機体設計と飛行経路の統合的最適化に関する研究

土屋武司

学位論文

機体設計と飛行経路の統合的最適化に関する研究

1999年12月

東京大学大学院工学系研究科

航空宇宙工学専攻

土屋武司

目次

図目次

表目次

第1章
序論

1.1 はじめに

あるシステムを最適化するとは，システム設計者の願望を表した目的関数と呼ばれる関数を最小化あるいは最大化するように，システムの状態を表す変数を等式または不等式で表される状態方程式のもとで見つけることである．このような問題を最適化問題と呼ぶ．理論的にはいかなるシステムも各関数を定義し最適化問題として記述さえできれば必ず解を得ることができる．しかしながら，ごく単純な最適化問題を除いてその求解には計算機の力を借りなければならず，結果として最適化問題が取り扱える範囲は計算機の発達と共に広がってきたといえる．一般的に，最適化計算は目的関数，状態方程式の数値計算，すなわちシステムの解析を数多く繰り返して解を得る．よって，解析が可能なシステム全てが最適化可能であるとは限らない．解析手法で扱える自由度のオーダに比べて，その解析手法を組み入れた最適化手法で扱える自由度のオーダは少ない．また，最適化問題の真の最適解を誰も知ることはできない．多くの時間を割けばより良い解を得ることができる．しかし，苦労して得られた解が果たして真の最適解であるのか決して分からない．これらの問題点を解消するために，計算機というハード面の発達に加えて新しい最適化手法を生み出しまた改良を加えるといったソフト面の発達も著しい．

今日，そうした最適化手法に注目が集まり発達の原動力となっているのが，実在するシステムに対する最適化手法適用への期待の高まりである．システム設計に対する要求は，高度化，高機能化してきている．例えば，ある製品を設計するとき，従来の経験に頼った手法では対処できない場合が発生している．そこで，計算機による支援と最適化技術を積極的に取り入れて設計を行う流れが強まっている．その考えを推し進めると，製品を

構成する個々の要素や製品を製造する各段階を個別に解析して最適化するのではなく，システム全体もしくは複数の領域（専門分野）（discipline）にまたがる解析と最適化を同時に考えることによってさらに高い効率を得ることができる．複数の領域にまたがる最適化を複合領域最適化（multidisciplinary optimization: MDO）[1]と呼ぶ．

航空機の設計においても，個別の領域を超えた設計を行うことの重要性は認識され，航空機メーカはシステム解析に計算機を積極的に活用し始めている．それでは複合領域最適化はどうであろうか．従来の航空機設計に対する複合領域最適化を考える一例として，機体設計最適化と飛行経路最適化の関係を考えてみる．まず，設計者には新たな航空機に対する大まかな飛行条件を含む設計仕様が指定され，それに沿った設計が行われる．その際，機体の個々の構成要素，または全要素に対して最適化が実行されることがあろう．しかし，このとき飛行経路も同時に最適化されることはない．なぜなら，従来の航空機は汎用性が求められ，詳細な運用法と飛行経路を航空機設計の段階では決められないからである．また，機体設計と飛行経路の解析と最適化技術はそれぞれ異なる技術領域にあるためでもある．よって，従来，機体設計がまず成され，設計された機体に対して飛行経路の最適化が行われることはあったが，両者の統合的最適化は行われていない．

一方，現在，世界中で研究が進められている次世代の宇宙往還機，およびその実験機等では，従来の航空機と異なり，要求される設計仕様，ミッションが厳しく，極限の性能が追求されなければならない．しかし，こうした航空機は，厳格に決められた限られたミッションの中でしか運用されない．よって，機体設計とミッションを達成する飛行経路を同時に決定することが可能である．また，こうした航空機には新しい材料，新しいエンジン，新しい制御技術等，これまでの経験に頼ることができない革新的技術が取り入れられなければならない．したがって，機体設計のみに適用範囲がとどまっていた複合領域最適化法に，飛行経路の最適化を加えることは可能であり，また必要不可欠なことでもある．特に本論文では機体設計と飛行経路の統合的最適化問題の対象としてスペースプレーン[2]を考える．スペースプレーンは完全再使用型の単段式宇宙往還機として研究されている．現在の技術水準では実現不可能であり，飛行制御技術，推進技術，材料構造技術，空力技術などの技術領域個別の進展と，それらを融合して最適化する技術が求められている．

以上を踏まえ，本論文では複合領域最適化問題の一種である機体設計と飛行経路の統合的最適化問題を中心とした大規模最適化問題に対する数値解法の提案を行う．そしてそれをスペースプレーンの統合的最適化問題に適用し，実現性に向けた考察を行う．

