) の定義にしたがって記述する。目的関数 (6.4c) は次のように離散化される。

$$D(\mathbf{u}; \phi, t_k) = \sum_{\mathbf{x} \in \hat{\Omega}} -\frac{1}{2} \nu \left(\langle \Delta \mathbf{u}, \mathbf{u} \rangle + \frac{1}{2} \|\text{rot} \mathbf{u}\|^2 \right) \delta r. \quad (6.6a)$$

第3章と第4章における最適化問題の関係と同様に、(3.2) は (6.5) に対応している。また本節において (3.16) に対応する最適化問題を書き下すと、次のようになる：

$$(P_{BB}) \quad \min_{\phi(\mathbf{x}, t_k), \mathbf{u}(\mathbf{x}, t_k)} D(\mathbf{u}(\mathbf{x}, t_k); \phi(\mathbf{x}, t_k)), \quad (6.7a)$$

$$\text{s. t. } \mathbf{u}(\mathbf{x}, t_k) = \mathbf{u}_{\text{LBM}}(\mathbf{f}(\mathbf{x}, t_{k-1}); \phi(\mathbf{x}, t_k)), \quad (6.7b)$$

$$0 \leq \phi(\mathbf{x}, t_k) \leq 1 \quad (6.7c)$$

$$\sum_{\mathbf{x} \in \hat{\Omega}} \phi(\mathbf{x}, t_k) \delta r = V_k. \quad (6.7d)$$

ただし (6.7b) において \mathbf{u} は χ を介して ϕ に依存している。設計変数 $\phi(t_k)$ も場の変数 \mathbf{u} も、最適化問題を解く過程で更新される。いま、 t_N は設計変数も場の変数も更新されなくなった時刻として定義し、そのような t_N が有限の時間内に実現されることを仮定する。後にこの仮定を緩和し、 t_N において設計変数のみが更新されなくなった場合について論ずる。(6.7d) は体積制約を示している。第4章で述べたように、 V_k は (4.5) を用いる。

(3.11) で定義された、iSA 法で用いる Lagrange 関数は次のように定式化される：

$$\mathcal{L}_k = D(t_k) - \langle \mathbf{u}_k - \mathbf{u}_{\text{LBM}}, \mathbf{v}_k \rangle - \langle \phi_k, \lambda_k^1 \rangle - \langle \mathbf{1} - \phi_k, \lambda_k^2 \rangle - \sum_{k=0}^N \left(\sum_{\mathbf{x} \in \hat{\Omega}} \phi_k \delta r - V_k \right) M_k,$$

\mathcal{L}_k を用いると (3.12) に相当する式はと次のようになる。

$$\frac{\partial \mathcal{L}_k}{\partial \phi_k(\mathbf{x})} = \frac{\partial D}{\partial \phi_k(\mathbf{x})} + \left\langle \frac{\partial \mathbf{u}_{\text{LBM}}}{\partial \phi_k(\mathbf{x})}, \mathbf{v}_k \right\rangle - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k, \quad (6.8a)$$

$$\frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k(\mathbf{x})} = \frac{\partial D}{\partial \mathbf{u}_k(\mathbf{x})} - \mathbf{v}_k^\top. \quad (6.8b)$$

したがって感度は次式であらわされる。

$$\begin{aligned} \frac{\partial \mathcal{L}_k}{\partial \phi_k(\mathbf{x})} &= \frac{\partial D}{\partial \phi_k(\mathbf{x})} + \frac{\partial D}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \phi_k(\mathbf{x})} - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k \\ &= \nabla_{\phi_k(\mathbf{x})} D - \lambda_k^1(\mathbf{x}) + \lambda_k^2(\mathbf{x}) - M_k. \end{aligned}$$

iSA 法にしたがって、(3.36) を今回の場合に当てはめると、次の方程式が得られる、

$$\mathbf{u}_{k+1}(\mathbf{x}) = \mathbf{u}_{\text{LBM}}(\mathbf{f}(\mathbf{x}, t_k), \phi_{k+1}(\mathbf{x})), \quad (6.9a)$$

$$\phi_{k+1}(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})} D + M_{k+1} \leq 0, \\ \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})} D + M_{k+1} & \text{if } 0 < \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})} D + M_{k+1} < 1, \\ 1 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}(\mathbf{x})} D + M_{k+1} \geq 1. \end{cases} \quad (6.9b)$$

ただし κ は収束性の速度を調整するために使用する正の定数であり、 M_k はアルゴリズム 2 にしたがって決定する。

6.3 最適化の過程における場の変数の初期化

iSA 法では設計変数と場の変数を同時に更新する。時刻 t_k では固体領域であった点が t_{k+1} では流体領域になった場合に、その点の場の変数を適切に初期化する必要がある。また逆に流体領域が固体領域に変化した場合に、場の変数に物理的に矛盾がおこらないように、適切に処理する必要がある。本節ではこれらの場の変数の初期化の方法を導く。

6.3.1 固体領域が流体領域に変化した場合の初期化方法

まず、ある格子点について時刻 t_{k-1} において $\mathbf{x} \in \Omega_s$ と仮定する。すなわち $\chi(\mathbf{x}, t_{k-1}) = 0$ である。その後時刻 t_k においてその点が流体領域に変化したと仮定する。すなわち $\chi(\mathbf{x}, t_k) = 1$ である。このとき \mathcal{I} と \mathcal{J} の二つの集合を定義する：

$$\begin{aligned} \mathcal{I} &= \{i \mid \chi(\mathbf{x} + \mathbf{e}_i, t_{k-1}) = 1\}, \\ \mathcal{J} &= \{j \mid \chi(\mathbf{x} + \mathbf{e}_j, t_{k-1}) = 0\}. \end{aligned}$$

\mathcal{I} は点 \mathbf{x} の i 方向の隣接点が流体領域であるような添え字 i の集合であり、 \mathcal{J} はそれが固体領域であるような添え字の集合である。

次に仮想粒子の考え方を導入し、bounce-back 条件を仮想粒子を用いて説明する。 Ω_f に接する $\mathbf{x} \in \Omega_s$ において、図 6.1(a) の上図に示すように仮想粒子 $f_i(\mathbf{x})$ ($i \in \mathcal{I}$) を定義する。

$$f_i(\mathbf{x}) = f_{-i}(\mathbf{x} + \mathbf{e}_i) \quad (i \in \mathcal{I})$$

仮想粒子は、隣接する流体領域の点の粒子を、固体壁に関して鏡対象になるように定義したものと考えても良い。この仮想粒子を隣接する流体領域の点 $\mathbf{x} + \mathbf{e}_i$ に移流させると、図 6.1(a) の下図に示すように、

$$f_i(\mathbf{x} + \mathbf{e}_i) = f_{-i}(\mathbf{x} + \mathbf{e}_i) \quad (6.10)$$

となる。すると $\mathbf{x} + \mathbf{e}_i$ において (6.10) は (2.14b) と等価である。以上より、bounce-back 条件は固体領域の点で仮想粒子を定義して移流させることと等価であると言える。そこで、 \mathbf{x} が流体領域に変化する場合に、仮想粒子を用いて $\mathbf{f}(\mathbf{x})$ を初期化する。

\boldsymbol{x} における仮想粒子は、添え字集合 \mathcal{I} に含まれるすべての i について定義される：

$$f_i(\boldsymbol{x}, t_k) = f_{-i}(\boldsymbol{x} + \boldsymbol{e}_i, t_k) \quad (i \in \mathcal{I}). \quad (6.11)$$

時刻 t_k において流体領域に変更されると、この仮想粒子は実際の粒子と同じく衝突と移流を行う。以上の様子を図 6.1(b) に図示する。

\mathcal{J} に含まれる添え字 j については、 $f_{-j}(\boldsymbol{x} + \boldsymbol{e}_j, t_k)$ が定義されないため、 $f_j(\boldsymbol{x})$ に同様の処理ができない。そこで、 $f_j(\boldsymbol{x})$ を初期化する代わりに $\rho = \sum_i f_i$ が適切な値になるように $f_j(\boldsymbol{x})$ の値を決める。連続的な場で考えると、密度 ρ は \boldsymbol{x} において空間的に連続であるべきである。そこで $\rho(\boldsymbol{x})$ を、隣接する格子点の密度の平均値とすることは妥当である。具体的には次式を満たすように ρ を決める：

$$\rho(\boldsymbol{x}, t_k) = \frac{\sum_{i \in \mathcal{I}} w_i \rho(\boldsymbol{x} + \boldsymbol{e}_i, t_k)}{\sum_{i \in \mathcal{I}} w_i}. \quad (6.12)$$

次にこの ρ を満たすように \boldsymbol{f}_j ($j \in \mathcal{J}$) を次式で決定する：

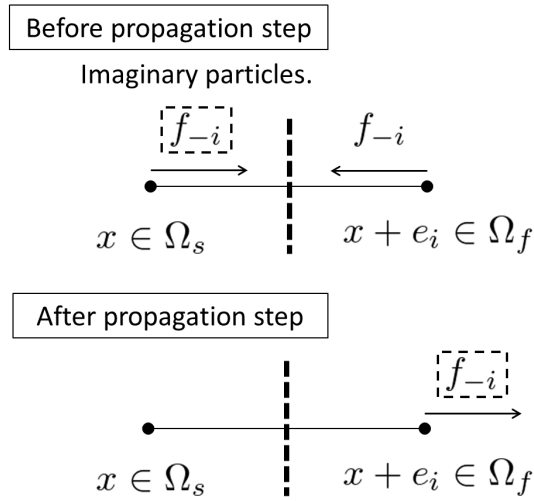
$$f_j = \frac{w_j}{\sum_{\ell \in \mathcal{J}} w_\ell} \left(\rho - \sum_{i \in \mathcal{I}} f_i \right) \quad (j \in \mathcal{J}). \quad (6.13)$$

(6.11) と (6.12), (6.13) により、 \mathcal{I} と \mathcal{J} に含まれる任意の添え字 i について f_i の値を与えることができる。 \boldsymbol{u} は f_i から定義 (2.3) に従って計算することで適切に初期化することができる。

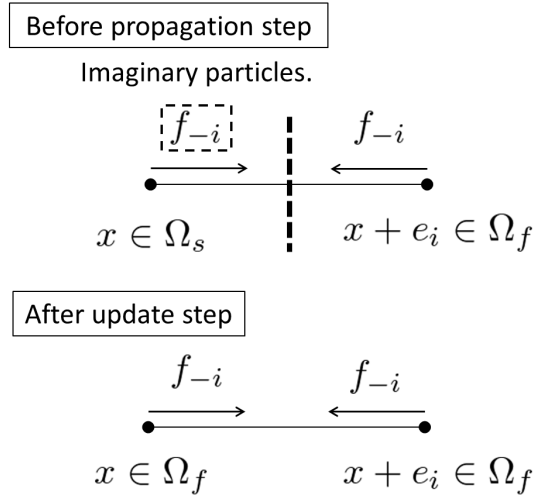
前述の議論の過程で密度の空間連続性を仮定した。この仮定は本論文で扱う非圧縮性ないし微圧縮性流れにおいては当然成り立つべき仮定である。ここで微圧縮性という用語は必ずしも一般的な用語ではない。LBM ではごく低速の流れを解く場合にも原理的に非圧縮性が成り立たないために使用される用語である。また圧縮性流れにおいても、流れの速度が音速を超えない亜音速流れにおいては自然な仮定である。しかし特に流速が音速を超えるような遷音速流れや超音速流れにおいては、密度は一般に空間非連続であり、前述の仮定が不適切である。このような場合には別の処理を考える必要がある。

6.3.2 流体領域が固体領域に変化した場合の初期化方法

Bounce-back 条件を用いた MRT-LBM では、固体領域 Ω_s 上で \boldsymbol{f} は更新されない。従って \boldsymbol{u} もその点が固体領域である間は変化しない。通常の LBM では Ω_s と Ω_f が固定されているため、初期値を $\boldsymbol{x} \in \Omega_s$ において $\boldsymbol{u}(\boldsymbol{x}, t_0) = \mathbf{0}$ と決めれば、その後の任意の時刻で $\boldsymbol{u}(\boldsymbol{x}, t_k) = \mathbf{0}$ が満たされ、力学的に矛盾しない。しかしトポロジー最適化を行う場合には、時刻 t_k において流体領域に含まれていた点が時刻 t_{k+1} において固体領域に変化する場合がある。この場合、単純に LBM に従うと、例えば i 方向の隣接格子点が固体領域であった場合に、 $f_i(\boldsymbol{x}, t_{k+1}) = f_{-i}(\boldsymbol{x}, t_k)$ となる。このとき、 \boldsymbol{x} は固体領域の点であるにも関わらず、一般的に $\boldsymbol{u}(\boldsymbol{x}, t_{k+1}) \neq \mathbf{0}$ になる。これは力学的に不適切である。ここでは i 方向の隣接格子点が固体領域という仮定を置いたがこれは頻りに満たされる仮定である。そこで明示的に Ω_s における



(a) 仮想粒子を用いた移流操作の様子.



(b) 相変化の操作の様子.

図 6.1. 相変化の操作の概念図.

f を定義して、力学的に意味のある場の変数を決める．固体領域 Ω_s には流れが存在しないため、同様に粒子分布関数 f も零であると考えerことは自然である．従って

$$f(x) = 0 \text{ if } x \in \Omega_s$$

と定義する．この定義より $\rho = 0$ が得られるが、 u を定義する (2.3b) では u は不定である．そこで改めて、

$$u(x) = 0 \text{ if } x \in \Omega_s$$

と定義する．以上の定義により f は LBM の操作に関係なく変更されるが、その変更は該当する格子点以外に影響を与えず、特に流体領域には影響がない．

6.4 最適化問題の解法

6.4.1 差分作用素の定義

第2.5節において、bounce-back条件を適用した場合、固体壁は隣接する格子点の midpoint にあることを模擬していることを説明した。本節ではこの固体壁の位置を適切に考慮するために、一般的な定義とは異なる微分作用素を定義する。本節の考え方自体はLBMの任意の格子モデルで使用できるが、本論文ではD3Q19モデルでのみ使用するため、D3Q19モデルを想定して定式化を行う。

一階と二階の微分作用素は離散的な数値計算では差分法を用いて近似される。連続関数の空間微分では一般的な両側差法は

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial x} &= \frac{\mathbf{u}(\mathbf{x} + \mathbf{e}_1) - \mathbf{u}(\mathbf{x} - \mathbf{e}_1)}{2\delta_x}, \\ \frac{\partial^2 \mathbf{u}}{\partial x^2} &= \frac{\mathbf{u}(\mathbf{x} - \mathbf{e}_1) - 2\mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \mathbf{e}_1)}{\delta_x^2}\end{aligned}$$

で定義される。ただし x のみに関する一階と二階の導関数を示した。また \mathbf{e}_1 は x 方向に隣接する格子点へのベクトルである。差分を計算する際、隣接する格子点の情報を用いており、本来はその midpoint で $\mathbf{u} = \mathbf{0}$ であるにもかかわらず、その情報が使用されない。そこで、midpoint での場の変数の値を使用して差分法を定義する。

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_1) - \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_1)}{\delta_x}, \quad (6.14a)$$

$$\frac{\partial \mathbf{u}}{\partial y} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_3) - \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_3)}{\delta_x}, \quad (6.14b)$$

$$\frac{\partial \mathbf{u}}{\partial z} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_5) - \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_5)}{\delta_x}, \quad (6.14c)$$

$$\frac{\partial^2 \mathbf{u}}{\partial x^2} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_1) - 2\mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_1)}{\frac{1}{4}\delta_x^2}, \quad (6.14d)$$

$$\frac{\partial^2 \mathbf{u}}{\partial y^2} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_3) - 2\mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_3)}{\frac{1}{4}\delta_x^2}, \quad (6.14e)$$

$$\frac{\partial^2 \mathbf{u}}{\partial z^2} = \frac{\mathbf{u}(\mathbf{x} - \frac{1}{2}\mathbf{e}_5) - 2\mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \frac{1}{2}\mathbf{e}_5)}{\frac{1}{4}\delta_x^2}, \quad (6.14f)$$

$$\mathbf{u}\left(\mathbf{x} + \frac{1}{2}\mathbf{e}_i\right) = \begin{cases} \frac{1}{2}(\mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \mathbf{e}_i)) & \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 1, \chi(\mathbf{x}) = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (6.14g)$$

ただし、 \mathbf{e}_1 と \mathbf{e}_3 、 \mathbf{e}_5 は (2.2) で定義されたベクトルのうち、それぞれ x 、 y 、 z 軸に平行なベクトルである。この差分の定義は χ に依存している。そこでこの差分法を設計変数依存の差分法と呼ぶ。

この設計変数依存の差分法 (6.14) は、流体領域 Ω_f の内点、つまり $\chi(\mathbf{x} + \mathbf{e}_i) = 1$ ($\forall i \in [0, N]$)、 $\chi(\mathbf{x}) = 1$ であるような \mathbf{x} については、一般的な両側差分法と一致する。また (6.14)

は全領域 Ω で定義されており，固体領域 Ω_s においても矛盾なく定義される．このとき，第 6.3.2 節で述べた通り $\mathbf{u}(\mathbf{x}) = \mathbf{0}$ ($\mathbf{x} \in \Omega_s$) であるから，

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} &= 0 & \text{if } \mathbf{x} \in \Omega_s, \\ \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2} &= 0 & \text{if } \mathbf{x} \in \Omega_s \end{aligned}$$

が成り立つ．これは物理的にも矛盾しない．

6.4.2 $\nabla_\phi D$ の定式化

本節では (6.9) で使用する $\nabla_\phi D$ を定式化する．まず目的関数 D は \mathbf{u} を陽に用いると

$$D(\mathbf{u}; \phi, t_k) = -\frac{1}{2}\nu \sum_{\mathbf{x} \in \hat{\Omega}} \left(\langle \Delta_\chi \mathbf{u}, \mathbf{u} \rangle + \frac{1}{2} \|\text{rot}_\chi \mathbf{u}\|^2 \right)$$

で定義される．ここで Δ_χ は，(6.14) で示したように微分作用素が χ に依存することを示している．次に連鎖側により $\nabla_\phi D$ は

$$\begin{aligned} \nabla_\phi D &= \frac{\partial D}{\partial \chi} \frac{\partial \chi}{\partial \phi} \\ &= -\frac{1}{2}\nu \left(\left\langle \frac{\partial \Delta_\chi \mathbf{u}}{\partial \chi}, \mathbf{u} \right\rangle + \left\langle \Delta_\chi \mathbf{u}, \frac{\partial \mathbf{u}}{\partial \chi} \right\rangle + \left\langle \text{rot}_\chi \mathbf{u}, \frac{\partial (\text{rot}_\chi \mathbf{u})}{\partial \chi} \right\rangle \right) \frac{\partial \chi}{\partial \phi} \end{aligned} \quad (6.15)$$

と分解できる．続いて $\frac{\partial \Delta_\chi}{\partial \chi}$ と $\frac{\partial \mathbf{u}}{\partial \chi}$, $\frac{\partial \text{rot}_\chi}{\partial \chi}$ を定式化する．まず $\frac{\partial \Delta_\chi}{\partial \chi}$ について， Δ_χ は (6.14) により $\chi(\mathbf{x})$ と関係している．しかし (6.14) は χ に関する非連続関数である．そこで滑らかな関数 Δ_χ^s を次のように定義する：

$$\Delta_\chi^s \mathbf{u} \Big|_{\mathbf{x}+\mathbf{e}_i} = \chi(\mathbf{x}) \Delta_{\chi(\mathbf{x})=1} \mathbf{u} \Big|_{\mathbf{x}+\mathbf{e}_i} + (1 - \chi(\mathbf{x})) \Delta_{\chi(\mathbf{x})=0} \mathbf{u} \Big|_{\mathbf{x}+\mathbf{e}_i} \quad (6.16)$$

ただし $\Delta \mathbf{u} \Big|_{\mathbf{y}}$ は $\mathbf{x} = \mathbf{y}$ における $\Delta \mathbf{u}$ の値を表している．滑らかな関数 (6.16) は， $\chi = 0$ および 1 における Δ_χ の値を線形に結んだものであり， $\chi = 0$ および 1 における値は変化していないことから妥当である．(6.16) より，

$$\frac{\partial \Delta_\chi^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = \Delta_{\chi(\mathbf{x})=1} \mathbf{u} \Big|_{\mathbf{x}+\mathbf{e}_i} - \Delta_{\chi(\mathbf{x})=0} \mathbf{u} \Big|_{\mathbf{x}+\mathbf{e}_i} \quad (6.17)$$

が得られる．したがって (6.17) は次のように定式化される：

$$i = 0, \quad \frac{\partial \Delta_\chi^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_0} = \Delta_{\chi(\mathbf{x})=1} \mathbf{u} \Big|_{\mathbf{x}}, \quad (6.18a)$$

$$i = 1, \dots, 6, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 0 \quad \frac{\partial \Delta_\chi^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = 0, \quad (6.18b)$$

$$i = 1, \dots, 6, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 1 \quad \frac{\partial \Delta_\chi^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = \frac{\frac{1}{2}(\mathbf{u}(\mathbf{x} + \mathbf{e}_i) + \mathbf{u}(\mathbf{x}))}{\frac{1}{4}\delta_x^2}. \quad (6.18c)$$

ここで (6.18a) は (6.17) に $\Delta_{\chi(\mathbf{x})=0}\mathbf{u}|_{\mathbf{x}} = 0$ を代入することで得られる. また (6.18b) は, $\Delta_{\chi}\mathbf{u}|_{\mathbf{x}+\mathbf{e}_i}$ が $\chi(\mathbf{x})$ にかかわらず零であることから得られる. さらに (6.18c) は, (6.17) に (6.14) を代入することで得られる.