1.2 複合領域最適化の研究

複合領域にわたる解析法，最適化法には考慮すべき様々な要素が含まれている．例えば，各領域におけるモデリング法と解析法，また最適化過程において繰り返される解析の計算コストを軽減するための近似化技術，システムの感度解析法，各領域ごとに最適化問題を分割するための最適化問題の定義法と解法，また人間の判断を必要とする場合や汎用プログラムとして完成するためにはインターフェイスについても考慮しなければならない．本論文ではこれらの問題点のうち最適化問題の定義法と解法を中心に考察を加える．

一般的に，複合領域最適化の対象となる大きなシステムではシステム全体を一挙に扱って総合的な解析や設計を行うことは困難な場合が多い．そこで，大きな負荷を要するこれら一連の作業を軽減するために，システムを複数のサブシステムに分割して，解析，最適化を進める分割解法が考えられる．例えば航空機では，構造，熱，空力，制御などの専門領域ごとのサブシステムが考えられる．また，翼，胴体，エンジンなどの部分の構造も考えられる．さらにこれらの混合システムもあるであろう．サブシステムをより細分化して階層構造にすることも考えられる．このようにすることで，解析，最適化の負荷を軽減するだけでなく，各サブシステムに関する専門家達は他のサブシステムに対する配慮を最小限にとどめて，自分達が持つ解析ツール，最適化手法を最大限に活用することができる．

分割して得られたサブシステムは分散的にあるいは並列的に解析，感度解析，最適化の計算が実行されなければならない．そして，個別に最適化が施されたサブシステムを最終的には統合し，1 つの最適システムとして構成する必要がある．ここで，サブシステムはお互いに独立して存在しているのではなく，お互いの情報の交換によって統合的なシステムを形成していることに注意しなければならない．このサブシステム間の接続のことをカップリング（coupling）という．個別のサブシステムを統合するには，サブシステム間のカップリングに対する配慮が必要である．この統合方法についてもいろいろ研究されている．基本となる方法は複数サブシステムにまたがる変数に関する各サブシステムの感度を計算し，全システムの最適化計算を行う方法である．このシステムレベルの最適化問題は元の最適化問題が簡素化された問題と見ることができ，また coordination 問題と呼ばれることがある．サブシステム解析を含むシステム解析，サブシステムとシステムの感度解析，サブシステムレベルの最適化計算，システムレベルの最適化計算は複合領域最適化法を構成する重要な要素である．

最適化問題の分割解法に関する多くの研究は，大規模線形計画法に対する Dantzig-Wolfe

の分割原理[3]（Dantzig-Wolfe decomposition principle）に関する研究に代表されるオペレーションリサーチ（operation research: OR）の領域から始まった．一方で，実在する工学問題に対して分割解法を適用しようと各分野の研究者が理論と応用を試みてきた．そのほとんどは構造設計への適用を試みようとする研究であった．その理由として，元々，工学問題に対する最適化理論の適用は構造設計の専門家達が中心となって進められてきたことと，有限要素法などの構造解析法で扱うことができる変数の数に比べて最適化法で扱うことができる変数の数が少ないため，大規模な最適化問題を分割解法により解く必要があったためである．一般的な大規模非線形計画問題に対する分割法と最適化法は，Kirsch がテキスト（文献4）の中でまとめたモデル調整（model coordination）法とゴール調整（goal coordination）法と呼ばれる 2 つの方法が広く知られている．前述した通り coordination とはシステムレベルでの最適化を意味するが，初期の複合領域最適化法において coordination は必ずしも存在しないことが多かった[5]．しかし，Kirsch の解法が知られて以降，coordination は重要な位置を占めるようになった．最近でもモデル調整法にゴール調整法的な考えを用いた方法[6]が試みられている．

一方，前節で述べた航空宇宙業界を取り巻く状況から，現在，米国には NASA を中心に High Performance Computing and Communications Program（HPCCP）というプロジェクトが存在する．その目的は高性能コンピュータを航空宇宙分野に応用するために，コンピュータツール，可視化技術，データベース，ユーザインターフェイス等の研究を促進させることにある．そのプロジェクト内に Computational Aerosciences（CAS）プロジェクトが存在し，ラングレー研究所において次世代航空機の複合領域設計と最適化を行うソフトウェアの開発を行っているグループがある．80 年代後半から 90 年代にかけて，より実践的な複合領域最適化法がこのグループの関係者から提案されている．All-at-Once（All-in-One）法[7]，Concurrent Subspace Optimization（CSSO）法[8]，Collaborative Optimization（CO）法[9, 10, 11]と呼ばれる手法である．

また，これらの成果を受け，具体的なソフトウェアも市販されつつある．例えば Engineous Software 社から「iSIGHT」[12]と呼ばれる複合領域最適化支援ソフトが販売されている．

これまでのところ，モデル調整法とゴール調整法も含めたこれら複合領域最適化法は，元の大規模最適化問題を直接解いて解を得るのと比較して，収束性が悪く，多くのシステム解析と最適化計算の繰り返しが必要であると報告されている[13]．また，数値解法として安定しているか，また計算途中でサブシステムの最適化問題が解けなくなる現象に陥る可

能性があるかどうかを意味する最適化手法におけるロバスト性が悪いことも知られている．特に分割されたサブシステムどうしのカップリングが強いと顕著である．なぜならば，カップリングが強いと元の最適化問題が簡素化されていると考えられる coordination 問題が無意味な解を出すことがあるためである．収束性とロバスト性が悪いために，最適化計算の収束判定において妥協が必要となり，得られた解が厳密な最適解であることも期待できない．代表的なこれらの解法と特徴は補遺Bに詳しくまとめておく．

以上のような複合領域最適化法を用いて，様々な領域間で応用例が報告されている．なかでも複合領域の設計における最適化の発展の原動力になっているといえる代表的な領域の組み合わせは，構造問題を中心とした「構造問題と制御問題」，「構造問題と熱・流体問題」の２つである．構造問題と制御問題の統合的最適化において，構造系では形状と位相最適化，制御系ではアクチュエータ，センサの最適配置とフィードバック制御におけるフィードバックゲインの最適化問題が考えられる[13]．また，熱・流体問題と構造問題の統合的最適化問題は，航空宇宙分野において，航空機の翼または全機形状の最適化という問題設定が多数見受けられる[15]．現在までのところ，SST，HST のような航空機と宇宙往還機の概念設計に対する適用例は多数見受けられる[7, 16, 17]が，実用レベルにある開発ツールがまだ存在しないため，実在する航空機の設計に使用したという報告はまだない．

1.3 並列最適化の研究

分割して得られたサブシステムでは解析，最適化の計算が並列的に行われることはすでに述べた．ところで，複合領域の解析，最適化とは別に，並列計算に関してはソフト面，ハード面から多くの研究が進められており[18, 19]，最適化計算に対する並列計算の研究も例外ではない[20]．その理由の１つとして，一般に，最適化計算では変数の数，制約条件の数，非線形性が増加するにつれ，計算コストが比例を超える量で増加することが挙げられる．図1.1，1.2は第 3 章で示す「不連続な潮流の中を進む船」に関する最適化問題について，変数の数が増加するにつれて，最適解を得るまでの計算時間と計算に必要なメモリ量がどう変化するかを示した図である．これらの図から，計算時間は変数の数の 3〜4 乗に比例し，メモリ量は変数の数の２乗に比例して増加するのが分かる．よって，多くの変数と制約条件を持つ大規模最適化問題を並列計算によって解くことが考えられている．

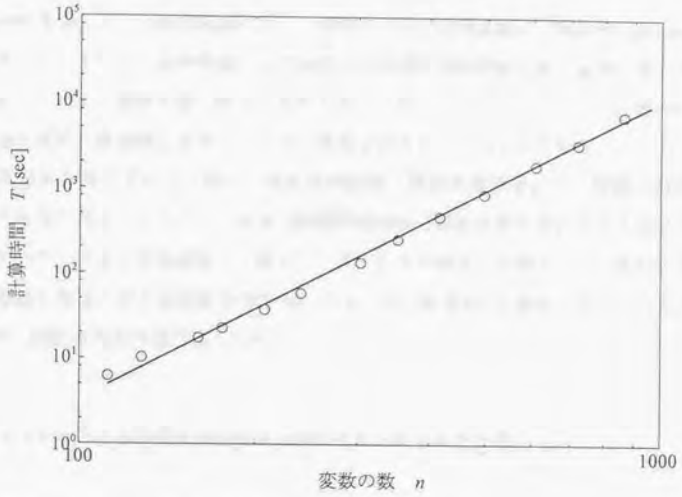


図1.1 変数の数と計算時間の関係

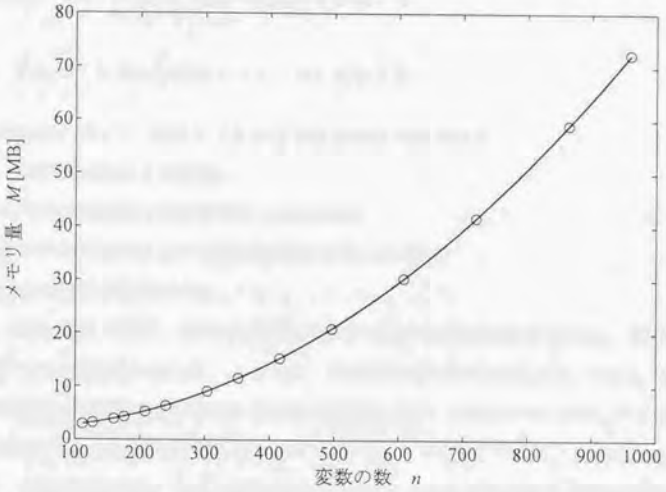


図1.2 変数の数とメモリ量の関係

複合領域最適化法とは並列計算によって最適化を行う並列最適化（parallel optimization）法の一種であるといえ，並列最適化法の研究成果を複合領域最適化法に活用することは可能である．しかし，通常の並列計算と異なる点は，並列化された1つ1つの計算内容が個別の領域の解析，最適化に相当し，工学的意味を有するということである．