続いて, 再び連鎖側により $\frac{\partial \mathbf{u}}{\partial \chi}$ は

$$\frac{\partial \mathbf{u}}{\partial \chi} = \sum_{i=0}^{n-1} \frac{\partial \mathbf{u}}{\partial f_i} \frac{\partial f_i}{\partial \chi} = \frac{1}{\rho \delta t} \sum_{i=0}^{n-1} \mathbf{c}_i \frac{\partial f_i}{\partial \chi} \quad (6.19)$$

と分解される. f_i と χ の関係性は (2.14) で定義されているが, (2.14) は χ に関して非連続な関数である. 微分を定義するため, 次式により滑らかな関数を定義する.

$$f_i^s(\mathbf{x}, t_{k+1}) = \chi(\mathbf{x} - \mathbf{e}_i, t_k) f_i^{\text{col}}(\mathbf{x} - \mathbf{e}_i, t_k) + (1 - \chi(\mathbf{x} - \mathbf{e}_i, t_k)) f_{-i}^{\text{col}}(\mathbf{x}, t_k). \quad (6.20)$$

ここで $\chi = 0$ および 1 において, (6.20) と (2.14b) は等価である, また (6.20) は χ に関して連続な関数である. ここで, (6.20) の左辺の f_i は $\mathbf{x} = \mathbf{x}$ における値である一方で, 右辺の χ は $\mathbf{x} = \mathbf{x} - \mathbf{e}_i$ における値である. この座標の違いに注意して微分すると, $\frac{\partial f_i}{\partial \chi}$ は

$$\frac{\partial f_i^s(\mathbf{x}, t_{k+1})}{\partial \chi(\mathbf{x} - \mathbf{e}_i, t_k)} = f_i^{\text{col}}(\mathbf{x} - \mathbf{e}_i, t_k) - f_{-i}^{\text{col}}(\mathbf{x}, t_k) \quad (6.21)$$

である. 最後に, $\frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi}$ を導く. これまでと同様にして滑らかな関数 $\text{rot}_{\chi}^s \mathbf{u}$ を定義する:

$$\text{rot}_{\chi}^s \mathbf{u} = \chi(\mathbf{x}) \text{rot}_{\chi(\mathbf{x})=1} \mathbf{u}|_{\mathbf{x}+\mathbf{e}_i} + (1 - \chi(\mathbf{x})) \text{rot}_{\chi(\mathbf{x})=0} \mathbf{u}|_{\mathbf{x}+\mathbf{e}_i}.$$

$\text{rot}_{\chi}^s \mathbf{u}$ を用いることで, 次の導関数が得られる:

$$i = 0, \quad \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_0} = \text{rot}_{\chi(\mathbf{x})=1} \mathbf{u} \Big|_{\mathbf{x}}, \quad (6.22a)$$

$$i = 1, \dots, 6, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 0 \quad \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = 0, \quad (6.22b)$$

$$i = 1, 2, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 1 \quad \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = \frac{1}{2\delta_x} \begin{pmatrix} 0 \\ w(\mathbf{x} + \mathbf{e}_i) + w(\mathbf{x}) \\ -(v(\mathbf{x} + \mathbf{e}_i) + v(\mathbf{x})) \end{pmatrix}, \quad (6.22c)$$

$$i = 3, 4, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 1 \quad \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = \frac{1}{2\delta_x} \begin{pmatrix} -(w(\mathbf{x} + \mathbf{e}_i) + w(\mathbf{x})) \\ 0 \\ u(\mathbf{x} + \mathbf{e}_i) + u(\mathbf{x}) \end{pmatrix}, \quad (6.22d)$$

$$i = 5, 6, \quad \text{if } \chi(\mathbf{x} + \mathbf{e}_i) = 1 \quad \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x})} \Big|_{\mathbf{x}+\mathbf{e}_i} = \frac{1}{2\delta_x} \begin{pmatrix} v(\mathbf{x} + \mathbf{e}_i) + v(\mathbf{x}) \\ -(u(\mathbf{x} + \mathbf{e}_i) + u(\mathbf{x})) \\ 0 \end{pmatrix}. \quad (6.22e)$$

これまでに見たように, 導関数を計算するときはその座標の位置に注意して微分する必要が

ある．そこで (6.15) を次のように書き直す：

$$\begin{aligned} \frac{\partial D}{\partial \phi(\mathbf{x}, t_k)} = & -\frac{1}{2}\nu \left(\sum_{i=0}^6 \left\langle \frac{\partial \Delta_{\chi}^s \mathbf{u}}{\partial \chi(\mathbf{x}, t_k)} \Big|_{(\mathbf{x}+\mathbf{e}_i, t_k)}, \mathbf{u}(\mathbf{x} + \mathbf{e}_i, t_k) \right\rangle \right. \\ & + \sum_{i=0}^{n-1} \left\langle \Delta_{\chi} \mathbf{u} \Big|_{(\mathbf{x}+\mathbf{e}_i, t_k)}, \frac{\partial \mathbf{u}(\mathbf{x} + \mathbf{e}_i, t_k)}{\partial \chi(\mathbf{x}, t_k)} \right\rangle \\ & \left. + \sum_{i=0}^6 \left\langle \text{rot}_{\chi} \mathbf{u}, \frac{\partial \text{rot}_{\chi}^s \mathbf{u}}{\partial \chi} \Big|_{(\mathbf{x}+\mathbf{e}_i, t_k)} \right\rangle \right) \frac{\partial \chi(\mathbf{x}, t_k)}{\partial \phi(\mathbf{x}, t_k)}. \end{aligned} \quad (6.23)$$

以上より，(6.23) を用いて (6.9) を解く． M_k はアルゴリズム 2 にしたがって計算する．

6.4.3 最適化アルゴリズム

本節では，これまでの結果を整理して最適化アルゴリズムを記述する．最適化アルゴリズムをアルゴリズム 5 に示す．

アルゴリズム 5.

- Step 1. $k = 0$ とし，適当な値を用いて \mathbf{u}_0 と ϕ_0 を初期化する．
- Step 2. $\phi = \phi_k$ とし，LBM(2.7) を用いて \mathbf{u}_k を計算する．
- Step 3. 感度を計算する．
- Step 4. (6.9b) により ϕ_{k+1} を計算する．
- Step 5. 第 6.3 節を用いて場の変数を初期化する．
- Step 6. 収束判定を満たしていれば停止．
- Step 7. $k = k + 1$ として Step 2 へ．

Step 1 では初期の流れ場，すなわち \mathbf{u}_0 を設定する．第 6.5 節では初期値である ϕ_0 を用いて流れが発達するまで流れ場を解いて \mathbf{u}_0 を計算するが，適当な一様流を初期値として用いることもできる．収束判定には

$$\delta \phi = \|\phi_k - \phi_{k-1}\| - \|\Omega\|$$

で定義される ϕ の変化量を用いて，閾値を下回った段階で計算を停止する．

6.5 数値計算例

本節では (6.9) を用いた最適化の数値計算例を示す．一つ目の計算例は，多くの先行研究でベンチマーク問題として計算されている縮流管の最適化問題である．縮流管問題により，本手法が妥当な最適解を与えることを示す．二つ目の計算例はより現実の問題に近い合流管問題である．合流管問題により，工業的に意味のある問題を解けることを示す．数値計算に用いた計算機は XEON X5690 (3.33GHz with 8GB memory) で，C++ により実装を行った．以下の図では，赤い部分が流体を示し，青い部分が示されている場合はそれが固体領域を示す．アル

ゴリズム 5 に示したように，初期流れ場を作成するために，最適化計算を行う前に流れ場だけの計算を行った．どの数値計算例でも初期流れ場の作成は 5000 ステップあれば十分であった．収束履歴を示す図では，最適化計算の部分に加えて，その直前の初期解を作成するための計算の部分 500 ステップだけ示した．また，目的関数は初期流れ場における目的関数の値を 1 として正規化している．

6.5.1 縮流管問題

本節では流れ場の最適化におけるベンチマーク問題として縮流管問題を解く． $Re = 1000$ と 10 の二種類の異なる Reynolds 数で同じ縮流管問題を最適化し，その違いを議論する．Reynolds 数を変更する際は動粘性係数などを変えず，代表流速を変更した．最適化の領域と境界条件を図 6.2 に示す．格子点数は x, y, z 軸の方向にそれぞれ 100 点，60，30 点とした，入口境界 Γ_u における速度境界条件は $\mathbf{u} = (\bar{U}, 0, 0)^\top$ とし，出口境界 Γ_p における圧力境界条件は $p = 0.101325 \text{ MPa}$ とした． Γ_u と Γ_p 以外の境界は非滑り境界条件とした．Reynolds 数は $Re = \frac{UL}{\nu}$ で定義される．ただし L は図 6.2 で示される代表長さである．体積制約は $\bar{V} = 0.6|\Omega|$ とした．

まず，最適解に近い形状を初期形状として使用した (図 6.3(a))．これを初期解 A とする．最適化の過程における流体領域と固体領域の形状を図 6.3 および図 6.4 に示す．また $Re = 1000$ の場合の収束の履歴を図 6.5 に示す．図 6.5 に示すように，目的関数の値はほぼ単調に減少する．ただし，流れの非定常性に起因する目的関数値の振動も見られる． ϕ がおおよそ定常になり，形状が変化しなくなった時点で最適化計算を停止した．0–2000 ステップでは，目的関数値はわずかな振動を伴いつつ急激に減少する．この期間では，設計空間の上部と下部に曲率の大きい領域がみられる (図 6.3(b))．曲率の大きい領域は流れが大きく曲げられ，また剥離する原因になるためエネルギー損失が大きい．図 6.6 に，500 ステップ目における iSA 法の感度 $\frac{\partial \mathcal{L}}{\partial \phi}$ を示す．ただし感度は $\|\frac{\partial \mathcal{L}}{\partial \phi}\|_\infty = 1$ で正規化されている．前述の曲率が大きい部分と，設計領域の左半分のうちそれ以外の部分を比べると，感度は曲率が大きい領域で比較的大きな値を示している．出口領域付近で感度が大きいのは，流速 $\|\mathbf{u}\|$ が大きいことに起因している．ステップ 2000 以降，目的関数値は徐々に減少する．この期間は境界部分が徐々に変化し，ステップ 3000 では滑らかな境界が得られる．ステップ 2000 から 5000 の間では形状はあまり変わらない．したがって目的関数値の変化もごく僅かである．ステップ 5000 で $\|\phi_k - \phi_{k-1}\|/|\Omega| \leq 1.0 \times 10^{-6}$ が満たされたため，計算を停止した．目的関数値は 0.60 と 0.65 の間を僅かに振動しているものの， ϕ の変化が零に近いことから，最適解であると考えことは妥当である．反復回数は 5000 回で，計算時間は 96 分 38 秒であった．この計算時間は実用上許容できる長さである．

次に，手法のロバスト性を見るために最適解とは大きく異なる初期解を用いて最適化を行った．初期解と最適解を図 6.7 に示す．この最適解は図 6.3 の最適解とはわずかに形状が異なる．この最適解は直線 $y = 1/2L$ ， $z = 1/4L$ に対して線対称であるが，面 $z = 1/4L$ に関して非対称である．非対称な形状は，計算の過程で流れ場が非対称になったことに起因する．

収束の履歴を図 6.8 に示す。目的関数値は 0.47 と 0.77 の間で振動している。図 6.5 では 0.60 から 0.65 の間を振動していたことと比較すると、目的関数の平均値は今回の結果の方が良い。一方で最悪ケースの目的関数値は図 6.3 の方が良い。両者は計算の過程で別の局所最適解に捕われたと考えることができる。ただしどちらの場合も目的関数値は近い値であり、工業的にはどちらの結果が得られても満足できる。

数値計算には 30000 回の反復計算が必要であり、最適解と似た形状の初期解を用いた場合よりも計算時間が長かった。これは初期解と最適解の距離が大きいため、より時間がかかっていると考えることができる。また計算時間は 242 分 12 秒であった。結果として、本手法は初期解の形とトポロジーに関わらず、局所最適解を得ることができることが分かった。しかしながら得られる解は初期解の形状に依存する。

次に $Re = 10$ として、最適解と大きく異なる形状を初期解として最適化を行った。図 6.9(f) に示すように、妥当な最適解が得られた。

ステップ 0 から 1000 において、流体領域はいくつかの流路に分断されており、目的関数値は最適解のそれよりも大きい値になっている。その後、ステップ 1000 から 3000 にかけて、流体領域が結合されて一つの流路になる。さらにステップ 3000 以降、流体領域は徐々に最適解に収束する。 ϕ が $\|\phi_k - \phi_{k-1}\|/|\Omega| \leq 1.0 \times 10^{-6}$ を満たし、 ϕ の振動 $\|\phi_k - \phi_{k-1}\|/|\Omega|$ が 1.0×10^{-6} 以下になったため、最適化計算を 5000 ステップで停止した。 $Re = 1000$ の場合と異なり、目的関数値は振動しなかった。これは Reynolds 数が低いため比較的定常流れになりやすいことに起因すると考えられる。図 6.9(f) と図 6.10(f) に示す最適解は既存の、境界条件が適切に適用されていない最適化手法の結果とよく一致する [7]。この結果から、本手法の妥当性が示される。計算時間は 78 分 4 秒であった。この計算時間は実用上十分許容できる長さである。

異なる Reynolds 数における最適解（図 6.3(f) と図 6.9(f)）を比べると、両者はわずかに異なっていることがわかる。特に曲率は、 $Re = 10$ の場合の方が $Re = 1000$ の場合よりも大きい。二つの最適解において、 $Re = 10$ と $Re = 1000$ で流れ場の計算を行い、目的関数値を比較した。その結果を表 6.1 に示す。目的関数値は振動するため、比較には最大値を用いた。表 6.1 を見ると、それぞれの Reynolds 数においてそれぞれの最適解の方が目的関数値が小さくなっている。 $Re = 1000$ の場合の最適解は図 6.3(f) に示した $Re = 10$ の場合の最適解よりも直線的な形状をしている。この違いは Reynolds 数の違いに起因している。Reynolds 数が大きい場合、慣性力が粘性力よりも大きいため、曲率が大きい領域でエネルギー損失が大きくなる。逆に Reynolds 数が低い場合は慣性力よりも粘性力の方が大きくなるため、曲率が大きくてもエネルギー損失は小さい。このように、最適化の結果は力学的に妥当である。

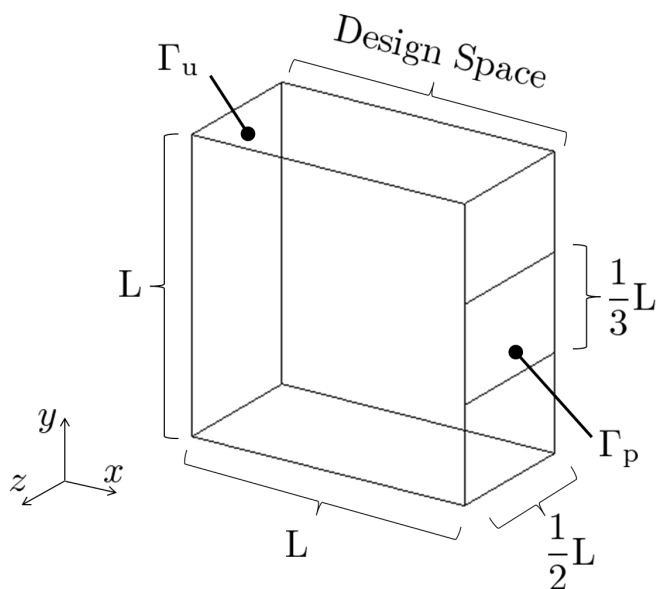
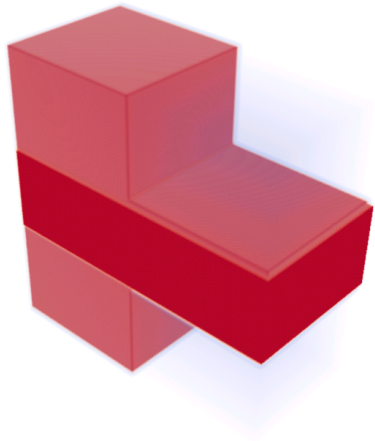


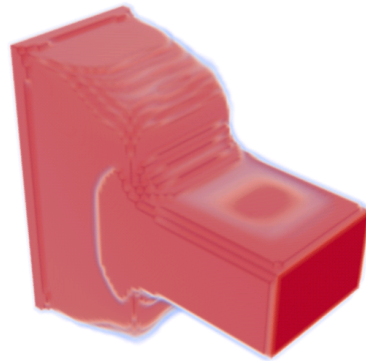
図 6.2. 縮流管問題の設計領域と境界条件.

表 6.1. 二種類の Reynolds 数に対する最適解に対して二種類の Reynolds 数の流れを流した場合の目的関数値の比較.

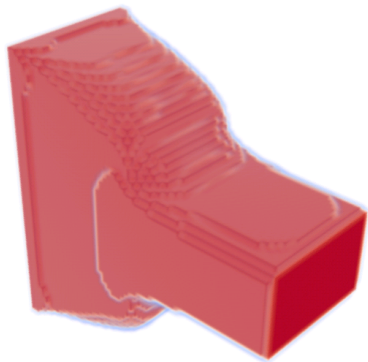
	The optimal shape at Re = 1000	The optimal shape at Re = 10
Re = 1000	0.6521	0.7603
Re = 10	0.1031	0.0940



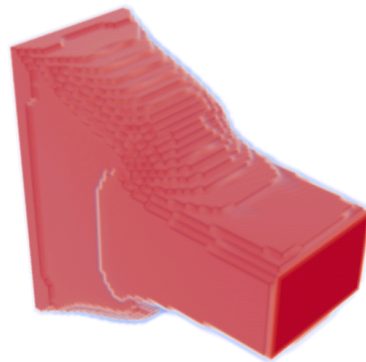
(a) Step 0 (初期解 A)



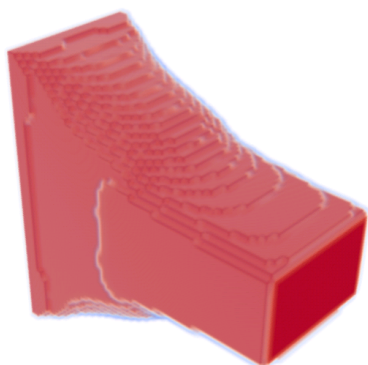
(b) Step 200



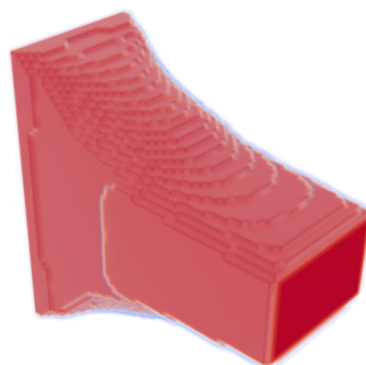
(c) Step 500



(d) Step 1,000



(e) Step 2,000



(f) Step 3,000

図 6.3. 初期解 A を用いた縮流管問題の形状変化の様子 ($Re = 1000$).