並列最適化に限らず並列計算の研究の方向性は「解法の並列化」と「問題の並列化」に大きく分けられる．このうち，最適化問題に対する「解法の並列化」からの研究は，主に並列アルゴリズムの理論研究の一環として進められている．一例として，制約条件を持たない最適化問題に対する逐次 2 次計画（SQP）法（補遺A）の並列アルゴリズムを考えてみる[20]．SQP 法は次の繰り返しからなっている．

Step 1.

近似ヘッセ行列と目的関数の勾配から解の更新ベクトルを計算．

Step 2.

そのベクトル方向で1次元探索を行い，最小値を見つける．

Step 3.

目的関数の勾配を計算し，収束判定条件を評価する．

Step 4.

勾配，方向ベクトルから近似ヘッセ行列を更新する．

以上の解法において，次のような部分で並列化が可能である．

(1) Step 1.での線形解析の並列化．

(2) Step 2.での目的関数の複数評価による並列化．

(3) Step 3.での勾配計算における偏導関数の評価の並列化．

(4) Step 4.での算術計算の並列化．

また，SQP 法に限らず，局所最適解を持つ多峰性の目的関数の場合には，初期値に依存して得られる最適解が異なる．この場合，大域的最適解を得る方法の1つとして，解が存在すると思われる解空間の内部に複数の初期解を選び，複数プロセッサによって多点を並列的に探索することが考えられている．

一方，「問題の並列化」に基づく並列最適化は 1 つの最適化問題を複数の最適化問題に分割して複数のプロセッサによって解く最適化法であり，前節で説明した複合領域最適化法はここに含まれる．複合領域最適化法とは別に OR の研究の一環としても研究されて

いる[21]．本論文ではこれら「問題の並列化」による並列最適化法を特に分割解法と呼ぶ．

「問題の並列化」と「解法の並列化」の2つを比較してみると，一般に「解法の並列化」で行われる1つ1つの並列計算に工学的意味を持たせることは不可能であり，複合領域最適化への適用はできない．しかし，並列計算の効率（第3章）の点では「問題の並列化」より「解法の並列化」のほうが優れる．つまり，1つの問題を複数の問題に細分化して並列計算を行うよりは，1つの解法を構成する細かい計算を並列的に処理したほうがより効率的である．また，「問題の並列化」に注目した並列最適化法では，並列に行われる計算に工学的意味を持たせることが可能であるが，前節で述べた複合領域最適化法に見られる収束性とロバスト性が悪いという問題点を抱えている．

1.4 本研究の目的

前節までの既存の研究を踏まえ，機体設計と飛行経路の統合的最適化問題に適用可能であり，かつ並列化効率，収束性，ロバスト性に優れた並列最適化法を提案することを本論文の目的の1つとする．本論文では「解法の並列化」にまず注目し，既存の最適化法の並列化を試みる．後に説明されるが，得られた解法は見方を変えると「問題の並列化」による並列最適化法の1つと考えることが可能である．よって，1つ1つの並列に行われる計算に工学的意味をもたせることが可能であるため，統合的最適化問題への適用も可能であり，既存の最適化法に比べて，優れた並列化効率，収束性とロバスト性を持つ．

高い精度をもった並列最適化法が必要な問題として飛行経路最適化問題が挙げられる．飛行経路最適化問題は最適制御問題として定式化されるが，制御時間が長く複雑な最適制御問題に対して，精度の良い解を得ようとすると，大規模最適化問題を解かなくてはならない．そこで，最適制御問題に並列最適化法を適用することが考えられるが，前々節で述べた分割解法を適用すると，収束性の悪さから精度の良い解を得ることができない．そこで本論文で提案される並列最適化法を適用する．

次に飛行経路を最適化するだけでなく，機体設計の最適化問題も含めた機体設計と飛行経路の統合的最適化問題に並列最適化法の適用を行う．本論文の問題設定の特徴の1つとして，従来のように構造系を中心にするのではなく，航空機の誘導・制御問題を中心にして，そこから構造問題，空力問題へと最適化の領域を広げていくことが挙げられる．まず最適化手法検証のために平易な紙飛行機のサイズと投げ方を統合的に最適化する問題を考え，その後にスペースプレーンの機体設計と飛行経路の統合的最適化問題を考慮する．

特に，スペースプレーンの統合的最適化問題については，スペースプレーンを現在の技術水準から最も効率的に実現するためにはどのような機体設計と上昇飛行経路が必要となるか，実現にはどの程度の技術革新が必要なのかも合わせて考察する．

1.5 本論文の構成

本論文は 7 章から構成されている．

第 1 章は本章の序論であり，機体設計と飛行経路の統合的最適化に代表される複合領域最適化に関する重要性とこれまでの研究，および並列最適化法との関連をまとめた．また，本研究の目的，論文の概要を説明した．

第 2 章では，統合的最適化問題に代表される大規模最適化問題に対し，本論文で提案する並列最適化法の導出とアルゴリズムを説明し，合わせて軌道最適化問題に適用する方法を述べる．

第 3 章では第 2 章で提案された並列最適化法の有効性を調べるために簡単な軌道最適化問題を用意し適用する．この章の問題は理論解が求められる問題である．理論解と数値解を比較すると共に数値計算過程を調査し，本論文で提案する並列最適化法の特徴がまとめられる．

第 4 章より以降は，理論解が求められない最適化問題に対し，本論文の並列最適化法を適用していく．第 4 章で取り上げる問題は紙飛行機のサイズと投げ方の統合的最適化問題である．最も良く飛ぶ紙飛行機のサイズとその投げ方を求める．より複雑なスペースプレーンの統合的最適化問題へ並列最適化法を適用する前にその有効性を確認する．

第 5 章ではスペースプレーンの上昇飛行経路最適化問題に対して並列最適化法を適用する．従来の研究で求められていた解より高精度の解を得るために，大規模最適化問題を並列最適化法で解く．複雑な大規模最適化問題に並列最適化法を適用した場合の有効性を確認し，その特徴を調べる．

第 6 章では第 5 章で飛行経路のみの最適化を行ったスペースプレーンに関する問題を拡張し，機体設計も同時に最適化する．本論文で取り上げられる最適化問題のうち，最も難しい問題である．この章では並列最適化法の有効性を確認して，その特徴を調べることに合わせ，スペースプレーンの実現性についても考察する．

最終章である第 7 章では結論，今後の研究課題についてまとめる．

第2章
並列最適化法

本章では前章で述べた統合的最適化問題に代表される大規模最適化問題に対する並列最適化法を提案する．前章で述べたように，大規模システムは全ての変数を一括して最適化すると計算機に対して非常に大きな負荷をかけることになる．そこで大規模システムを幾つかの工学的意味をもつサブシステムに分け，大規模最適化問題を小規模な複数の部分最適化問題の集合として定義し直す．どのように部分最適化問題を定義するのか，部分最適化問題を解いて得られる解をどのようにまとめてシステム全体の最適解を得るのかが研究の焦点となる．ここでは，まず最適化手法のなかで現在最も有力であるとされる逐次 2 次計画法を，前章で述べた「解法の並列化」に注目して並列化することを考える．こうして得られた並列計算法は，解釈によっては「問題の並列化」を行ったことにもなっている．定義された部分最適化問題を個々の各サブシステムに当てはめることで，並列最適化法によって大規模最適化問題を解くことができる．一連の導出過程から，この解法は逐次 2 次計画法の特性を保ち，かつ優れた並列化効率をもつことが期待できる．それら有効性の確認は後の章で行われる．また，本章の最後で並列最適化法を軌道最適化問題に適用する方法を説明し，次章からの例題適用に備える．

2.1 大規模システムと大規模最適化問題

あるシステムの状態を表す変数を $x \in R^n$ とし，その変数に依存しシステムを特徴づける状態方程式を

$$g_E(x) = 0 \qquad (g_E : R^n \to R^{m_E}) \tag{2.1a}$$

$$g_I(x) \le 0 \qquad (g_I : R^n \to R^{m_I}) \tag{2.1b}$$

と表す．変数と状態方程式の次元 n，m_E，m_I が大きいシステムを大規模システムという．また，ある大規模システムの最適化問題

$$\text{variable} \quad x \tag{2.2a}$$

$$\text{minimize} \quad f(x) \tag{2.2b}$$

$$\text{subject to} \quad g_E(x) = 0 \tag{2.2c}$$

$$g_I(x) \leq 0 \tag{2.2d}$$

を，本論文では大規模最適化問題，あるいは単に大規模問題と呼ぶ．ここで，関数 f を目的関数，関数 g_E を等式制約関数，関数 g_I を不等式制約関数といい，式(2.2c)，(2.2d)をそれぞれ等式制約条件，不等式制約条件と呼ぶ．また，等式及び不等式制約関数，等式及び不等式制約条件を合わせて，それぞれ制約関数，制約条件と呼ぶこともある．最適化問題に関するより詳しい説明は補遺Aを参照されたい．

「大きい」「大規模」という言葉は主観的な表現であるが，その程度は実際に最適化計算を行う計算機の性能，最適化手法に依存する．大きなシステムでは，システム全体を一挙に扱って総合的な解析や設計を行うことは困難な場合が多い．よって，大規模最適化問題を従来の手法で解こうとすると，多大な計算時間を必要とし，また計算機に大きな負荷をかけることになる．