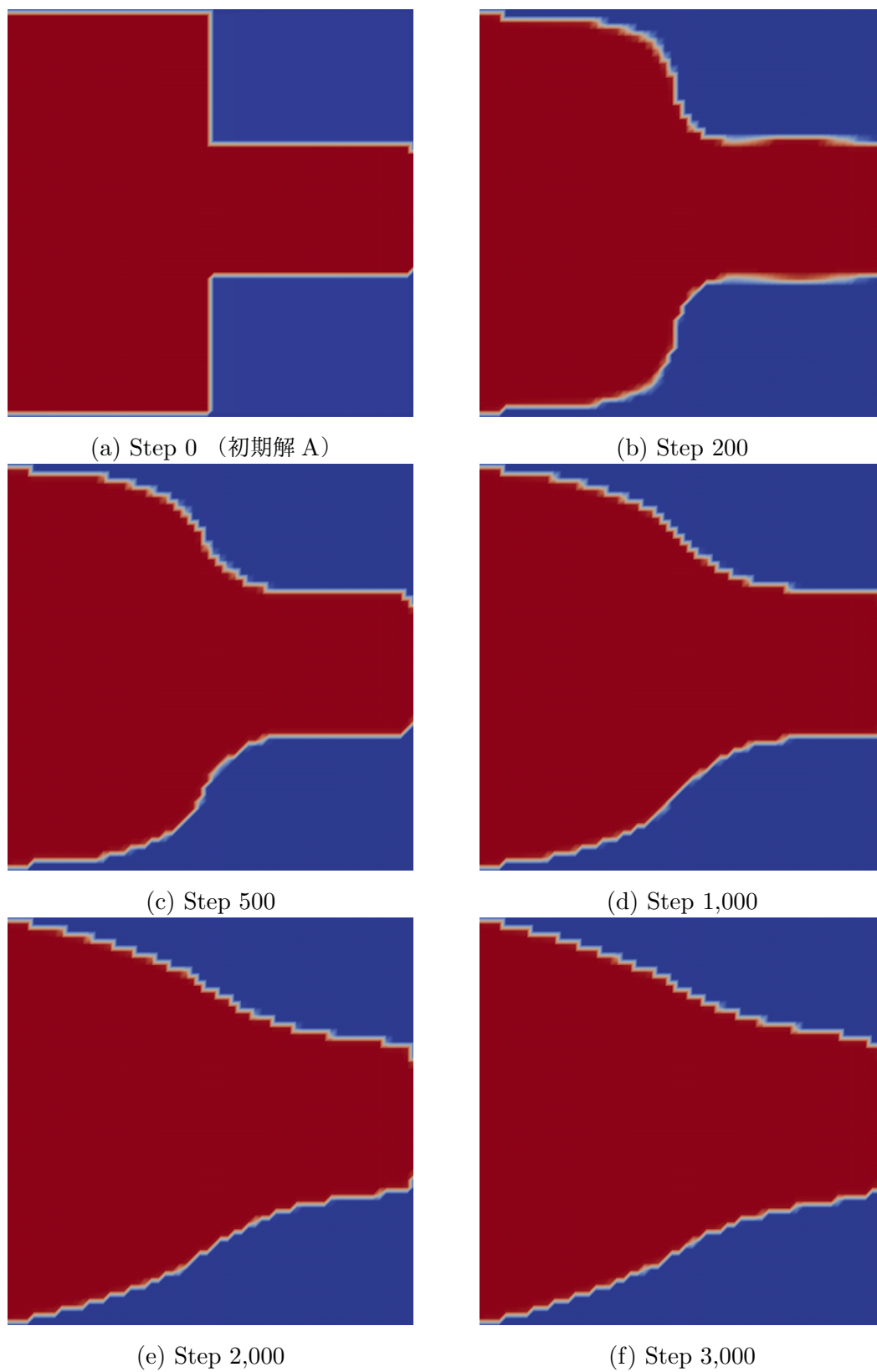
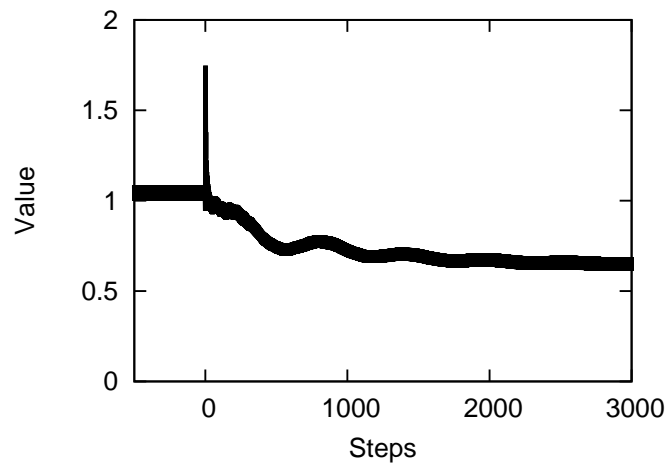


図 6.4. 初期解 A を用いた縮流管問題の最適化の過程における平面 $z = \frac{1}{4}L$ における形状変化の様子 ($Re = 1000$).



(a) 目的関数値.

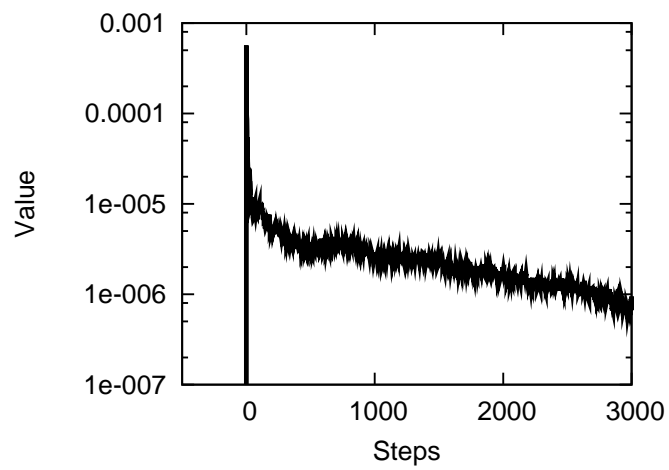
(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$

図 6.5. 初期解 A を用いた縮流管問題の収束の履歴 (Re = 1000).

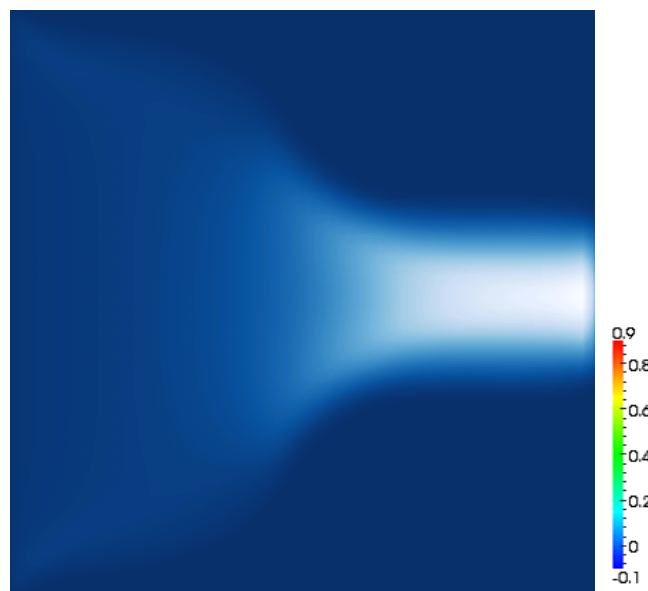
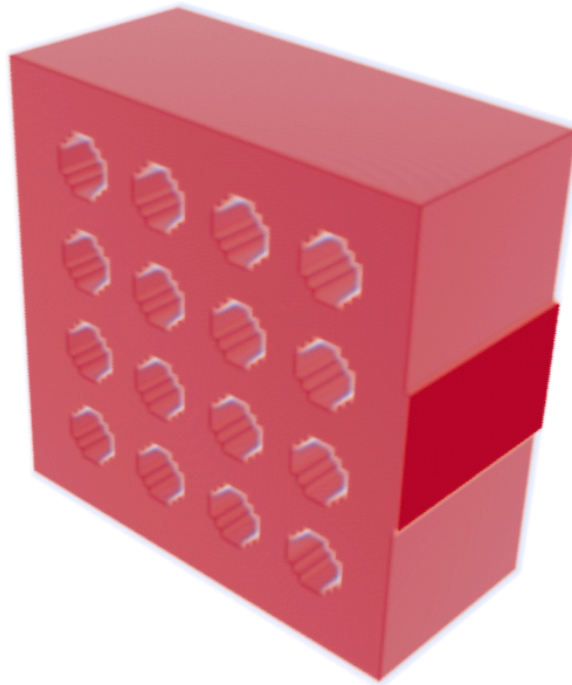
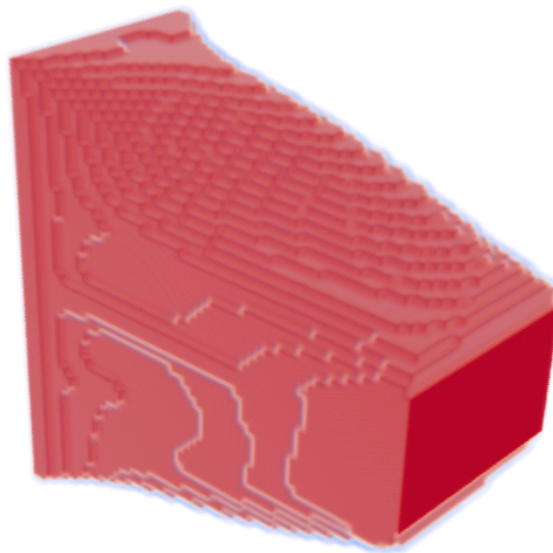


図 6.6. 初期解 A を用いた縮流管問題のステップ 500 における感度 $Re = 1000$.

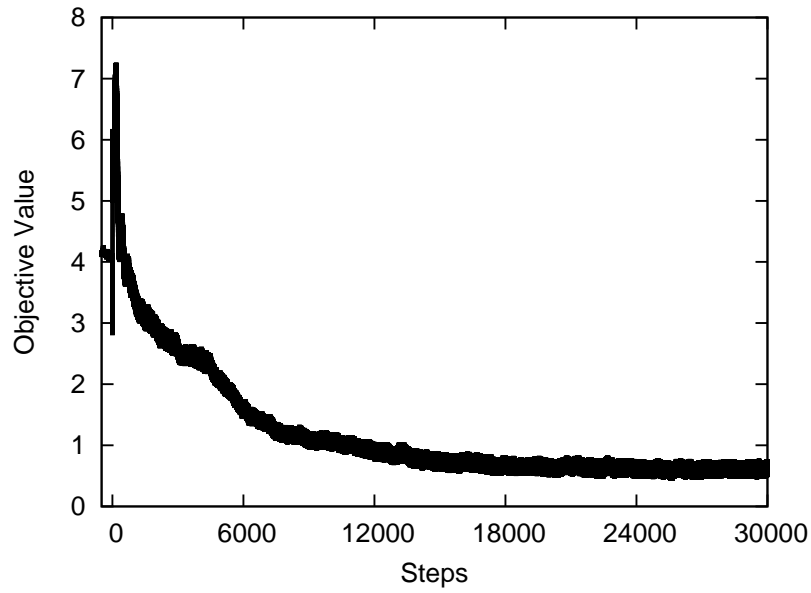


(a) Step 0 (初期解 B)

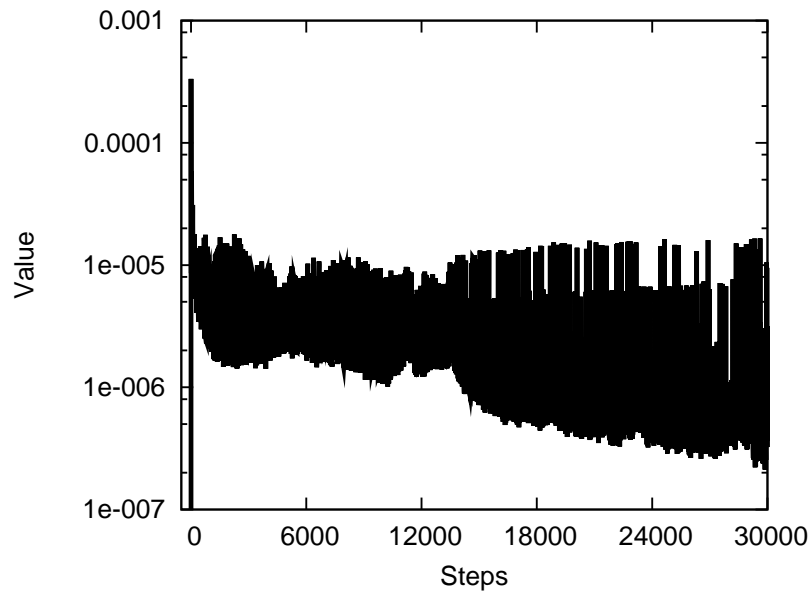


(b) Step 30000

図 6.7. 初期解 B を用いた場合の縮流管問題の形状変化の様子 ($Re = 10$).

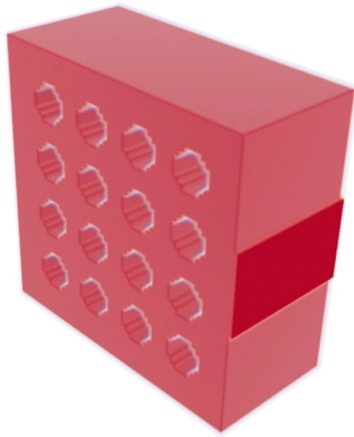


(a) 目的関数値.

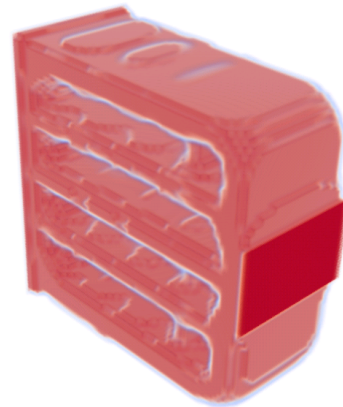


(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$

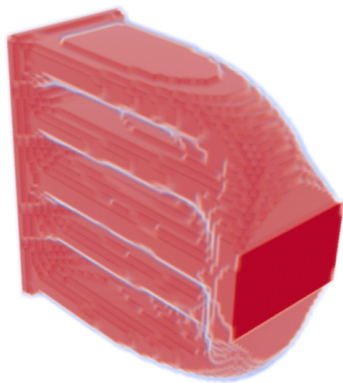
図 6.8. 初期解 B を用いた縮流管問題の収束の履歴.



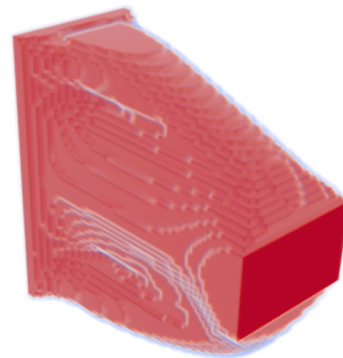
(a) Step 0



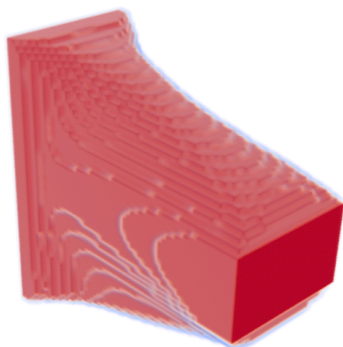
(b) Step 100



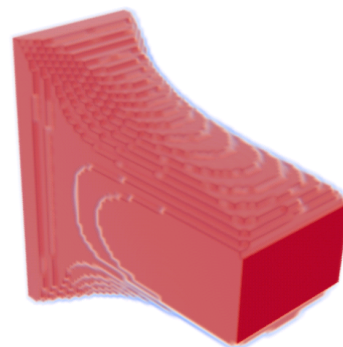
(c) Step 1000



(d) Step 2000



(e) Step 3000



(f) Step 5,000

図 6.9. 縮流管問題の形状変化の様子 ($Re = 10$).

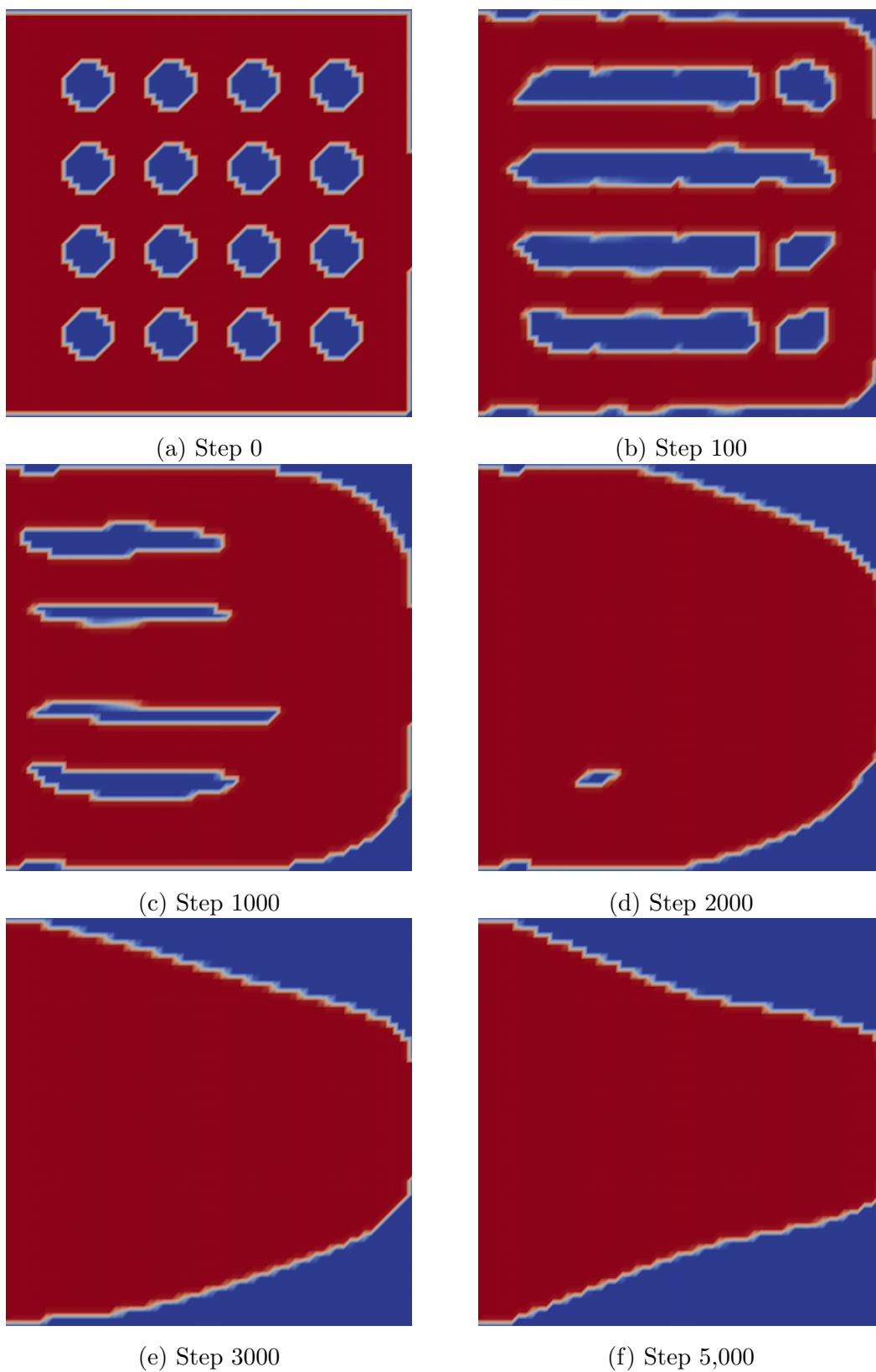
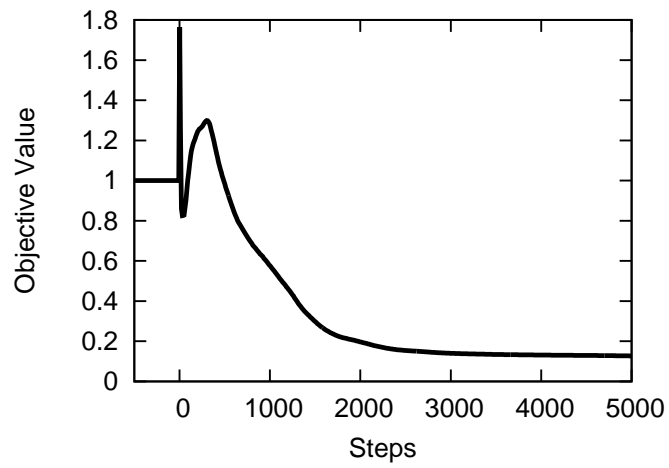


図 6.10. 縮流管問題の平面 $z = \frac{1}{4}L$ における形状変化の様子 ($Re = 10$).



(a) 目的関数値.