2.2 大規模システムの分割

一般に，現実に存在する大規模システムと大規模最適化問題は幾つかの小さいサブシステムとそこから定義される最適化問題に分解できる場合が多い．例えば，航空機の機体形状の設計問題を例に挙げると，機体（大規模システム）をその構成要素である胴体，翼，エンジンなど（サブシステム）に分け，機体形状全体の最適化問題を各構成要素の最適化問題に分割することが考えられる．また，航空機，宇宙機の運用も視野に入れたシステムを最適化する問題は，決定しなければならない変数が多く，複数の専門分野が関わり合う複合領域最適化問題といえ，典型的な大規模システムに対する大規模問題といえる．しかし，この大規模システムは，空力，熱，構造，材料，原動機，制御などの専門分野（サブシステム）が統合されて構成されている．よって，この大規模問題はこれら専門分野ごとの最適化問題に分解することができるはずである．

いま，ある大規模システムが N 個のサブシステムから成るとする．その上で，これら

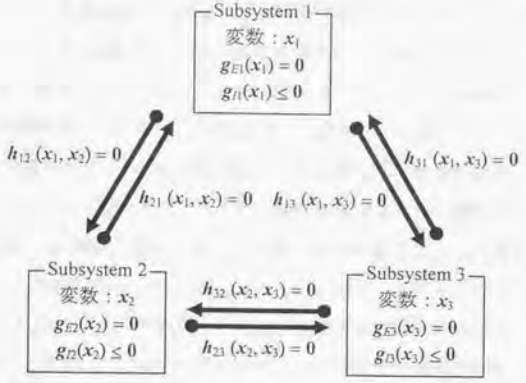


図2.1　3つのサブシステムからなる大規模システムの例

サブシステムの1つである i 番目のサブシステム（サブシステム i と呼ぶ）の状態を表す変数を $x_i \in R^{n_i}$ とし，サブシステム i を特徴づける等式制約条件，不等式制約条件を

$$g_{Ei}(x_i)=0 \qquad (g_{Ei}:R^{n_i}\to R^{m_{Ei}}) \tag{2.3a}$$

$$g_{Ii}(x_i)\le 0 \qquad (g_{Ii}:R^{n_i}\to R^{m_{Ii}}) \tag{2.3b}$$

と表す．ここで，個々のサブシステムは完全に独立しているのではないことに注意する．すなわち，サブシステム i の変数 x_i はある他のサブシステム j の変数 x_j と等式制約条件，

$$\begin{bmatrix} h_{ij}(x_i,x_j) \\ h_{ji}(x_j,x_i) \end{bmatrix}=0 \tag{2.4}$$

によって結び付けられている（カップリング）．ここで，関数 $h_{ij}:R^{n_i+n_j}\to R^{c_{ij}}$，$h_{ji}:R^{n_j+n_i}\to R^{c_{ji}}$ を接続関数（conjunctive function）と呼ぶことにし，式(2.4)の等式制約条件を接続条件（conjunctive condition）と呼ぶことにする．なお，接続条件を式(2.4)のように2つに分ける理由は後述する．また，本章で述べる並列最適化法における接続関数は，後に述べる理由により，変数 x_i のみの関数と，変数 x_j のみの関数に分離された形式

$$h_{ij}(x_i,x_j)\equiv h_{ij}^i(x_i)+h_{ij}^j(x_j) \qquad (h_{ij}^i:R^{n_i}\to R^{c_{ij}},\ h_{ij}^j:R^{n_j}\to R^{c_{ij}}) \tag{2.5}$$

をしていることが要求される．接続関数がこのような形式であることは特別なことではな

い．後に例題として示される全ての問題の接続条件はあるサブシステム i の変数 $\boldsymbol{x}_i$ の一部と別のサブシステム j の変数 $\boldsymbol{x}_j$ の一部が等しくなるという条件式で表される．よって式(2.5)の形式は自然に満たされる．サブシステム間のカップリングの表現方法は分割解法に依存して様々な形式が考えられるが，本論文では上述のように表すことにする．

図2.1はある大規模システムが 3 つのサブシステムからなる場合の変数，制約条件，接続条件を示している．この図で，サブシステム間に存在する 2 つの接続条件の違いを矢印で示している．沿え字の順番の違いによって矢印の向きが違うことに注意する．すなわち，$\boldsymbol{h}_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{0}$ という接続条件は，サブシステム i からサブシステム j に向かう矢印で表現する．なお，この矢印はデータの流れ，または解析順序を示しているのではない．また，後の章では，サブシステム i からサブシステム j に向かう矢印で表現される接続条件 $\boldsymbol{h}_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{0}$ に含まれる変数に対し，サブシステム i の側の変数を自由度が大きい変数，サブシステム j の側の変数を自由度が小さい変数と呼ぶことがある．その理由と意味については後に述べる．