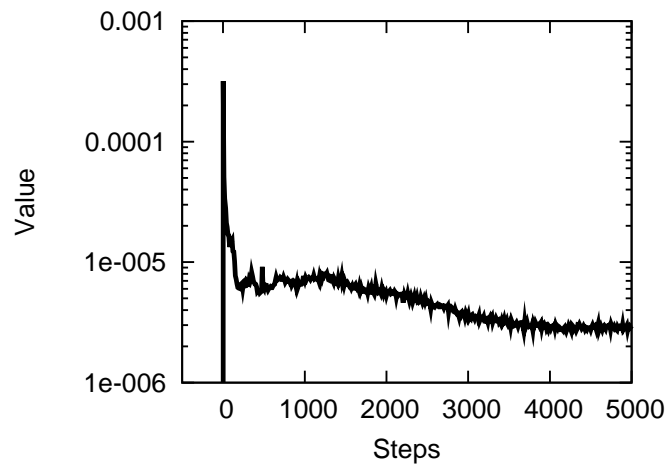
(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$

図 6.11. 縮流管問題の収束の履歴 (Re = 10).

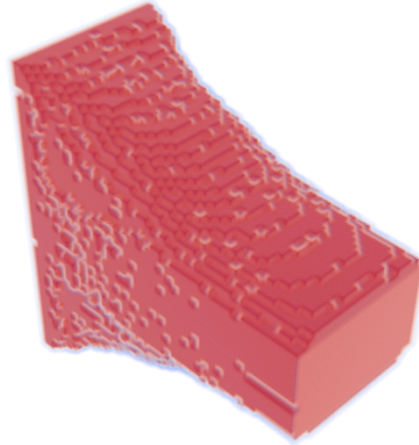
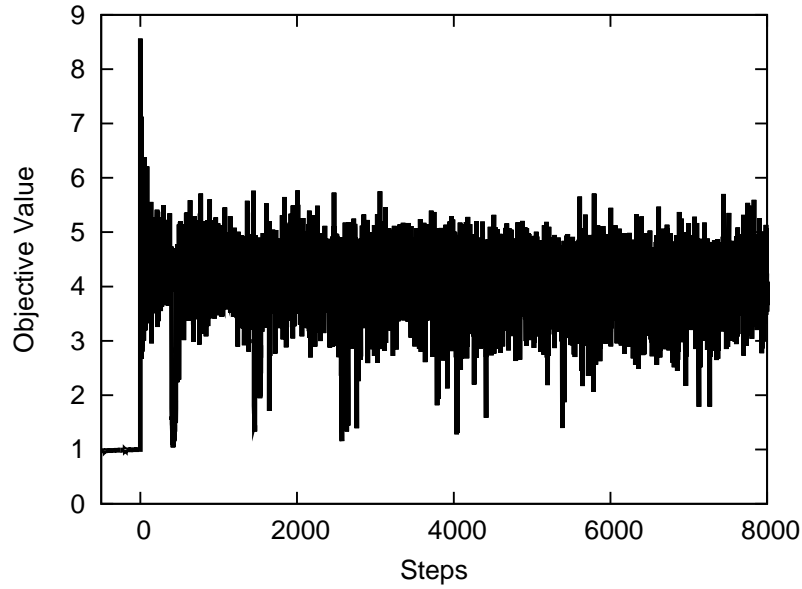


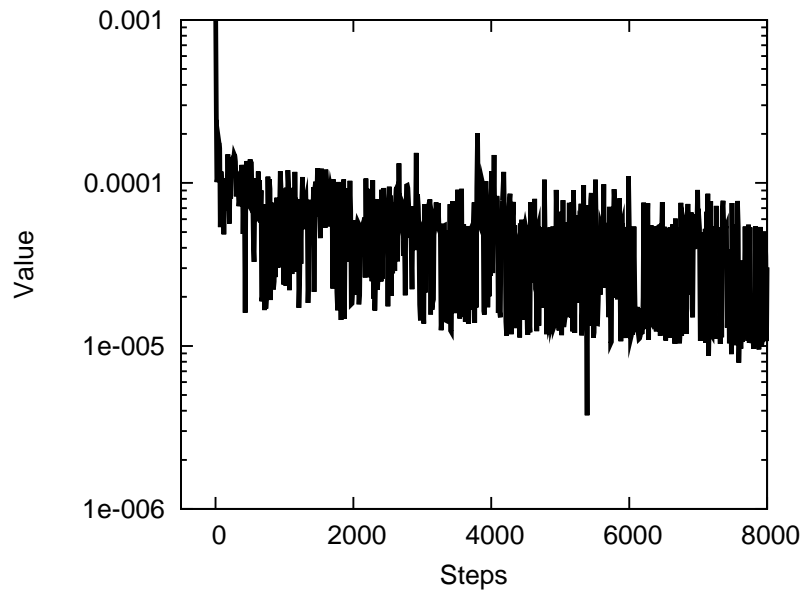
図 6.12. 通常の両側差分作用素を用いた場合の最適解.

6.5.2 微分作用素による結果の違い

本節では、通常の微分作用素を用いた場合と本章で定義した微分作用素を用いた場合の最適化の結果の違いを示す。問題設定は 6.5.1 節の $Re = 1000$ の場合と同じである。設計変数に依存する微分作用素を用いて計算し、妥当な解を得た。本節では通常の両側差分作用素を用いて最適化を行う。結果を図 6.12 に示し、収束の履歴を図 6.13 に示す。収束履歴を見ればわかるように、 ϕ は収束しない。反復を重ねても収束する様子がなかったため、8000 ステップまで計算して停止させた。ステップ 8000 における形状（図 6.12）を見ると、表面に凹凸が存在することがわかる。この凹凸は、力学的にはエネルギー損失を増大させる。この凹凸は何かのきっかけで表面に現れたのち、計算が進むに従って流れに沿って移流し、下流で消える。目的関数が大きく振動して収束しないのはこの凹凸が現れたり消えたりすることに起因している。以上のように、通常の差分作用素を用いると妥当な解を得ることができない。このように凹凸が現れる理由は、前述したように bounce-back 条件における境界の位置を適切に計算できないためと考えられる。



(a) 目的関数値.

(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$ 図 6.13. 通常の両側微分作用素を用いた場合の収束の履歴 ($\text{Re} = 1000$).

6.5.3 合流管問題

本節では四箇所入口と1箇所出口を持つ合流管の問題を最適化する。合流管では流れが合流する部分で大きなエネルギー損失の原因となり得る。最適化の設計領域と境界条件を図6.14に示す。格子点数は x 方向、 y 方向、 z 方向にそれぞれ72点、48点、120点とした。入口境界 Γ_u における速度境界条件は $\mathbf{u} = (\bar{U}, 0, 0)^\top$ とし、出口 Γ_p における圧力境界条件は $p = 0.101325 \text{ MPa}$ とした。 Γ_u および Γ_p 以外の境界条件は非滑り境界条件とした。また体積制約は $\bar{V} = 0.25|\Omega|$ とした。Reynolds数は $\text{Re} = 1000$ とした。Reynolds数の定義 $\text{Re} = \frac{UL}{\nu}$ における代表長さ L は図6.14に示す L とした。

まず、最適解と全く異なる初期解を用いた最適化の結果を示す。図6.15に示す形状を初期解Aとし、これを用いて最適化を行った、最適化の過程における形状の変化の様子を図6.16に示す。最適解は図6.16(f)である。収束の履歴を図6.17に示す。初期解の流体領域の体積は $V = 0.95|\Omega|$ であり、最適解における体積制約を満たしていない。そのため、ステップ0付近での目的関数値は最適解の目的関数値よりも小さい。

ステップ0からステップ5,000までの、最適化の初期においては、設計空間内にいくつもの流路が存在する。この複雑な形状のためにエネルギー損失が大きく、目的関数値が大きい。その後、徐々に流路が統合されていき、最後にはそれぞれの入口から出口まで一本の流路で結ばれる。ステップ10,000以降、形状は最適解に向けて徐々に変形する。 ϕ の変分が $\|\phi_k - \phi_{k-1}\|/|\Omega| \leq 2.0 \times 10^{-7}$ を満たしたため、最適化計算を15,000ステップで停止させた。このとき、目的関数値はわずかに振動するものの、一定の範囲に含まれている。Reynolds数が $\text{Re} = 1000$ であるため、流れ場は層流ではない。

最適解において、四本の流路は出口境界付近で合流しており、合流地点の付近は流れの断面積が大きくなってエネルギー損失を小さくしている。計算時間は156分38秒であった。これは実用上十分に許容できる時間である。

次に最適解に近い形状を初期解として用いて最適化を行った。図6.18(a)に示す形状を初期解Bとし、これを用いて最適化を行った。本節の例題では初期解の状態では体積制約を満たしている。最適化の過程における形状の変化の様子と最適解を図6.18と図6.19に示し、収束の履歴を図6.20に示す。初期解の状態では体積制約を満たしており、それぞれの入口から出口までの流路が形作られているため、目的関数は前回と違っておおよそ単調に減少する。最初は形状が直線的であり曲率が大きい部分があったが、徐々に曲線的な形状になり、曲率も小さくなっていく様子が見て取れる。13,000ステップで停止し、計算時間は140分15秒であった。

本節の例題では、初期解が違っても同じ最適解が得られた。適切な初期解を用いることで短い計算時間で計算を行うことができることがわかる。

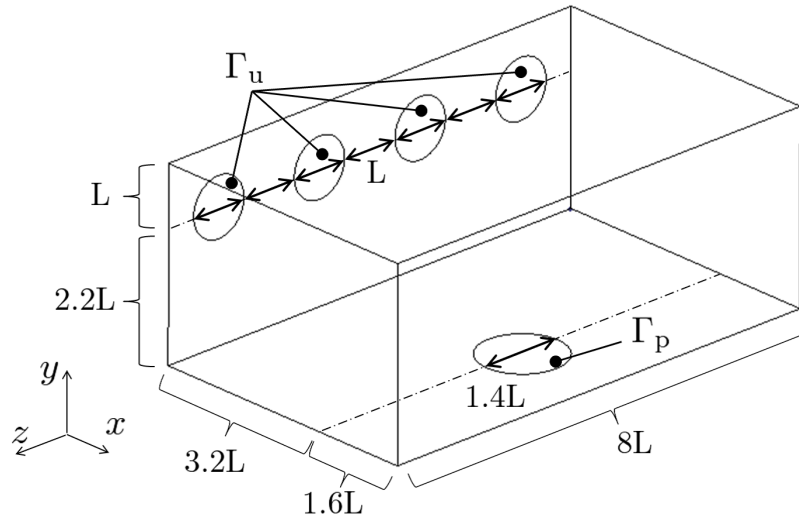


図 6.14. 合流管問題の設計領域と境界条件.

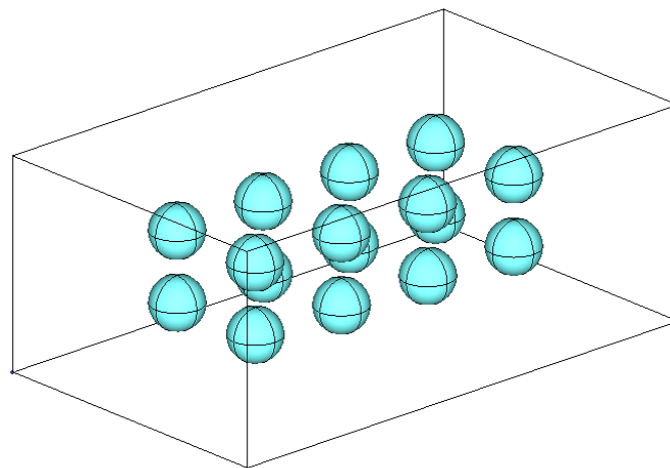
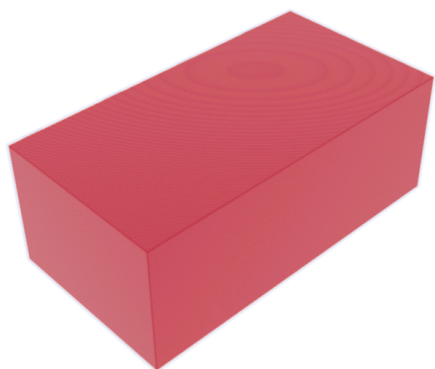


図 6.15. 初期解 A の形状. 内部に 16 個の球が存在する.



(a) Step 0



(b) Step 1,000



(c) Step 5,000



(d) Step 10,000

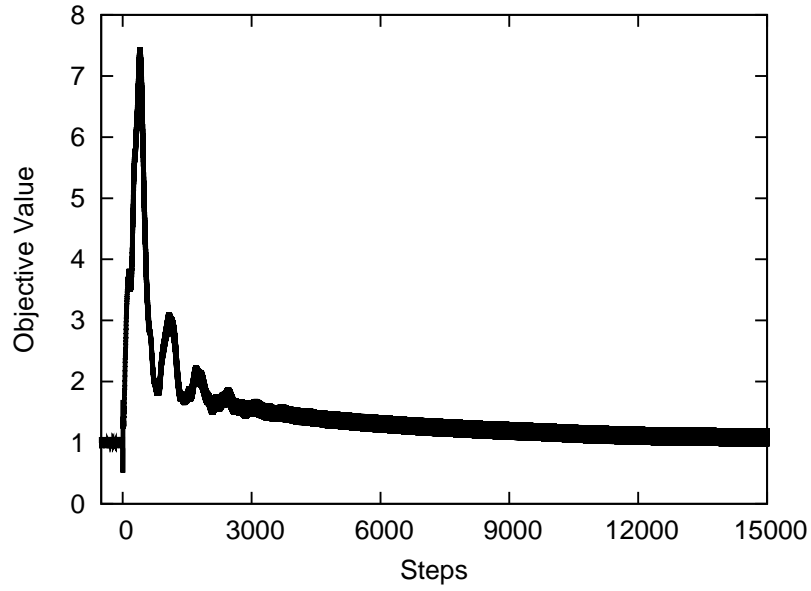


(e) Step 13,000



(f) Step 15,000

図 6.16. 最適化の過程における形状の変化の様子.



(a) 目的関数値.

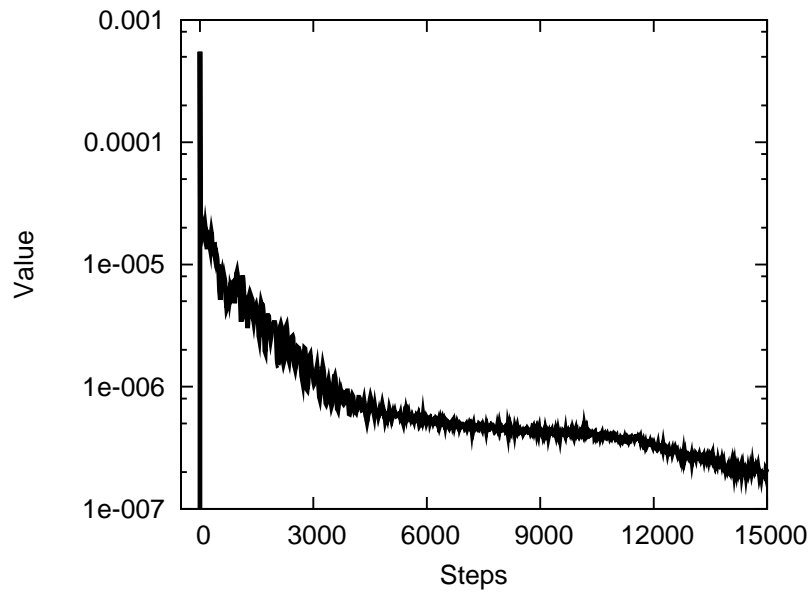
(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$

図 6.17. 合流管問題の収束の履歴 (初期解 A).

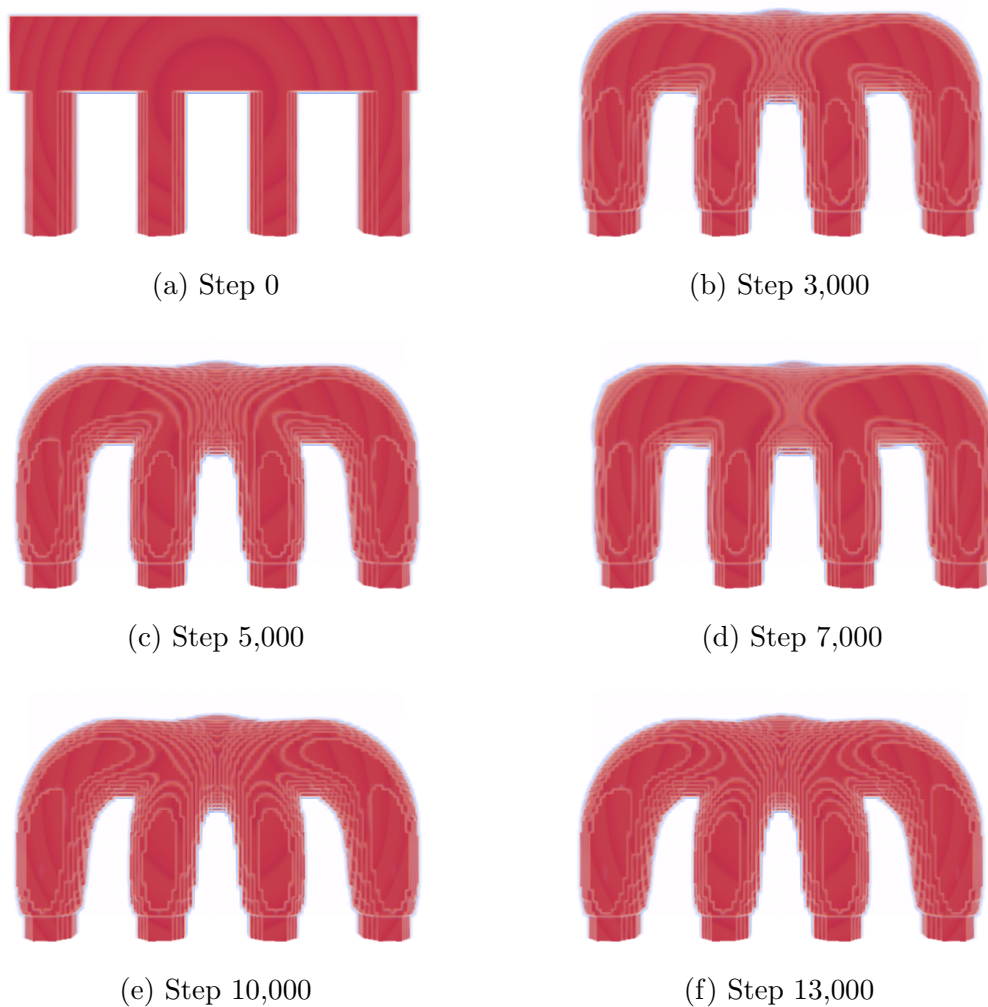


図 6.18. 最適化の過程における形状の変化の様子 (初期解 B).

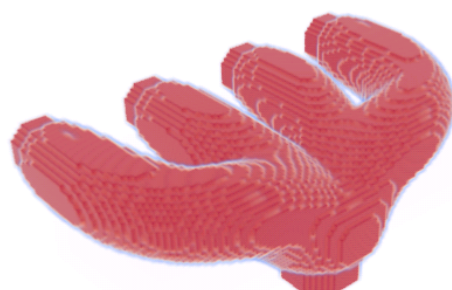
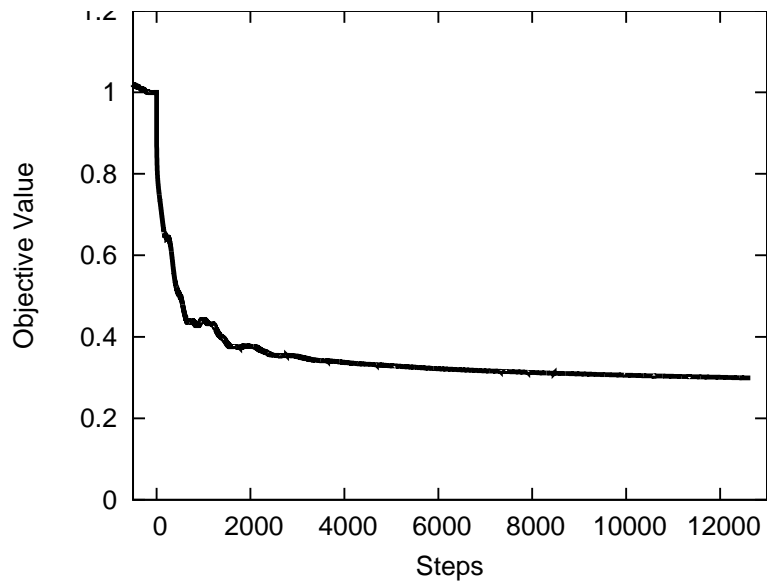


図 6.19. 合流管問題の最適解 (初期解 B).



(a) 目的関数値.

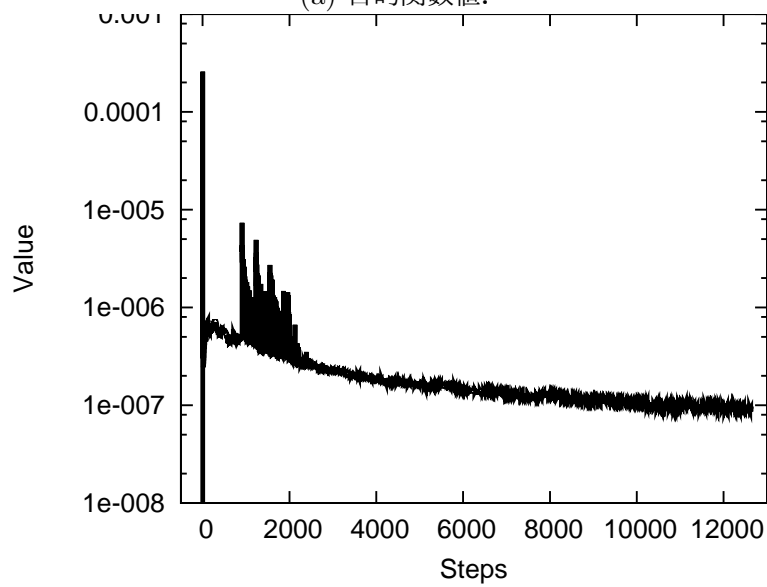
(b) $\|\phi_k - \phi_{k-1}\|/|\Omega|$

図 6.20. 合流管問題の収束の履歴 (初期解 B).

6.6 iSA 法を用いて境界条件を適切に考慮した場合のトポロジー最適化の結論

本章では壁面場の境界条件を適切に考慮した上で流れ場のトポロジー最適化手法を構築した。流れ場の数値計算では壁面境界条件は重要であり、最適化においてもこれを適切に考慮することが重要である。壁面境界条件を考慮するためにはまず壁面の位置を明確に取り扱う必要がある。しかし設計変数は連続関数であり、そのままでは壁面の位置を明確に取り扱えない。そこで新たに 0-1 変数を用いて境界の位置を明確にした。その上で iSA 法に基づいて感度を定式化した最適化法を構築した。iSA 法を用いた場合、LBM の一連の計算の最中に境界が移動するため、固体領域であった格子点が流体領域に変化したり、またその逆が起こる。本章ではそのたびに格子点の情報を適切に初期化する方法を構築した。感度の計算には設計変数による微分が必要であるが、通常の両側差分公式を用いた場合、最適化計算が収束せず、適切な解が得られないことがわかった。非滑り条件として bounce-back 条件を用いた場合、境界は隣接する格子点の midpoint に位置する。この特徴を考慮した上で差分法を構築した。ただし、本手法以外の手法を用いた場合は、このような不具合は報告されておらず、本手法に独特な欠点であることも考えられる。以上を考慮した上で最適化アルゴリズムを構築した。数値計算によりその妥当性を検討し、妥当な解が得られることを示した。また数値計算により、最適化の過程でトポロジーが変化することを確認した。数値計算に要する時間は実用上の観点からも十分短く、iSA 法の利点が活かされていると言える。

第 7 章

時間周期的な境界条件の下での流れ場の最適化

本章では時間周期的な境界条件をもつ流れ場に対するトポロジー最適化問題を定式化し、iSA 法に基づいた解法を提案する。提案手法で用いる感度は随伴変数法を用いて計算し、随伴方程式の解法も提案する。数値計算例により提案手法の妥当性を検証する。また提案手法がある極限において既存手法と一致することを示し、iSA 法と一般的な流れ場の最適化手法の関係を述べる。

7.1 トポロジー最適化における時間周期的な流れ場の取り扱い

流れ場には定常な流れ場と非定常な流れ場が存在するが、本章では非定常な場を取り扱う。流れは Newton の運動方程式に従う非定常な系であり、Navier–Stokes 方程式も Boltzmann 方程式も非定常方程式である。身の周りの流れは渦を含む非定常流れが多く、工学的に興味のある自動車や航空機などの輸送機械、内燃機関などの流れの多くも非定常流れである。このように非定常流れを対象とした最適化手法を構築することは実用上重要である。流れ場はカオス的であるとされ、明確な周期が存在しない。特に Navier–Stokes 方程式は大域的な解の存在が不明であり [55]、そのまま最適化の対象として取り扱うことが困難である。そこで本章では、以下のように時間周期的な境界条件の下での流れ場に限定して手法を構築する。

例えば複数の気筒をもつレシプロエンジンではシリンダポートが交互に開いて燃焼ガスの排気を行う [34]。このとき排気ガスには排気のタイミングに依存する周期が存在する。このとき、細かな現象に着目した場合は周期性は見られない。しかし設計の際にまず着目するのは細かな現象ではなく、全体的に見たときの周期である。つまり、エンジンが一周期分動いたときの流れ場の積分に興味がある。そこで本節では、流れ場に時間周期的な境界条件を仮定し、その一周期分の流れ場のエネルギー損失を目的関数とする。本節で取り扱うような、Reynolds 数が 1000 を超える流れにおいては、流れ場は非周期的な挙動を示すが、本節ではあくまでも境界条件の一周期分を考える。そのうえで、iSA 法の枠組みに基づいて最適化手法を構築する。iSA 法では場の変数と設計変数を並行して解いて収束させることを狙うが、本節のような

対象では、流れ場は収束しない。

数値計算において流れ場の非定常性を取り扱う場合、時間平均をとるか否かで大きく分けて二種類の方法が存在する。時間平均を取る方法の代表的なものとして RANS (Reynolds averaged Navier–Stokes) が挙げられる。RANS は流れ場が完全に周期的であると仮定してその周期の間の時間平均 (Reynolds 平均) を取り、Navier–Stokes 方程式を書き直す手法である。RANS については多くの研究が行われており、そのモデル化も多岐にわたるが、特に RANS モデルを LBM に適用する研究には [16] がある。RANS では支配方程式は Navier–Stokes 方程式とよく似た形の定常方程式になる。そのため第6章と同様のアプローチで解くことができる。時間平均をとらない方法には LES (Large Eddy Simulation) や DNS (Direct Numerical Simulation) などの手法があり、これらは流れ場の非定常性をそのまま数値計算する方法である。LSE では格子解像度以下の渦の影響を粘性項としてモデル化する。このとき、妥当な解を得るためには時間と空間の離散化幅を小さく取る必要があり、RANS による計算に比べて計算量が多くなる傾向にある。DNS は渦をモデル化せずに直接 Navier–Stokes 方程式を解く方法であるが、モデル化をしない分、LES よりもさらに高い格子解像度が必要になるため、一般に LES よりも計算コストが大きい。

LBM は、大規模な流体の非定常数値計算を LBM で実施した例 [62] や、また LBM を実装した市販のソフトウェア [70] に見られるように、本来は非定常な方程式の解法である。この点から時間平均を取らない方法と相性が良いと考えられる。そこで本節では、DNS に比べて計算コストの小さい LES 法を LBM に適用したモデルを用いて最適化手法を構築する。ただし LES モデルはコヒーレント構造 Smagorinsky モデル [33] を用いる。

本章は次のように構成される。まず第7.2節で最適化問題を定義し、iSA 法に基づいた解法を構築する。さらに最適化手法では随伴方程式を解く必要があるため、その随伴方程式の解法を構築する。また最適化アルゴリズムを記述し、必要なメモリ量について考察する。さらに第7.3節で、本章で構築した手法と既存の最適化手法の関係について考察し、両者がある条件の元で等価であることを示す。その後第7.4節で数値計算例を示す。第7.5節ではアルゴリズムを並列実装した時の効率に関して議論し、第7.6節で結論を述べる。

7.2 最適化問題の定式化と iSA 法による解法

7.2.1 最適化問題と iSA 法の適用

ある時間の幅を考えた時の、非定常な流れ場に対して最適化問題を定式化する。第6章では境界条件を適切に考慮した最適化手法を提案した。本節でも引き続き境界条件を適切に考慮して最適化を行う。したがって、設計変数 ϕ は第6章と同じく設計領域 Ω 全体で定義される $0 \leq \phi \leq 1$ を満たす連続変数とする。また χ も同じく次式で定義する：

$$\chi(\phi; \mathbf{x}, t) = \begin{cases} 1 & \text{if } \frac{1}{2} \leq \phi(\mathbf{x}, t), \\ 0 & \text{if } \phi(\mathbf{x}, t) < \frac{1}{2}. \end{cases} \quad (7.1)$$

さらに χ を用いて次のように流体領域 Ω_f と固体領域 Ω_s を判別する：

$$\begin{cases} \mathbf{x} \in \Omega_f & \text{if } \chi(\mathbf{x}, t) = 1, \\ \mathbf{x} \in \Omega_s & \text{if } \chi(\mathbf{x}, t) = 0. \end{cases}$$

さらに $\frac{\partial \chi}{\partial \phi}$ は次式で定義される：

$$\frac{\partial \chi}{\partial \phi} = \delta \left(\phi - \frac{1}{2} \right), \quad (7.2)$$

ただし δ は Dirac のデルタ関数である．数値計算上は (7.2) は次のように離散化される：

$$\frac{\partial \chi}{\partial \phi}(\mathbf{x}) = \begin{cases} \frac{1}{2\delta r} & \text{if } \exists i \text{ such that } (\phi(\mathbf{x}) - \frac{1}{2})(\phi(\mathbf{x} + \mathbf{e}_i) - \frac{1}{2}) \leq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (7.3)$$

場の変数は MRT-LBM を用いて解く．第 6 章までは場の変数は流速ベクトル \mathbf{u} であったが，本章では表記を簡易にするため，分布関数 \mathbf{f} を場の変数とする．さらに， \mathbf{f} の更新は (2.7) に基づいて行うが，これも表記を簡易にするため，

$$\mathbf{f}_{k+1} = \mathcal{F}(\mathbf{f}_k) = \mathcal{F}^{\text{prop}} \circ \mathcal{F}^{\text{col}}(\mathbf{f}_k) \quad (7.4)$$

という表記を用いる．ただし記号 \circ は関数の合成を表し， \mathcal{F}^{col} と $\mathcal{F}^{\text{prop}}$ はそれぞれ衝突操作と移流操作を表し，

$$\mathcal{F}^{\text{col}}(\mathbf{f}_k) = \mathbf{f}^{\text{col}}(\mathbf{x}, t_k) = \mathbf{f}_k - M^{-1}S(M\mathbf{f}_k - \mathbf{m}_k^{\text{eq}}), \quad (7.5)$$

$$\mathcal{F}^{\text{prop}}(\mathbf{f}_k) = (f_0(\mathbf{x}, t_{k+1}), \dots, f_{18}(\mathbf{x}, t_{k+1}))^\top, \quad (7.6)$$

$$f_i(\mathbf{x}, t_{k+1}) = f_i(\mathbf{x} - \mathbf{e}_i, t_k)$$

により定義される．

次に目的関数を定義する．流れ場の周期を \hat{T} と仮定し，目的関数はエネルギー損失を一周分，すなわち時刻 $t = T$ から $t = T + \hat{T}$ まで積分したものとする． $T = K\delta t$ ， $\hat{T} = m\delta t$ ， $m \in \mathbb{Z}$ とすると，目的関数は

$$D(\mathbf{u}; \phi, T) = \int_T^{T+\hat{T}} \int_{\Omega_f} \frac{1}{2} \nu \|\varepsilon(\mathbf{u}(\mathbf{x}, t))\|^2 d\mathbf{x} dt$$

と定式化される．最適化の目的は D を最小化する ϕ を求めることであるから，解くべき最適化問題は次のように定式化される：

$$(\mathbf{P}_{\text{NS}}'') : \min_{\phi, \mathbf{u}} D(\mathbf{u}, \phi) \quad (7.7a)$$

$$\text{s. t. } \mathbf{u} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = -\text{grad} p + \nu \Delta \mathbf{u}, \quad (7.7b)$$

$$0 \leq \phi(\mathbf{x}, t) \leq 1, \quad (7.7c)$$

$$\int_{\Omega} \phi(\mathbf{x}, t) d\mathbf{x} = \bar{V}. \quad (7.7d)$$

最適化問題 (P''_{NS}) を, iSA 法を用いて近似的に解く. 第3章と第4章における最適化問題の関係と同様に, (3.2) は (7.7) に対応している.

iSA 法を用いて (P''_{NS}) を解く. 以降では時間と空間について離散化したものを考え, (1.1) の定義にしたがって記述する. 目的関数は, 時間と空間について離散化すると,

$$D_K(\mathbf{u}; \phi) = \sum_{k=0}^m \sum_{\mathbf{x} \in \hat{\Omega}} \frac{1}{2} \nu \|\varepsilon(\mathbf{u}_{K+k})\|^2 \delta r \delta t, \quad (7.8)$$

となる. ここで D_K は K をパラメータとする関数であることに注意する. 本節において (3.16) に対応する最適化問題を書き下すと, 次のようになる:

$$(P_{\text{cyclic}}) \quad \min_{\phi_k, \mathbf{u}_k} D(\mathbf{u}_k; \phi_k), \quad (7.9a)$$

$$\text{s. t.} \quad \mathbf{u}_k = \mathbf{u}_{\text{LBM}}(\mathbf{u}_{k-1}, \phi_k), \quad (7.9b)$$

$$0 \leq \phi_k(\mathbf{x}_i) \leq 1, \quad \forall \mathbf{x}_i \in \hat{\Omega} \quad (7.9c)$$

$$\sum_{\mathbf{x} \in \hat{\Omega}} \phi_k(\mathbf{x}) \delta r = V_k. \quad (7.9d)$$

以降では iSA 法に基づいて設計変数 ϕ と場の変数 \mathbf{u} を同時に更新し, (P_{cyclic}) を近似的に解く.

k に依存する Lagrange 関数を次式で定義する:

$$\begin{aligned} \mathcal{L}_k = & D_{km} - \sum_{i=0}^n \sum_{j=1}^m \langle f_{i,km+j+1} - \mathcal{F}_i(\mathbf{f}_{km+j}), g_{i,km+j} \rangle \\ & - \left(\sum_{\mathbf{x} \in \hat{\Omega}} \phi_k(\mathbf{x}) \delta r - V_k \right) M_k. \end{aligned}$$

ただし \mathbf{v}_k と $\lambda^1, \lambda^2, M_k$ は未定乗数である. また $\langle \cdot, \cdot \rangle$ は内積を表し, $f_{i,k}$ は, k ステップ目

における f_i を意味する. \mathcal{L}_k を用いて, 以下のように感度が定義される:

$$\begin{aligned}
\frac{\partial \mathcal{L}_k}{\partial \phi_k} &= \sum_{j=1}^m \frac{\partial D}{\partial \mathbf{f}_{km+j}} \frac{\partial \mathbf{f}_{km+j}}{\partial \phi_k} + \frac{\partial D}{\partial \phi_k} \\
&\quad - \sum_{i=0}^n \sum_{j=1}^m \left\langle \frac{\partial f_{i,km+j+1}}{\partial \phi_k}, g_{i,km+j} \right\rangle \\
&\quad + \sum_{i=0}^n \sum_{j=1}^m \left\langle \frac{\partial \mathcal{F}_i(\mathbf{f}_{km+j})}{\partial f_{i,km+j}} \frac{\partial f_{i,km+j}}{\partial \phi_k}, g_i^{km+j} \right\rangle \\
&\quad + \sum_{i=0}^n \sum_{j=1}^m \left\langle \frac{\partial \mathcal{F}_i(\mathbf{f}_{km+j})}{\partial \phi_k}, g_{i,km+j} \right\rangle - M(t) \\
&= \sum_{i=0}^n \sum_{j=1}^m \left\langle \left(\frac{\partial D}{\partial \mathbf{f}_{km+j}} - g_{i,km+j-1} \right. \right. \\
&\quad \left. \left. + \left\langle \frac{\partial \mathcal{F}_i(\mathbf{f}_{km+j})}{\partial f_{i,km+j}}, g_{i,km+j} \right\rangle \right), \frac{\partial f_{i,km+j}}{\partial \phi_k} \right\rangle + \frac{\partial D}{\partial \phi_k} \\
&\quad + \sum_{i=0}^n \sum_{j=1}^m \left\langle \frac{\partial \mathcal{F}_i(\mathbf{f}_{km+j})}{\partial \phi_k}, g_{i,km+j} \right\rangle - M(t). \tag{7.10}
\end{aligned}$$

第 6 章までは直接法により感度を計算した. 直接法を用いるためには $\frac{\partial f_{i,km+j}}{\partial \phi_k}$ を計算する必要があるが, これは一般に計算が困難である. その場合には第 2 章で述べたように随伴変数法が有効である. (7.10) では $\frac{\partial f_{i,km+j}}{\partial \phi_k}$ の係数部分が零になるように g_i を決定する. すなわち $\mathbf{g} = (g_0, g_1, \dots, g_N)^\top$ とし,

$$\frac{\partial D}{\partial \mathbf{f}_{km+j}} - \mathbf{g}_{km+j-1} + \left\langle \frac{\partial \mathcal{F}(\mathbf{f}_{km+j})}{\partial \mathbf{f}_{km+j}}, \mathbf{g}_{km+j} \right\rangle = 0, \quad (j = 0, \dots, m) \tag{7.11}$$

を満たすように \mathbf{g} を決めることで, $\frac{\partial f_{i,km+j}}{\partial \phi_k}$ を計算する必要がなくなる. (7.11) は随伴方程式と呼ばれ, \mathbf{g} は随伴変数と呼ばれる. (7.11) の計算方法は第 7.2.2 節で述べる. (7.11) を満たす \mathbf{g} を用いることで, (7.10) は次のように変形できる:

$$\frac{\partial \mathcal{L}}{\partial \phi_k(\mathbf{x}_i)} = \frac{\partial D}{\partial \phi_k(\mathbf{x}_i)} + \sum_{j=1}^m \left\langle \frac{\partial \mathcal{F}(f_{i,km+j})}{\partial \phi_k(\mathbf{x}_i)}, g_{i,km+j} \right\rangle. \tag{7.12}$$

場の変数 \mathbf{u} と設計変数 ϕ は次の式を用いて更新する:

$$\mathbf{u}(\mathbf{x}, t_{km+1}) = \mathbf{u}_{\text{LBM}}(\mathbf{f}(\mathbf{x}, t_{km}), \phi_{k+1}(\mathbf{x})), \tag{7.13a}$$

$$\phi_{k+1}(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}} D + M_{k+1} \leq 0, \\ \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}} D + M_{k+1} & \text{if } 0 < \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}} D + M_{k+1} < 1, \\ 1 & \text{if } \phi_k(\mathbf{x}) - \kappa \nabla_{\phi_{k+1}} D + M_{k+1} \geq 1. \end{cases} \tag{7.13b}$$

ただし M_k はアルゴリズム 2 を用いて計算する. $\kappa > 0$ は収束性を調整するための定数である. 数値計算においては κ は $\|\phi_1 - \phi_0\|_\infty = 0.1$ を満たすように設定した.