ところで，従来の分割解法ではサブシステム i と j の間にある接続関数の次元 $c_{ij}+c_{ji}$ は変数 $\boldsymbol{x}_i$，$\boldsymbol{x}_j$ の次元 n_i，n_j，制約条件式の次元 $m_{Ei}+m_{Ii}$，$m_{Ej}+m_{Ij}$ に比べて小さいことが望ましいとされている．このサブシステム間の接続関係の相対的な大きさ（カップリングの強さ）が並列最適化法のロバスト性と収束性に関わっていることは第 1 章で述べた．すなわち，接続関数値の変動がサブシステムの解を大きく変化させるようだと，従来の分割解法のロバスト性と収束性は悪化する．そこで，本論文では取り上げないが，カップリングができるだけ弱くなるように大規模システムをサブシステムに分割する方法も研究されている．例えば，変数，目的関数，制約条件の相互依存性をブロックダイアグラムとして表現し，グラフ理論などを用いて分割する方法が検討されている[23]が，まだ多くの議論の余地があり，最終的にはヒューリスティックな判断が求められるのが現状である．後に示されるが，本論文で提案する並列最適化法の特徴の 1 つはカップリングが強くても優れた収束性とロバスト性を持つことである．よって，上述のようにサブシステムと接続条件が定義さえされていれば，カップリングが強くても，また必ずしも大規模システムでなくても構わない．

2.3 全体最適化

2.3.1 全体最適化問題の定義

各サブシステムの最適化する変数，制約条件，接続条件をまとめ，最小化を行う目的関数を定義すると，次のような大規模最適化問題が定義できる．

$$\text{variable} \quad x \equiv \left(x_1^{\mathrm{T}}, x_2^{\mathrm{T}}, \ldots, x_N^{\mathrm{T}}\right)^{\mathrm{T}} \in R^{n_1+n_2+\cdots+n_N} \tag{2.6a}$$

$$\text{minimize} \quad f(x_1, x_2, \ldots, x_N) \tag{2.6b}$$

$$\text{subject to} \quad g_E(x) \equiv \begin{bmatrix} g_{E1}(x_1) \\ \vdots \\ g_{EN}(x_N) \end{bmatrix} = 0 \quad \in R^{m_E} \tag{2.6c}$$

$$g_I(x) \equiv \begin{bmatrix} g_{I1}(x_1) \\ \vdots \\ g_{IN}(x_N) \end{bmatrix} \le 0 \quad \in R^{m_I} \tag{2.6d}$$

$$h(x) \equiv \begin{bmatrix} h_{12}(x_1, x_2) \\ h_{21}(x_2, x_1) \\ \vdots \\ h_{ij}(x_i, x_j) \\ h_{ji}(x_j, x_i) \\ \vdots \\ h_{N-1N}(x_{N-1}, x_N) \\ h_{NN-1}(x_N, x_{N-1}) \end{bmatrix} = 0 \quad \in R^{c} \tag{2.6e}$$

この大規模最適化問題を，システム全体の最適化問題を一括して表したという意味で全体最適化問題，あるいは全体問題といい，この解法を全体最適化法と呼ぶ．

式(2.6a)～(2.6e)の全体問題に対する最適解を x^* とすると，これが最適解であるための最適性の条件は Kuhn-Tucker 条件（補遺A）によって与えられる．まず，ラグランジュ関数を以下のように定義する．

$$L(x, \lambda_E, \lambda_I, \mu) \equiv f(x) + \lambda_E{}^{\mathrm{T}} g_E(x) + \lambda_I{}^{\mathrm{T}} g_I(x) + \mu^{\mathrm{T}} h(x) \tag{2.7}$$

ここで，$\lambda_E \in R^{m_E}$，$\lambda_I \in R^{m_I}$，$\mu \in R^{c}$ はそれぞれ等式制約条件，不等式制約条件，接続条件に対するラグランジュ乗数である．関数 f，g_E，g_I が C^1 級であるとき，x^* が最適解ならば，

$$\nabla_x^T L(x^*, \lambda_E^*, \lambda_I^*, \mu^*) = 0 \tag{2.8a}$$

$$g_E(x^*) = 0 \tag{2.8b}$$

$$g_I(x^*) \le 0 \tag{2.8c}$$

$$\lambda_{Ik}^* g_{Ik}(x^*) = 0 \tag{2.8d}$$

$$\lambda_I^* \ge 0 \tag{2.8e}$$

$$h(x^*) = 0 \tag{2.8f}$$

を満たす λ_E^*, λ_I^*, μ^* が存在する．ここで，λ_{Ik}^*, g_{Ik} はラグランジュ乗数 λ_I^* と不等式制約関数 g_I の第 k 成分である．