次に (7.12) を LBE により離散化して定式化する。 $\frac{\partial \mathcal{F}(\mathbf{f}_{km+j})}{\partial \phi}$ は $f_i = \mathcal{F}_i(\mathbf{f}_{km+j})$ であるから、

$$\frac{\partial \mathcal{F}(\mathbf{f}_{km+j})}{\partial \phi_k(\mathbf{x})} = \sum \frac{\partial f_{i,km+j+1}(\mathbf{x} + \mathbf{e}_i)}{\partial \chi_{km+j}(\mathbf{x})} \frac{\partial \chi}{\partial \phi_k(\mathbf{x})} = (f_{i,km+j}^{\text{col}}(\mathbf{x}) - f_{i,km+j}^{\text{col}}(\mathbf{x} - \mathbf{e}_i)) \frac{\partial \chi}{\partial \phi} \quad (7.14)$$

が成り立つ。

差分法については第6章で述べた通り、 χ に依存する差分演算子を用いることが自然である。したがって $\frac{\partial D}{\partial \phi}$ は

$$\frac{\partial D}{\partial \phi} = \frac{\partial D}{\partial \chi} \frac{\partial \chi}{\partial \phi} \quad (7.15)$$

$$= -\frac{1}{2}\nu \left(\left\langle \frac{\partial \Delta_\chi \mathbf{u}}{\partial \chi}, \mathbf{u} \right\rangle + \left\langle \Delta_\chi \mathbf{u}, \frac{\partial \mathbf{u}}{\partial \chi} \right\rangle + \left\langle \text{rot} \mathbf{u}, \frac{\partial (\text{rot}_\chi \mathbf{u})}{\partial \chi} \right\rangle \right) \frac{\partial \chi}{\partial \phi} \quad (7.16)$$

と定式化される。ここで差分演算子 $\frac{\partial \Delta_\chi \mathbf{u}}{\partial \chi}$, $\frac{\partial (\text{rot}_\chi \mathbf{u})}{\partial \chi}$ は第6.4.1節で導いたものを用いる。(7.14) および (7.16) と、第7.2.2節で導く \mathbf{g} より、(7.12) を計算することができる。

7.2.2 随伴方程式の解法

本節では (7.11) で導いた随伴方程式の解法を述べる。まず $\frac{\partial \mathcal{F}(\mathbf{f})}{\partial \mathbf{f}}$ は (7.4) を用いて次のように分解される：

$$\frac{\partial \mathcal{F}(\mathbf{f})}{\partial \mathbf{f}} = \mathcal{F}^{\text{prop}} \circ \frac{\partial \mathbf{f}^{\text{col}}}{\partial \mathbf{f}}$$

ここで $\mathcal{F}^{\text{prop}}$ は移流操作であり、(7.6) からわかるように \mathbf{f} に依存しない。(7.5) より、 $\frac{\partial \mathbf{f}^{\text{col}}(t)}{\partial \mathbf{f}}$ は

$$\frac{\partial \mathbf{f}^{\text{col}}}{\partial f_i} = \mathbf{I}_i - M^{-1} S \left(M + \frac{\partial \mathbf{m}^{\text{eq}}}{\partial f_i} \right),$$

と変形される．ただし $\mathbf{I}_i \in \mathbb{R}^n$ は i 番目の要素が 1 でそれ以外の要素が 0 であるようなベクトルを示す．続いて $\frac{\partial m^{\text{eq}}}{\partial f_i}$ は

$$\begin{aligned} \frac{\partial m_0^{\text{eq}}}{\partial f_i} &= 1, \quad \frac{\partial m_1^{\text{eq}}}{\partial f_i} = -11 + 19 \frac{1}{\rho_0} 2\mathbf{u}^\top \frac{\partial \mathbf{u}}{\partial f_i}, \quad \frac{\partial m_2^{\text{eq}}}{\partial f_i} = 3 - \frac{11}{2} \frac{1}{\rho_0} 2\mathbf{u}^\top \frac{\partial \mathbf{u}}{\partial f_i}, \\ \frac{\partial m_3^{\text{eq}}}{\partial f_i} &= \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x, \quad \frac{\partial m_4^{\text{eq}}}{\partial f_i} = -\frac{2}{3} \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x, \quad \frac{\partial m_5^{\text{eq}}}{\partial f_i} = \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y, \\ \frac{\partial m_6^{\text{eq}}}{\partial f_i} &= -\frac{2}{3} \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y, \quad \frac{\partial m_7^{\text{eq}}}{\partial f_i} = \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z, \quad \frac{\partial m_8^{\text{eq}}}{\partial f_i} = -\frac{2}{3} \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z, \\ \frac{\partial m_9^{\text{eq}}}{\partial f_i} &= \frac{1}{\rho_0} \left(6u \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x - 2\mathbf{u}^\top \frac{\partial \mathbf{u}}{\partial f_i} \right), \\ \frac{\partial m_{10}^{\text{eq}}}{\partial f_i} &= -0.5 \frac{1}{\rho_0} \left(6u \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x - 2\mathbf{u}^\top \frac{\partial \mathbf{u}}{\partial f_i} \right), \\ \frac{\partial m_{11}^{\text{eq}}}{\partial f_i} &= \frac{1}{\rho_0} \left(v \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y - w \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z \right), \\ \frac{\partial m_{12}^{\text{eq}}}{\partial f_i} &= -0.5 \frac{1}{\rho_0} \left(v \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y - w \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z \right), \\ \frac{\partial m_{13}^{\text{eq}}}{\partial f_i} &= \frac{1}{\rho_0} \left(u \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y + v \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x \right), \\ \frac{\partial m_{14}^{\text{eq}}}{\partial f_i} &= \frac{1}{\rho_0} \left(v \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z + w \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_y \right), \\ \frac{\partial m_{15}^{\text{eq}}}{\partial f_i} &= \frac{1}{\rho_0} \left(w \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_x + u \left(\frac{\partial \mathbf{u}}{\partial f_i} \right)_z \right), \\ \frac{\partial m_{16}^{\text{eq}}}{\partial f_i} &= 0, \quad \frac{\partial m_{17}^{\text{eq}}}{\partial f_i} = 0, \quad \frac{\partial m_{18}^{\text{eq}}}{\partial f_i} = 0 \end{aligned}$$

で与えられる．任意のベクトル \mathbf{v} に対して $(\mathbf{v})_x$, $(\mathbf{v})_y$, $(\mathbf{v})_z$ は、それぞれ x , y , z 方向の要素を示す．

結果として (7.11) は衝突操作と移流操作の組み合わせで表される．

$$\text{Collision step} \quad \mathbf{g}^{\text{col}}(\mathbf{x}, t) = \mathbf{g}(t) - M^{-1} S \left(M + \frac{\partial m^{\text{eq}}}{\partial \mathbf{f}} \mathbf{g} \right) + \frac{\partial D}{\partial \mathbf{f}(t)}, \quad (7.17a)$$

$$\text{Propagate step} \quad \mathbf{g}_i(\mathbf{x}, t) = \begin{cases} \mathbf{g}_i^{\text{col}}(\mathbf{x} + \mathbf{e}_i, t_{k+1}) & \text{if } \chi_{\mathbf{x}+\mathbf{e}_i} = 1, \\ \mathbf{g}_{-i}^{\text{col}}(\mathbf{x}, t_{k+1}) & \text{if } \chi_{\mathbf{x}+\mathbf{e}_i} = 0. \end{cases} \quad (7.17b)$$

ここで、移流操作は bounce-back 条件を考慮している．移流操作では、時刻 t_{k+1} における \mathbf{g} の値を、 t_k における \mathbf{g} の値に移流する．すなわち、時間の進行方向に逆行して移流することができる．これは随伴変数法の特徴である．この特徴に関連して、LBE と随伴方程式の時間発展の様子を図 7.1 に示す．

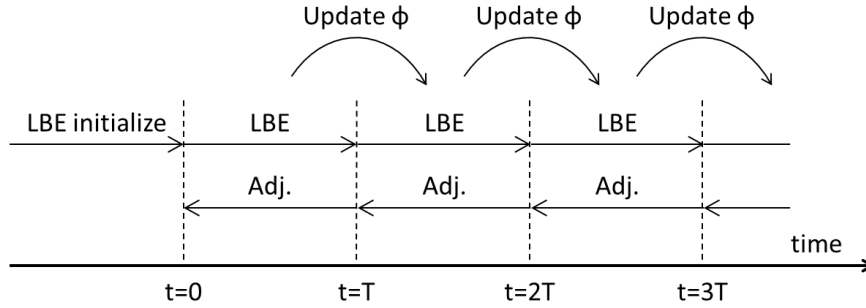


図 7.1. LBM と随伴方程式を解く際の概念図.

次に (7.17a) における $\frac{\partial D}{\partial f_i}$ は連鎖側により, 次のように分解される:

$$\begin{aligned} \frac{\partial D}{\partial f_i} &= \frac{\partial D}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial f_i} \\ &= \frac{\mathbf{u} - \mathbf{c}_i}{\rho} \sum_{k=0}^m \sum_{\mathbf{x} \in \Omega_f} \nu \varepsilon(\mathbf{u}_{K+k}) : \frac{\partial \varepsilon(\mathbf{u}_{K+k})}{\partial \mathbf{u}} \delta r \delta t. \end{aligned} \quad (7.18)$$

ただし (7.8) と $\frac{\partial \mathbf{u}}{\partial f_i} = \frac{\mathbf{u} - \mathbf{c}_i}{\rho}$ を用いた. 以上を用いて随伴方程式を解くことができる.

7.2.3 最適化アルゴリズム

本節では前節までの内容を整理して最適化アルゴリズムを記述する. 最適化では (7.12) を数値的に解く. 本章の手法でも, LBM の計算の過程で固体領域と流体領域の位置が変化するため, ある時点で固体領域であった格子点が, 次のステップでは流体領域に変化したり, またその逆が起こる. そこで第6章と同様に, 第6.3節で構築した手法を用いて場の変数を初期化する. アルゴリズムをアルゴリズム 6 に示す.

アルゴリズム 6.

- Step 1. $k = 0$ とし, 適当な値を用いて \mathbf{f}_0 と ϕ_0 を初期化する.
- Step 2. $\phi = \phi_k$ とし, LBM(2.7) を用いて \mathbf{f}_k と \mathbf{u}_k ($k = 1, 2, \dots, m$) を計算する.
- Step 3. \mathbf{u}_k と (7.18), (7.16) より $\frac{\partial D}{\partial f_i}$ と $\frac{\partial D}{\partial \phi}$ を計算する.
- Step 4. \mathbf{f}_k ($k = 1, 2, \dots, m$) と (7.17), (7.18) より \mathbf{g}_k ($k = m, m-1, \dots, 1$) を計算する.
- Step 5. \mathbf{f}_k と \mathbf{g}_k ($k = 1, 2, \dots, m$), $\frac{\partial D}{\partial \phi}$, (7.12) により感度を計算する.
- Step 6. (7.13) により ϕ_{k+1} を計算する.
- Step 7. 第6.3節の内容を用いて場の変数を初期化する.
- Step 8. 収束判定を満たしていれば停止.
- Step 9. $k = k + 1$ として Step 2 へ.

本アルゴリズムでは本質的に非定常な流れを扱う. 周期性を仮定しているが, 数値計算上は完

全な周期性が得られることは稀である。そのため、場の変数 \mathbf{u} の収束は期待できない。したがって収束判定は設計変数 ϕ のみにして行い、

$$\|\phi_k - \phi_{k-1}\|_\infty \leq \delta_\phi$$

を用いる。

アルゴリズム 6 の Step 4 では、 \mathbf{g}_k の計算のために \mathbf{f}_k と \mathbf{u}_k の情報が必要である。したがって、一回の最適化反復で感度を計算するにあたって、LBM で解いた情報を保存しておく必要がある。したがって、周期 m ステップ分、 \mathbf{f} の次元 n 、 \mathbf{u} の次元 \dim 、格子点数 N に対して $Nm(n + \dim)$ 個の実数を記憶するメモリが必要である。例として $m = 1000$ 、 $n = 19$ 、 $\dim = 3$ 、 $N = 10^6$ とし倍精度計算をすると仮定すると、必要なメモリ量は約 170GB になる。これは一般に普及している計算機のメモリを超えているため、現実的には並列計算をすることが必須である。

7.3 iSA 法と既存手法の関係

アルゴリズム 6 は、場の変数と設計変数を同時に更新しており、iSA 法に基づいている。一方で、感度を定義する (7.12) は、ある仮定のもとで既存手法における感度の定義と一致する。本節ではこれを示す。前述の定式化では流れの周期 \hat{T} に対して m を $\hat{T} = m\delta t$ としたが、ここでは m を $\frac{\hat{T}}{\delta t}$ の定数倍として、 $m \rightarrow \infty$ の場合を考える。つまり場の変数が周期 $\hat{T} < \infty$ を持っているとして仮定する。この仮定は、Reynolds 数が高い流れに対しては必ずしも妥当ではなく、Reynolds 数が低い場合には満たされる場合がある。本節では iSA 法と既存手法の関係を議論するために前述の仮定をおく。従って、アルゴリズム 6 の Step 2 で場の変数を解くが、 \mathbf{f} は最初は周期が定まっていなくても、いずれ周期 \hat{T} の周期的な流れに収束する。これは正確には、改めて $\hat{T} = \hat{m}\delta t$ で定義される \hat{m} を用いて、

$$\mathbf{f}_{K+k\hat{m}+j} = \mathbf{f}_{K+k\hat{m}+j} \quad (7.19)$$

を満たす十分大きな $K \in \mathbb{Z}$ が存在する、と言える。このとき、感度 (7.12) は

$$\frac{\partial \mathcal{L}}{\partial \phi_k(\mathbf{x})} = \sum_{j=1}^m \frac{\partial d}{\partial \phi_k(\mathbf{x})} + \sum_{j=1}^m \left\langle \frac{\partial \mathcal{F}(f_i(t_{km+j}))}{\partial \phi_k(\mathbf{x})}, g_i(t_{km+j}) \right\rangle \quad (7.20)$$

となる。両辺を m で割って \hat{m} を乗じ、 $m \rightarrow \infty$ の極限を取ると、

$$\begin{aligned} \lim_{m \rightarrow \infty} \frac{\hat{m}}{m} \frac{\partial \mathcal{L}}{\partial \phi_k(\mathbf{x})} &= \lim_{m \rightarrow \infty} \frac{\hat{m}}{m} \sum_{j=1}^m \left(\frac{\partial D}{\partial \phi_k(\mathbf{x})} + \left\langle \frac{\partial \mathcal{F}(f_{i,km+j})}{\partial \phi_k(\mathbf{x})}, g_{i,km+j} \right\rangle \right) \\ &= \sum_{j=1}^{\hat{m}} \left(\frac{\partial D}{\partial \phi_k(\mathbf{x})} + \left\langle \frac{\partial \mathcal{F}(f_{i,K+k\hat{m}+j})}{\partial \phi_k(\mathbf{x})}, g_{i,K+k\hat{m}+j} \right\rangle \right) \end{aligned} \quad (7.21)$$

とできる。

ここで、(7.19) で示される場の変数は、周期 \hat{T} を持つ場の方程式の定常解である。このような解は、例えば流れ場の方程式では RANS (Reynolds Average Navier–Stokes) モデルの根拠

となっている [11]. 特に $\hat{m} = 1$ の場合は通常の変常解を表す. さらに $\hat{m} = 1$ の場合, (7.21) で表される感度は, 既存手法で随伴方程式を用いる場合に得られる感度と等価である. 特に [79] では場の支配方程式として LBM を用いており, 表記の上でも一致する. また $K = 0$ であれば,

$$\lim_{m \rightarrow \hat{m}} \frac{\partial \mathcal{L}}{\partial \phi_k(\mathbf{x})} = \lim_{m \rightarrow \infty} \frac{\hat{m}}{m} \frac{\partial \mathcal{L}}{\partial \phi_k(\mathbf{x})} \quad (7.22)$$

であり, (7.12) で定義される感度は極限を取ることなく, 既存手法の感度と一致する. $K = 0$ とはすなわち, 場の変数が周期的になっていることを示しており, この感度により ϕ が更新されなくなればそれが最適解であり, その最適性は既存手法の最適性と等価である.

7.4 数値計算例

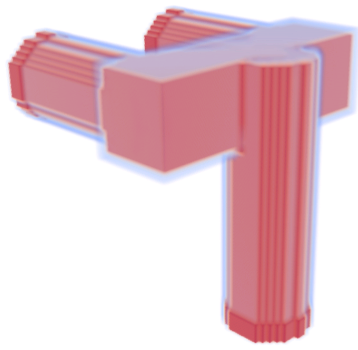
本節では集合管問題を例として数値計算例を示す. 数値計算は XEON X5690 (3.33GHz with 8GB memory) で実施した. アルゴリズムは C++ により実装し, Intel MPI を使用して並列化した. 二か所の入口と一か所の出口を持つような集合管問題を考える. 例えば二気筒のレシプロエンジンでは, シリンダの排気が二か所から排出され, その排気は一本の排気管にまとめられて排出される. このときシリンダポートは交互に開閉を繰り返し, 排気も周期的になる. 本例題はこの様子を模擬する. まず, 入口の速度境界条件は時間に依存した形で

$$\begin{aligned} U_1 &= \bar{U} (0.5 + 0.5 \sin^2(\omega t)), \\ U_2 &= \bar{U} (0.5 + 0.5 \cos^2(\omega t)) \end{aligned}$$

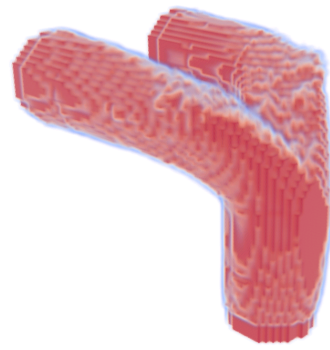
と定義する. ただし $\omega = 2\pi/\bar{T}$, $\bar{T} = 200\delta t$ とする. 出口は一定圧力とし, 圧力が 0.101325 MPa である圧力境界条件として与える. レイノルズ数は $Re = 1000$ とする. 格子点数は x, y, z 方向にそれぞれ 80 点とし, 合計 51 万 2 千点とする. 体積制約は $\bar{V} = 0.2|\Omega|$ とする.

最適化の仮定における形状の変化の様子を図 7.2 に示す. 最適解は図 7.2(f) に示す. 1900 ステップで収束条件を満足したので計算を停止した. ただし, 1 ステップあたり 200 ステップの LBM の計算を行っている.

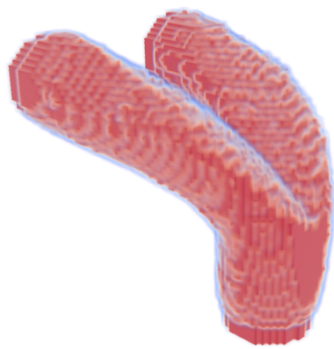
最適化の過程では, 二箇所の入口からの流路が同様に変化の様が見て取れる. ただし 1200 ステップ目 (図 7.2(e)) では奥側の流路だけに, 中央上側に小さな流体領域ができています. これは流路が変形していく過程で取り残された部分である. これが片方の流路にしかできていないことは, 流れ場が完全に左右対称ではないことを意味している. しかしながら最終的な最適解はおおよそ左右対称であり, 工学的に納得できる形状である.



(a) Step 0



(b) Step 300



(c) Step 600



(d) Step 900



(e) Step 1200



(f) Step 1900

図 7.2. 集合管問題の形状変化の様子.

表 7.1. 並列数と計算時間.

並列数	8	16	32	48	64	128
ノード数	8	16	16	16	16	16
計算時間 (秒)	9093.28	5311.9	2736.73	2005.68	1457.06	1035.02

7.5 並列効率

本節ではプログラムの並列計算の効率について述べる. 第 7.4 節で使用したプログラムを並列化し, 並列計算機を用いて計算を行った. 並列実装には Intel MPI [30] を用い, 計算機は XEON X5690 (3.33GHz with 8GB memory) を用いた. なお, 第 7.4 節で挙げた数値計算は全てシリアル計算の結果である. 最適化問題は縮流管問題とし, 格子点数と並列数を変えて計算時間を比較した. 各並列数における計算時間を表 7.1 に示す. 並列計算の効率を調べるため, 高速化率 S と実効並列化率 α の二つの指標を用いる [47].

格子点数 1,742,400 点, 総計算ステップ数 6,000 ステップの問題を解き, 実効並列化率は, 逐次実行時間に対する並列処理が可能な部分の実行時間の割合であり, プロセッサ数 m , n およびそれぞれのプロセッサ数に対する計算時間 $T(m)$, $T(n)$ を用いて

$$\alpha = \frac{T(m) - T(n)}{(1 - \frac{1}{n})T(m) - (1 - \frac{1}{m})T(n)}$$

と定義される. $m = 8$, $n = 128$ としたとき, 並列化率は 99.3% であった. すなわち, 逐次計算した場合の計算時間のうち, 99.3% が並列化されているということである. この並列化率は十分に大きな数値であり, 提案手法は並列化に適していると言える.

次に高速化率はプロセッサ数 p を用いて

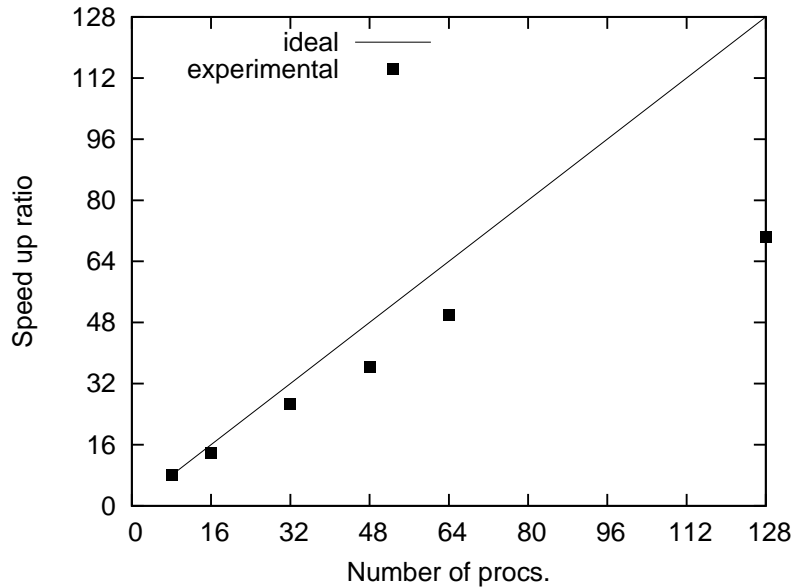
$$S = \frac{T(1)}{T(p)}$$

で定義される. しかしこの数値実験で扱った例題は問題規模が大きすぎるために逐次計算ができず, $T(1)$ が得られなかったため, 計算可能な最小の並列数 n を用いて

$$S_n = n \frac{T(n)}{T(p)}$$

とする. ここでは $n = 8$ とした. 高速化率を図 7.3 に示す. 64 並列まではほぼ理想値に近い高速化率を示している. 128 並列では理想値より小さくなるものの, 64 並列よりは増加する.

以上の結果から, 本提案手法は並列計算に適していると言える. これは LBM と最適化のアルゴリズムが並列計算に適していることに起因している.

図 7.3. 高速化率 S_8 .

7.6 iSA 法を用いた時間周期的な境界条件の下での流れ場のトポロジー最適化の結論

本章では、時間周期的な境界条件を持つ流れ場における流れ場のトポロジー最適化手法を構築した。流れ場は力学的にはカオス的であるとされ、また数値計算の上でも完全な周期性を持つ流れ場を再現することは困難である。そこで時間周期的な境界条件を想定して人為的な周期性を作り出し、その流れ場に対する最適化手法を構築した。最適化手法は iSA 法の枠組みを用い、随伴変数法を用いて感度を定式化した。この感度の定式化は既存手法と同様である。iSA 法での非定常流れの計算ステップ数を無限に長くした極限では、iSA 法と既存手法が等価になることを示し、iSA 法が既存手法の近似であることを示した。ただし、既存手法では一周期ごとに流れ場の計算をやり直すが、本手法は iSA 法を基礎にしているため、流れ場の計算と設計変数の更新は平行して行われる。一般に iSA 法の方が計算時間が短い。この事実より、非定常流れの計算ステップ数を制御することで計算時間と近似の精度をトレードオフにすることも可能である。最後に数値計算例により手法の妥当性を検討した。数値計算令では入口境界条件を時間周期的な境界条件で与えることで人為的な周期性を模擬した。このような時間周期的な境界条件は工学的に頻出し、重要な対象である。本手法によりそのような問題に対する最適化が可能である。

本手法ではメモリの使用量が多いという特徴もある。使用するメモリの量は周期の長さ依存し、一般的に複雑な流れ場は周期が長いことが多いため、より複雑な流れ場を解くためには必要なメモリが増大する。そのため、プログラムを並列化して使用できるメモリを増やすことが実用上必須である。本章ではプログラムを並列化した場合の性能を確認するために、同じ問

題に対して並列数を変えて計算速度を測定し、プログラムの並列化率と高速化率を計算した。その結果、プログラムの並列化率は十分に高く並列計算に適しており、また並列数が増加しても十分な高速化率が得られることがわかった。並列計算を用いることで、より大規模な流れ場や、より空間の離散化幅の小さな問題などの、大規模問題を解くことが可能になる。

第 8 章

結言

本論文では最適設計手法と、その流れ場のトポロジー最適化への適用を取り扱った。既存の流れ場のトポロジー最適化手法は場の変数を繰り返し解く必要があり、計算時間の増大の一因となっていた。これに対し本論文では、場の変数の計算と最適化計算を並行して行うことで計算時間を短縮する手法を構築した。第 3 章では一般の偏微分方程式に対して提案手法を構築し、時間発展方程式の形で定式化した。提案手法は解くべき最適解を近似的に解く手法であり、最適性は必ずしも保証されない。しかし一定の条件のもとでは最適性が保障される。提案手法は、近似をすることにより計算時間を短縮する手法と言える。また数値計算例として二次元弾性問題の剛性最大化問題を解き、手法の妥当性を示した。

次に第 4 章以降では流れ場のトポロジー最適化に提案手法を適用した。第 4 章では領域が多孔質媒体で充たされていると想定してその空隙率を変数として最適化を行う多孔質媒体モデルを用いて最適化を行った。流れ場は LBM で解き、感度も LBM を用いて定式化した。第 4 章以降で扱った問題は、提案手法が停止することが数学的には保証されない。そこで数値計算例を示すことで手法の妥当性を検証し、提案手法により妥当な最適解を得ることができた。一方で、最適解近傍における収束速度が遅いことも明らかになった。また、流体領域と固体領域の境界部分が不明確になる現象も確認された。

第 5 章では第 4 章で提案した手法の最適解近傍での収束速度の向上を目的として、目的関数の Hesse 行列を用いた最適化手法を提案した。流れ場の問題は非凸最適化問題であり、Hesse 行列は必ずしも正定値ではない。そこで Hesse 行列の固有値の正負により最急降下方向と Newton 方向を使い分ける方法を提案した。このとき、LBM の特徴を利用することで Hesse 行列が対角行列になる。提案手法と第 4 章の手法を用いて同じ問題を解き、収束速度を比較したところ、提案手法により収束速度が向上した。また、提案手法は局所最適解に収束しやすく、流体領域と固体領域の境界部分が明瞭になりやすい。

第 6 章では、流れ場の数値計算における境界条件の取り扱いを明確にして、流れ場のトポロジー最適化手法を提案した。境界条件は流れ場の数値計算において重要な項目である。まず level-set 関数を用いて流体領域と固体領域の境界を明確に表現し、LBM における非滑り境界条件として bounce-back 条件を用いて境界条件を適用した。また感度は bounce-back 条件を考慮して定式化した。空間微分の離散化として素朴な両側差分を用いた場合、妥当な解が得ら

れない。そこで bounce-back 条件における境界の位置を考慮した差分法を構築し、これを用いることで妥当な解を得た。

第7章では、時間周期的な境界条件のもとでの流れ場について、トポロジー最適化手法を提案した。流れ場のエネルギー損失の一周期分の積分を考えるために、感度の計算に随伴変数法を用いた定式化を行った。境界条件の取り扱いや差分法については第7章と同じ手法を用いて最適化手法を構築した。数値計算例により手法の妥当性を検証した。また、随伴変数法を用いる場合には計算に必要なメモリが多く、並列計算が必要であったため、並列プログラムを作成し、異なる並列数で同じ問題を計算して計算時間を計測し、プログラムの並列化率と高速化率を求めた。本手法を実装したプログラムの高速化率を調べたところ、十分な並列数まで望ましい高速化率が得られた。

第4章から第6章で用いた感度は、ある時刻において1ステップ未来を予測した感度である。このことから、提案手法を逐次感度近似法 (iSA 法) と呼ぶ。一方で第7章で提案した手法で用いた感度は、ある極限において既存のトポロジー最適化手法で用いる感度と一致する。また第3章から第6章で実施した数値計算では、いずれも実用上十分に短い時間で計算ができた。このことから、提案手法は既存手法の感度を近似する代わりに計算時間を短縮していると捉える事もできる。

以上のように、iSA 法を構築することで、当初の目的であった流れ場のトポロジー最適化の計算時間の短縮を実現した。また並列計算にも適した手法であり、より大規模な計算への適用も期待できる。今後、最適設計への要求は強まることが予想されており、より大規模な対象やより多くの格子点数が必要な対象を最適化することが求められる。iSA 法は、これらの要求に応えられる可能性があると言える。

本論文で iSA 法を適用した対象は流れ場のエネルギー損失最小化問題だけであった。今後はより幅広い問題に本手法を適用することが課題である。例えば流量を所望の値にするような問題や、振動する流れの周波数を特定の値から離れさせる問題などである。これらの目的関数を用いる場合、随伴変数法を用いることが必要である。

謝辞

本論文を執筆するに当たり、多くの方々からご指導とご助言、ご協力をいただいたことに對して、ここに深く感謝の意を表する。

指導教員の寒野善博准教授と駒木文保教授には、研究内容のみならず、研究への取り組み方についてもあたたかくご指導いただいた。特に大学に行く時間が十分に確保できない中で、終始、貴重で熱心なご指導をいただいたことは、感謝の念に堪えない。寒野善博准教授には修士号取得の際にもご指導いただき、またその後博士課程に進学するまでの間も折に触れて助言をいただいた。格別のご配慮をいただき、長い間にわたって常に貴重なご指導をいただいたことに深く感謝する。論文審査に際しては松尾宇泰教授、鈴木克幸教授、武田朗子准教授に多くの助言をいただいたことに厚く感謝する。また長尾大道准教授と廣瀬善大氏には研究室内での議論を通してさまざまな助言をいただいたことに御礼申し上げる。

株式会社 I H I には一定の時間を大学での研究活動に費やすことを許していただくとともに留学も援助していただき、また社内の方々には多くの助言や支援をいただいた。特に筆者に社会人博士に行くよう勧めてくださった西土隆幸氏と、筆者に留学を勧めてくださり多大なご支援をいただいた馬場秀成氏、社会人博士としての活動を応援し業務の調整に快く応じてくださった渡邊修氏に深く感謝を申し上げる。また市東素明氏、久保世志氏には流体の数値解析法および最適化について、海野大氏にはそれらに加えて並列計算手法とその実装について多くのご指摘と助言をいただいた。ここに厚く感謝を申し上げる。

University of Colorado Boulder の Kurt Maute 教授は、筆者の留学を快く引き受けてくださり、流れ場のトポロジー最適化と計算力学について実りある議論をしてくださった。ここに厚く御礼申し上げる。

京都大学西脇眞二教授ならびに山田崇恭氏には、学会や研究会を通して示唆に富むご意見をいただいた。ここに深く御礼申し上げる。また筆者と同様に流れ場の最適化を研究されている矢地謙太郎氏とはさまざまな機会に議論をさせていただき、多くの刺激をいただいた。ここに厚く感謝を申し上げる。

筆者が博士課程に在籍した間に、本論文の内容と直接関係のない業務にも多く携わった。なかでも実際の製品に関係する最適化技術や、京コンピュータの産業応用、積層造形技術の研究開発などは、本論文の内容とも関係が深く、これらの技術に関する経験や知見は本論文の内容に少なからず影響を与えている。このような豊かなテーマに恵まれた幸運に感謝するとともに、各技術に関する議論に応じていただき、叱咤激励のお言葉と貴重なご意見をいただいた多

130 謝辞

くの方々に感謝申し上げます。

最後に、常に支えてくださった家族と友人達に深く感謝する。

参考文献

- [1] N. Aage, T. H. Poulsen, A. Gersborg-Hansen, and O. Sigmund. Topology optimization of large scale Stokes flow problems. *Structural and Multidisciplinary Optimization*, 35:175–180, 2008.
- [2] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194:363–393, 2004.
- [3] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.
- [4] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, second edition, 2003.
- [5] A. A. S. Bhagat, E. T. K. Peterson, and I. Papautsky. A passive planar micromixer with obstructions for mixing at low reynolds numbers. *Journal of Micromechanics and Microengineering*, 17(5):1017, 2007.
- [6] P. Bhatnager, E. Gross, and M. Krook. A model for collision processes in gases, I: Small amplitude processes in charged and neutral one-component system. *Physical Review*, 94:511–525, 1954.
- [7] T. Borrval and J. Peterson. Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids*, 41:77–107, 2003.
- [8] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics. Springer-Verlag, New York, NY, USA, third edition, 2008.
- [9] M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *International Journal of Heat and Fluid Flow*, 21:186–196, 2000.
- [10] V. J. Challis and J. K. Guest. Level set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering*, 79:1284–1308, 2009.
- [11] T. J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, New York, USA, 2010.
- [12] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L. Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Trans-*

- actions of the Royale Society A*, 360:437–451, 2002.
- [13] D. d’Humières and P. Lallemand. Numerical simulations of hydrodynamics with lattice gas automata in two dimensions. *Complex Systems*, 1:599–632, 1987.
- [14] A. Evgrafov. The limits of porous materials in the topology optimization of Stokes flows. *Applied Mathematics and Optimization*, 52:263–277, 2005.
- [15] U. Frisch, D. D’Humières, D. Hasslacher, P. Lallemand, Y. Pomeau, and J. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707, 1987.
- [16] 藤岡奨, 牛島省. レイノルズ応力を考慮した格子ボルツマン法の構築. 応用力学論文集, 7: 1109–1116, 2004.
- [17] M. A. Gallivan, D. R. Noble, J. G. Georgiadis, and R. O. Buckius. An evaluation of the bounce-back boundary condition for lattice boltzmann simulations. *International Journal for Numerical Methods in Fluids*, 25(3):249–263, 1997.
- [18] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Computers and Fluids*, 35:888–897, 2006.
- [19] A. Gersborg-Hansen, M. P. Bendsøe, and O. Sigmund. Topology optimization of heat conduction problems using the finite volume method. *Structural and Multidisciplinary Optimization*, 31(4):251–259, 2006.
- [20] D. E. Goldberg. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Boston, MA, USA, 1989.
- [21] J. K. Guest and J. H. Précost. Topology optimization of creeping fluid flows using a Darcy–Stokes finite element. *International Journal for Numerical Methods in Engineering*, 66:461–484, 2006.
- [22] Y. Guo, C. Kato, Y. Yamade, Y. Ohta, T. Iwase, and R. Takayama. Numerical prediction of noise from the internal flow in a centrifugal fan. In *Proceedings of Fluids Engineering Conference*, paper no. 1104, Toyama, Japan, 2014. The Japan Society of Mechanical Engineers.
- [23] Z. Guo and T. S. Zhao. Lattice Boltzmann model for incompressible flows through porous media. *Physical Review E*, 66:036304, 2002.
- [24] R. T. Haftka. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, 1985.
- [25] F. J. Higeura, S. Succi, and R. Benzi. Lattice gas dynamics with enhanced collisions. *Europhysics Letters*, 9:345–349, 1989.
- [26] 平尾公彦, 横川三津夫. 京コンピュータと計算科学研究機構 (<特集>次世代スーパーコンピュータ「京」:動き出した大型プロジェクトの全体像). 日本物理學會誌, 66(7):524–528, 2011.
- [27] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge,

- MA, USA, 1992.
- [28] S. Hou, Q. Zou, S. Chen, G. Doolen, and A. Cogley. Simulation of cavity flow by the lattice Boltzmann method. *Journal of Computational Physics*, 118:329–347, 1995.
- [29] 茨木誠一, 冨田勲, 杉本浩一. 遺伝的アルゴリズムとニューラルネットワークを用いたターボチャージャ用遠心圧縮機の最適空力設計. *三菱重工技報*, 52(1):82–86, 2015.
- [30] Intel Corporation. *Intel MPI Library for Linux OS*. Intel Corporation, 2015.
- [31] 加藤恭義, 光成友孝, 築山洋. *セルオートマトン法*. 森北出版, 東京, 日本, 1998.
- [32] A. Kirk, S. Kreissl, G. Pingen, and K. Maute. Lattice Boltzmann topology optimization for transient flow. In *MAESC 2011 Conference*. Christian Brothers University, Memphis, Tennessee, May 2011.
- [33] H. Kobayashi. The subgrid-scale models based on coherent structures for rotating homogeneous turbulence and turbulent channel flow. *Physics of Fluids*, 17:045104, 2005.
- [34] 河野通方, 藤本元, 角田敏一, 氏家康成. *最新内燃機関*. 朝倉書店, 東京, 日本, 1995.
- [35] M. Krafczyk, J. Tölke, E. Rank, and M. Schulz. Two-dimensional simulation of fluid-structure interaction using lattice-Boltzmann methods. *Computers and Structures*, 79:2031–2037, 2001.
- [36] M. Krause, G. Thater, and V. Heuveline. Adjoint-based fluid flow control and optimisation with lattice Boltzmann methods. *Computers and Mathematics with Applications*, 65:945–960, 2013.
- [37] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, pages 311–326, 2012.
- [38] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalized shape optimization of fluids with the lattice Boltzmann method. *International Journal for Numerical Methods in Engineering*, 65:496–519, 2011.
- [39] 久保世志, 矢地健太郎, 山田崇恭, 泉井一浩, 西脇眞二. 流量制約を考慮した非圧縮粘性流れのレベルセット法に基づくトポロジー最適化. 第23回設計工学・システム部門講演会講演論文集, paper no. 1113, 日本機械学会, 東京, 日本, 2013.
- [40] 倉田直弥, 渡邊修, 米倉一男. 熱疲労強度向上を目的としたエキゾーストマニホールドの形状最適化. *最適化シンポジウム: OPTIS*, paper no. 1106, 日本機械学会, 東京, 日本, 2014.
- [41] P. Lallemand and L.-S. Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E*, 61:6546–6562, 2000.
- [42] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH '87 Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 163–169, 1987.

- [43] J. I. Madsen, W. Shyy, and R. T. Haftka. Response surface techniques for diffuser shape optimization. *AIAA Journal*, 38(9):1512–1518, 2000.
- [44] K. Mattila, J. Hyväluoma, J. Timonen, and T. Rossi. Comparison of implementations of the lattice-Boltzmann method. *Computers and Mathematics with Applications*, 55: 1514–1524, 2008.
- [45] G. R. McNamara and B. Alder. Lattice Boltzmann simulation of high Reynolds number fluid flow in two dimension. In M. Mareschal and B. Holian, editors, *Microscopic Simulations of Complex Hydrodynamic Phenomena*, pages 125–136. Plenum Publishing, 1992.
- [46] G. R. McNamara and G. Zanetti. Use of Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61:2332–2335, 1988.
- [47] 南一生. 京速コンピュータ「京」におけるアプリケーション高性能化. *電子情報通信学会誌*, 95(2):125–130, 2012.
- [48] K. Mogami, S. Nishiwaki, K. Izui, M. Yoshimura, and N. Kogiso. Reliability-based structural optimization of frame structures for multiple failure criteria using topology optimization techniques. *Structural and Multidisciplinary Optimization*, 32(4):299–311, 2006.
- [49] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, New York, USA, 2010.
- [50] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Boston, MA, USA, 8th edition, 2012.
- [51] 森野裕之, 竹中啓三, 畑中圭太, ニツ寺直樹, 中橋和博, 大林茂. 環境適応型高性能小型航空機設計における他分野統合最適化技術 (MDO) の適用. *三菱重工技報*, 42(5):216–219, 2005.
- [52] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, third edition, 2009.
- [53] F. Nannelli and S. Succi. The lattice Boltzmann equation on irregular lattices. *Journal of Statistical Physics*, 68:401–407, 1992.
- [54] 大山聖, 立川智章, 野々村拓, 藤井孝藏. 空力音響多目的設計探索—ロケット射点形状設計への適用—. *ターボ機械*, 42(5):305–309, 2014.
- [55] 岡本久. *ナビエ-ストークス方程式の数値*. 東京大学出版会, 東京, 日本, 2009.
- [56] E. Özkaya and N. Gauger. Single-step one-shot aerodynamic shape optimization. In K. Kunisch, J. Sprekels, G. Leugering, and F. Troltzsch, editors, *Optimal Control of Coupled Systems of Partial Differential Equations, Vol. 158 of International Series of Numerical Mathematics*, pages 191–204. Birkhauser Basel, 2009.
- [57] G. Pingen, A. Evgrafov, and K. Maute. Topology optimization of flow domains using the lattice Boltzmann method. *Structural and Multidisciplinary Optimization*, 34:

- 507–524, 2007.
- [58] G. Pingen, M. Waidmann, A. Evgrafov, and K. Maute. Adjoint parameter sensitivity analysis for the hydrodynamic lattice Boltzmann method with applications to design optimization. *Computers and Fluids*, 38:910–923, 2009.
- [59] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973.
- [60] K. Qu, C. Shu, and Y. T. Chew. Alternative method to construct equilibrium distribution functions in lattice-Boltzmann method simulation of inviscid compressible flows at high mach number. *Physical Review E*, 75:036706, Mar 2007.
- [61] D. E. Rumelhart, J. L. McClelland, and PDP Research Group. *Parallel Distributed Processing*. MIT Press, Boston, MA, USA, 1986.
- [62] T. Shimokawabe, T. Aoki, T. Takaki, T. Endo, A. Yamanaka, N. Maruyama, A. Nukada, and S. Matsuoka. Peta-scale phase-field simulation for dendritic solidification on the tsubame 2.0 supercomputer. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11*, pages 3:1–3:11, New York, NY, USA, 2011. ACM.
- [63] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.
- [64] R. Stainko and M. Burger. A one shot approach to topology optimization with local stress constraints. In M. Bendsøe, N. Olhoff, and O. Sigmund, editors, *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, volume 137, pages 181–184. Springer Netherlands, 2006.
- [65] J. D. Sterling and S. Chen. Stability analysis of lattice Boltzmann method. *Journal of Computational Physics*, 123:196–206, 1996.
- [66] S. Succi. *The Lattice Boltzmann Equation: for Fluid Dynamics and Beyond*. Oxford University Press, New York, USA, 2001.
- [67] C. Sun. Lattice-boltzmann models for high speed flows. *Physical Review E*, 58:7283–7287, 1998.
- [68] A. Takezawa, S. Nii, M. Kitamura, and N. Kogiso. Topology optimization for worst load conditions based on the eigenvalue analysis of an aggregated linear system. *Computer Methods in Applied Mechanics and Engineering*, 200(2528):226–2281, 2011.
- [69] 田村明久, 村松正和. 最適化法. 共立出版, 東京, 日本, 2004.
- [70] C. M. Teixeira. Incorporating turbulence models into the lattice-Boltzmann method. *International Journal of Modern Physics C*, 9:1159–1175, 1998.
- [71] The MathWorks Inc. *Using MATLAB*, 2009.
- [72] J. Tölke, S. Freudiger, and M. Krafczyk. An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Computers and Fluids*, 35:820–830, 2006.

- [73] 葛原道久, 高田尚樹, 片岡武. 格子気体法・格子ボルツマン法—新しい数値流体力学の手法—. コロナ社, 東京, 日本, 1999.
- [74] H. K. Versteeg and W. Malalasekera. *An Introduction to Fluid Dynamics: the Finite Volume Method*. Pearson Education, Essex, UK, second edition, 2007.
- [75] O. Watanabe, K. Yonekura, and S. Kubo. Study of shape design via principal component analysis and its optimization. *IHI Engineering Review*, 48(1):41–46, 2015.
- [76] M. Watari. Finite difference lattice boltzmann method with arbitrary specific heat ratio applicable to supersonic flow simulations. *Physica A: Statistical Mechanics and its Applications*, 382(2):502 – 522, 2007.
- [77] J. F. Wendt, editor. *Computational Fluid Dynamics, An Introduction*. Springer-Verlag, Heidelberg, 2009.
- [78] S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific Publication, Singapore, 1986.
- [79] K. Yaji, T. Yamada, M. Yoshino, T. Matsumoto, K. Izui, and S. Nishiwaki. Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions. *Journal of Computational Physics*, 274:158–181, 2014.
- [80] T. Yamada, K. Izui, S. Nishiwaki, and A. Takezawa. A topology optimization method based on the level set method incorporating a fictitious interface energy. *Computer Methods in Applied Mechanics and Engineering*, 199:2876–2891, 2010.
- [81] X. Yin and J. Zhang. An improved bounce-back scheme for complex boundary conditions in lattice boltzmann method. *Journal of Computational Physics*, 231(11):4295–4303, 2012.
- [82] 米倉一男. 構造物の設計及び解析に対する逐次錐計画法. 東京大学大学院情報理工学系研究科, 修士論文, 2010.
- [83] K. Yonekura and Y. Kanno. Global optimization of robust truss topology via mixed integer semidefinite programming. *Optimization and Engineering*, 11(3):355–379, 2010.
- [84] K. Yonekura and Y. Kanno. Second-order cone programming with warm start for elastoplastic analysis with von mises yield criterion. *Optimization and Engineering*, 13(2):181–218, 2012.
- [85] K. Yonekura and O. Watanabe. A shape parameterization method using principal component analysis in applications to parametric shape optimization. *Journal of Mechanical Design*, 136(12):121401, 2014.
- [86] G. H. Yoon. Topology optimization for stationary fluid-structure interaction problems using a new monolithic formulation. *International Journal for Numerical Methods in Engineering*, 82(5):591–616, 2010.
- [87] 袁新, 手島智博, 新関良樹. CFD を用いた蒸気タービン通路部最適化と実設計への適用. 東芝レビュー, 66(6):10–13, 2011.

- [88] T. Zhang, B. Shi, Z. Guo, Z. Chai, and J. Lu. General bounce-back scheme for concentration boundary condition in the lattice-Boltzmann method. *Physical Review E*, 85:016701, 2012.
- [89] D. P. Ziegler. Boundary conditions for lattice Boltzmann simulations. *Journal of Statistical Physics*, 71:1171–1177, 1993.
- [90] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Journal of Physics of Fluids*, 9:1591–1598, 1997.

付録 A

LBM の定式化に関する補遺

A.1 BGK モデルにおける壁面境界条件の定式化

本節では BGK モデルにおける非滑り条件を定式化する．本節の内容は葛原ら [73] をもとに、本論文の記号等に合わせて書き直したものである．ある点 $\boldsymbol{x} \in \Gamma_w$ が壁面上に存在すると仮定する． \boldsymbol{x} において流速が $\bar{\boldsymbol{U}}_w$ で与えられるとする．非滑り境界条件では $\bar{\boldsymbol{U}}_w = \mathbf{0}$ である．移流操作と衝突操作をそれぞれ次のように置き換える．まず移流操作は bounce-back 条件と同様、

$$\begin{aligned} f_i(\boldsymbol{x} + \boldsymbol{e}_i, t_{k+1}) &= f_i^{\text{col}}(\boldsymbol{x}, t_k) \quad \text{if } \boldsymbol{x} + \boldsymbol{e}_i \notin \Omega_s, \\ f_{-i}(\boldsymbol{x}, t_{k+1}) &= f_i^{\text{col}}(\boldsymbol{x}, t_k) \quad \text{if } \boldsymbol{x} + \boldsymbol{e}_i \in \Omega_s \end{aligned}$$

となる．次に衝突操作は

$$f_i^{\text{col}}(\boldsymbol{x}, t_k) = f_i^{\text{eq}}(\boldsymbol{x}, t_k; \bar{\boldsymbol{U}}_w)$$

とする．ここで、 $f_i^{\text{eq}}(\boldsymbol{x}, t_k; \boldsymbol{u}(\boldsymbol{x}, t_k))$ は (2.10) において $\boldsymbol{u} = \bar{\boldsymbol{U}}_w$ としたものである．

A.2 BGK モデルにおける速度境界条件の定式化

本節では BGK モデルにおける速度境界条件を定式化する．本節の内容は葛原ら [73] をもとに、本論文の記号等に合わせて書き直したものである．点 \boldsymbol{x} に $\boldsymbol{u} = \bar{\boldsymbol{U}}_v$ の速度境界条件を適用する． Ω の内部 $\text{int}\Omega$ に向かう方向の集合を $\mathcal{I}(\boldsymbol{x}) = \{i \mid \boldsymbol{x} + \boldsymbol{e}_i \in \text{int}\Omega\}$ とする．まず移流操作は、(2.4b) と同じである．その後 $\text{int}(\Omega)$ に含まれる点に対して衝突操作を行い、密度 ρ を計算する．続いて \boldsymbol{x} における密度 $\rho(\boldsymbol{x})$ を、

$$\rho(\boldsymbol{x}) = \frac{1}{|\mathcal{I}(\boldsymbol{x})|} \sum_{i \in \mathcal{I}(\boldsymbol{x})} \rho(\boldsymbol{x} + \boldsymbol{e}_i)$$

で計算する．ただし $|\mathcal{I}(\boldsymbol{x})|$ は $\mathcal{I}(\boldsymbol{x})$ の要素の個数を表す．最後に \boldsymbol{x} において $\boldsymbol{u} = \bar{\boldsymbol{U}}_v$ と $\rho(\boldsymbol{x})$ を用いて平衡分布関数 (2.10) を計算し、

$$f_i^{\text{col}}(\boldsymbol{x}, t_k) = f_i^{\text{eq}}(\boldsymbol{x}, t_k; \bar{\boldsymbol{U}}_v, \rho(\boldsymbol{x}))$$

とする。

A.3 D3Q19 モデル MRT-LBM における速度境界条件の定式化

本節では MRT-LBM における速度境界条件を, Zou and He [90] にしたがって定式化する. 点 \mathbf{x} に $\mathbf{u} = \bar{U}_v$ の速度境界条件を適用する. Ω の内部 $\text{int}\Omega$ に向かう方向の集合を $\mathcal{I}(\mathbf{x}) = \{i \mid \mathbf{x} + \mathbf{e}_i \in \text{int}\Omega\}$ とし, 境界上の点に向かう方向の集合を $\mathcal{J}(\mathbf{x}) = \{i \mid \mathbf{x} + \mathbf{e}_i \in \Gamma\}$ とし, 領域の外に向かう方向の集合を $\mathcal{K}(\mathbf{x}) = \{i \mid \mathbf{x} + \mathbf{e}_i \notin \Omega\}$ とする. ここで \mathcal{I} , \mathcal{J} , \mathcal{K} は互いに素な集合である. いま \mathbf{x} における粒子分布関数と密度と速度は次の関係を満たす必要がある:

$$\rho = \sum_{i \in \mathcal{I} \cup \mathcal{J}} f_i + \sum_{j \in \mathcal{K}} f_j, \quad (\text{A.1a})$$

$$\rho \mathbf{u} = \sum_{i \in \mathcal{I} \cup \mathcal{J}} f_i \mathbf{e}_i + \sum_{j \in \mathcal{K}} f_j \mathbf{e}_j. \quad (\text{A.1b})$$

ここで $f_{i \in \mathcal{I} \cup \mathcal{J}}$ は既知だが, $f_{j \in \mathcal{K}}$ は未知である. そこで (A.1) を満たすように $f_{j \in \mathcal{K}}$ を決定する. 境界条件が適用される可能性のある六面のうち, 代表して図 A.1 の A 面における速度境界条件を考える. 各方向は

$$\begin{aligned} \mathcal{I} &= \{1, 7, 8, 11, 12\}, \\ \mathcal{J} &= \{0, 3, 4, 5, 6, 15, 16, 17, 18\}, \\ \mathcal{K} &= \{2, 9, 10, 13, 14\} \end{aligned}$$

と分類される. (2.2) より

$$\begin{aligned} (\mathbf{e}_i)_x &= 1, \quad i \in \mathcal{I}, \\ (\mathbf{e}_i)_x &= 0, \quad i \in \mathcal{J}, \\ (\mathbf{e}_i)_x &= -1, \quad i \in \mathcal{K} \end{aligned}$$

である. これを用いると (A.1) は

$$\rho = \sum_{i \in \mathcal{I} \cup \mathcal{J}} f_i + \sum_{j \in \mathcal{K}} f_j, \quad (\text{A.2a})$$

$$\rho u_x = \sum_{i \in \mathcal{I}} f_i - \sum_{j \in \mathcal{K}} f_j \quad (\text{A.2b})$$

と書き直される. (A.2) より $f_{j \in \mathcal{K}}$ を消去すると

$$\rho = \frac{1}{1 - u_x} \left(\sum_{i \in \mathcal{I}} f_i + 2 \sum_{i \in \mathcal{J}} f_i \right)$$

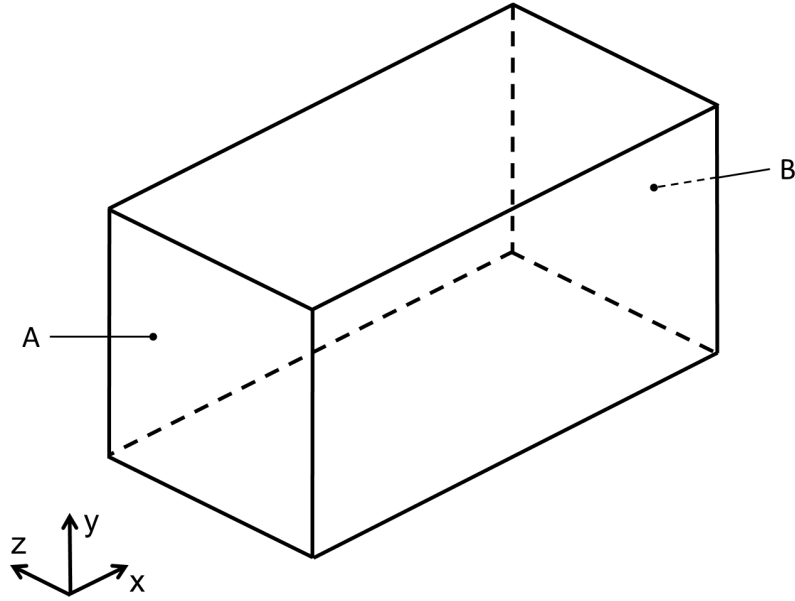


図 A.1. 境界条件を適用する面.

を得る. ρ を用いて $f_{j \in \mathcal{K}}$ を

$$\begin{aligned} f_2 &= f_1 - \frac{1}{3}\rho u_x, \\ f_9 &= f_8 - \frac{1}{2}(f_3 - f_4 + f_{15} - f_{16} + f_{17} - f_{18}) - \frac{1}{6}\rho u_x + \frac{1}{2}\rho u_y, \\ f_{10} &= f_7 + \frac{1}{2}(f_3 - f_4 + f_{15} - f_{16} + f_{17} - f_{18}) - \frac{1}{6}\rho u_x - \frac{1}{2}\rho u_y, \\ f_{13} &= f_{12} - \frac{1}{2}(f_5 - f_6 + f_{15} - f_{17} + f_{16} - f_{18}) - \frac{1}{6}\rho u_x + \frac{1}{2}\rho u_z, \\ f_{14} &= f_{11} + \frac{1}{2}(f_5 - f_6 + f_{15} - f_{17} + f_{16} - f_{18}) - \frac{1}{6}\rho u_x - \frac{1}{2}\rho u_z \end{aligned}$$

と定める. これは (A.1) を満たす.

A.4 D3Q19 モデル MRT-LBM における圧力境界条件の定式化

本節では MRT-LBM における圧力境界条件を, Zou and He [90] にしたがって定式化する. 記号の定義は第 A.3 節と同じとする. 本節では境界条件が適用される可能性のある六面のうち, 代表して図 A.1 の B 面における圧力境界条件を考える. 各方向は

$$\begin{aligned} \mathcal{I} &= \{2, 9, 10, 13, 14\}, \\ \mathcal{J} &= \{0, 3, 4, 5, 6, 15, 16, 17, 18\}, \\ \mathcal{K} &= \{1, 7, 8, 11, 12\} \end{aligned}$$

と分類される。圧力境界条件においても (A.1), (A.2) が満たされる必要がある。 (A.2) より $f_{j \in \mathcal{K}}$ を消去すると

$$u_x = 1 - \frac{1}{\rho} \left(\sum_{i \in \mathcal{I}} f_i + 2 \sum_{i \in \mathcal{J}} f_i \right)$$

を得る。 u_x を用いて $f_{j \in \mathcal{K}}$ を

$$\begin{aligned} f_1 &= f_2 + \frac{1}{3} \rho u_x, \\ f_8 &= f_9 + \frac{1}{2} (f_3 - f_4 + f_{15} - f_{16} + f_{17} - f_{18}) + \frac{1}{6} \rho u_x, \\ f_7 &= f_{10} - \frac{1}{2} (f_3 - f_4 + f_{15} - f_{16} + f_{17} - f_{18}) + \frac{1}{6} \rho u_x, \\ f_{12} &= f_{13} + \frac{1}{2} (f_5 - f_6 + f_{15} - f_{17} + f_{16} - f_{18}) + \frac{1}{6} \rho u_x, \\ f_{11} &= f_{14} - \frac{1}{2} (f_5 - f_6 + f_{15} - f_{17} + f_{16} - f_{18}) + \frac{1}{6} \rho u_x \end{aligned}$$

と定める。これは (A.1) を満たす。

A.5 行列 M の定義

D3Q19 モデルにおける MRT-LBM の行列 M は Humieres *et al.* [12] によると以下のよう
に定義される.

$$M = \begin{pmatrix}
 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
 -30.0 & -11.0 & -11.0 & -11.0 & -11.0 & -11.0 & -11.0 & 8.0 & 8.0 & 8.0 \\
 12.0 & -4.0 & -4.0 & -4.0 & -4.0 & -4.0 & -4.0 & 1.0 & 1.0 & 1.0 \\
 0.0 & 1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 \\
 0.0 & -4.0 & 4.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 0.0 & 0.0 & 1.0 & 1.0 & -1.0 \\
 0.0 & 0.0 & 0.0 & -4.0 & 4.0 & 0.0 & 0.0 & 1.0 & 1.0 & -1.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -4.0 & 4.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 2.0 & 2.0 & -1.0 & -1.0 & -1.0 & -1.0 & 1.0 & 1.0 & 1.0 \\
 0.0 & -4.0 & -4.0 & 2.0 & 2.0 & 2.0 & 2.0 & 1.0 & 1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & 1.0 & 1.0 & -1.0 & -1.0 & 1.0 & 1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & -2.0 & -2.0 & 2.0 & 2.0 & 1.0 & 1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & -1.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -1.0 & -1.0 & 1.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 8.0 & 8.0 \\
 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\
 -1.0 & 1.0 & -1.0 & 1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 -1.0 & 1.0 & -1.0 & 1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 -1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 & -1.0 \\
 -1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 & -1.0 \\
 0.0 & 1.0 & 1.0 & -1.0 & -1.0 & 1.0 & 1.0 & -1.0 & -1.0 \\
 0.0 & 1.0 & 1.0 & -1.0 & -1.0 & 1.0 & 1.0 & -1.0 & -1.0 \\
 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & -2.0 & -2.0 & -2.0 & -2.0 \\
 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & -2.0 & -2.0 & -2.0 & -2.0 \\
 1.0 & -1.0 & -1.0 & -1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 1.0 & -1.0 & -1.0 & -1.0 & -1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & -1.0 & 1.0 \\
 0.0 & 1.0 & -1.0 & -1.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 -1.0 & -1.0 & 1.0 & -1.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & -1.0 & 1.0 & -1.0 \\
 0.0 & 1.0 & 1.0 & -1.0 & -1.0 & -1.0 & -1.0 & 1.0 & 1.0
 \end{pmatrix} \tag{A.3}$$

付録 B

既存の文献における KKT 条件の定式化

既存の文献 ([80] など) での随伴法変数法の定式化と、本論文の定式化が等価であることを説明する。既存の定式化では、(3.2) で定義された最適化問題 (P) の独立変数は ϕ であり、 u は従属変数として取り扱う。改めて最適化問題を定式化すると、次のようになる。

$$\begin{aligned}
 (\text{P}') \quad & \text{find} \quad \phi \\
 & \text{min.} \quad E(u(\phi), \phi), \\
 & \text{s.t.} \quad \mathcal{A}(u(\phi), \phi) = 0, \\
 & \quad \quad \mathcal{G}(\phi) \leq 0.
 \end{aligned}$$

Lagrange 関数は

$$\mathcal{L}'(\phi) = E(u(\phi), \phi) - \int_{\Omega} \mathcal{A}(u(\phi), \phi) v dx - \int_{\Omega} \mathcal{G}(\phi) \lambda dx$$

となる。すると KKT 条件は、

$$\begin{aligned}
 \frac{\partial \mathcal{L}'}{\partial \phi} &= \frac{\partial E}{\partial \phi} + \frac{\partial E}{\partial u} \frac{\partial u}{\partial \phi} - \left(\frac{\partial \mathcal{A}}{\partial \phi} + \frac{\partial \mathcal{A}}{\partial u} \frac{\partial u}{\partial \phi} \right) v - \frac{\partial \mathcal{G}}{\partial \phi} \lambda = 0, \\
 \mathcal{A} &= 0, v \in \mathbb{R}, \\
 \mathcal{G} \lambda &= 0, \mathcal{G} \leq 0, \lambda \leq 0
 \end{aligned} \tag{B.1}$$

と書ける。(B.1) を $\frac{\partial u}{\partial \phi}$ で整理すると、

$$\begin{aligned}
 \frac{\partial \mathcal{L}'}{\partial \phi} &= \frac{\partial E}{\partial \phi} - \frac{\partial \mathcal{A}}{\partial \phi} v - \frac{\partial \mathcal{G}}{\partial \phi} \lambda + \left(\frac{\partial E}{\partial u} - \frac{\partial \mathcal{A}}{\partial u} v \right) \frac{\partial u}{\partial \phi} \\
 &= \frac{\partial \mathcal{L}}{\partial \phi} + \frac{\partial \mathcal{L}}{\partial u} \frac{\partial u}{\partial \phi}
 \end{aligned} \tag{B.2}$$

と書ける。ただし $\frac{\partial \mathcal{L}}{\partial \phi}$ および $\frac{\partial \mathcal{L}}{\partial u}$ は (3.4a) および (3.4b) を使用した。(B.2) より、 $\frac{\partial \mathcal{L}'}{\partial \phi} = 0$ の十分条件は $\frac{\partial \mathcal{L}}{\partial \phi} = 0$ かつ $\frac{\partial \mathcal{L}}{\partial u} \frac{\partial u}{\partial \phi} = 0$ である。随伴変数法ではこの十分条件を満たす解を探索す

る。従って,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \phi} &= 0, \\ \frac{\partial \mathcal{L}}{\partial u} &= 0, \\ \mathcal{A} &= 0, v \in \mathbb{R}, \\ \mathcal{G}\lambda &= 0, \mathcal{G} \leq 0, \lambda \leq 0\end{aligned}$$

を解くことになり, これは (3.4) と一致する.