2.3.2 全体最適化法

全体最適化法の有力な解法として逐次 2 次計画（Sequential Quadratic Programming: SQP）法がある．詳細は補遺Aで説明されているが，この解法の基本的アルゴリズムは，まず近似解 x において，変数の更新量 Δx を次の 2 次計画（Quadratic Programming: QP）問題から求め，

$$\text{variable} \quad \Delta x \tag{2.9a}$$

$$\text{minimize} \quad \nabla_x f \Delta x + \frac{1}{2} \Delta x^T \nabla_{xx}^2 L \Delta x \tag{2.9b}$$

$$\text{subject to} \quad g_E + \nabla_x g_E \Delta x = 0 \tag{2.9c}$$

$$g_I + \nabla_x g_I \Delta x \le 0 \tag{2.9d}$$

$$h + \nabla_x h \Delta x = 0 \tag{2.9e}$$

1 次元探索によってステップ幅 α を求めた後に，変数値を x から x^* に

$$x^* = x + \alpha \Delta x \tag{2.10}$$

と更新し，これらを繰り返す．なお，以後，変数に沿え字の + が付けられた場合，次の繰り返し計算の変数値を意味する．また，式(2.9b)の目的関数の第 2 項目にある $\nabla_{xx}^2 L$ はラグランジュ関数 L の変数 x によるヘッセ行列を意味するが，SQP 法では近似行列によって代用される（補遺A）．

ここで，式(2.9a)～(2.9e)の QP 問題を解くことは，式(2.9d)の不等式制約条件の中から活性になる制約条件（解 Δx を代入すると等号を満たす制約条件）を見つけ，次の連立 1 次

方程式を解くことに等しい．

$$\begin{bmatrix} \nabla_{xx}^2 L & \nabla_x g(x)^T & \nabla_x h(x)^T \\ \nabla_x g(x) & & \\ \nabla_x h(x) & \mathbf{0} & \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda^+ \\ \mu^+ \end{bmatrix} = - \begin{bmatrix} \nabla_x f(x)^T \\ g(x) \\ h(x) \end{bmatrix} \tag{2.11}$$

なお，等式制約条件と不等式制約条件のうち活性な制約条件の制約関数をまとめて関数 g，その制約条件に対するラグランジュ乗数をλとし，ラグランジュ関数を

$$L(x,\lambda,\mu) \equiv f(x) + \lambda^T g(x) + \mu^T h(x) \tag{2.12}$$

と再定義した．また，ラグランジュ乗数μ，λの更新後の値を μ^+，λ^+ とした．大規模最適化問題である全体最適化問題を SQP 法で解くためには，この大規模連立 1 次方程式を解かなくてはならない．多くの計算時間とメモリがこの方程式の求解に費やされる．

ところで，補遺Aで述べられているが，SQP 法とは，未知変数を

$$y \equiv \left(x^T, \lambda^T, \mu^T\right)^T \tag{2.13}$$

とし，最適解において活性となる制約条件を見つけ，Kuhn-Tucker 条件から得られる次の非線形方程式

$$\nabla_y L(y)^T = \begin{bmatrix} \nabla_x L(x,\lambda,\mu)^T \\ \nabla_\lambda L(x,\lambda,\mu)^T \\ \nabla_\mu L(x,\lambda,\mu)^T \end{bmatrix} = \begin{bmatrix} \nabla_x L(x,\lambda,\mu)^T \\ g(x) \\ h(x) \end{bmatrix} = \mathbf{0} \tag{2.14}$$

を Newton 法で解くことに等しい．Newton 法では，反復計算中のある近似解 y における解の更新量 Δy は，

$$\nabla_{yy}^2 L(y) \Delta y = -\nabla_y L(y)^T \tag{2.15}$$

すなわち，

$$\begin{bmatrix} \nabla_{xx}^2 L & \nabla_x g(x)^T & \nabla_x h(x)^T \\ \nabla_x g(x) & & \\ \nabla_x h(x) & \mathbf{0} & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\mu \end{bmatrix} = - \begin{bmatrix} \nabla_x L(x,\lambda,\mu)^T \\ g(x) \\ h(x) \end{bmatrix} \tag{2.16}$$

より求められ，反復計算1回で以下のように変数値 y を y^+ へ更新する．

$$y^+ = y + \Delta y \tag{2.17}$$

ここで，式(2.16)の1行目を抜き出すと，

$$\begin{aligned} \nabla_{xx}^2 L\Delta x + \nabla_x g^\top \Delta\lambda + \nabla_x h^\top \Delta\mu &= -\nabla_x L^\top \\ &= -\nabla_x f^\top - \nabla_x g^\top \lambda - \nabla_x h^\top \mu \\ \nabla_{xx}^2 L\Delta x + \nabla_x g^\top \lambda^+ + \nabla_x h^\top \mu^+ &= -\nabla_x f^\top \end{aligned} \tag{2.18}$$

と変形できることから，結局，式(2.16)は

$$\begin{bmatrix} \nabla_{xx}^2 L & \nabla_x g(x)^\top & \nabla_x h(x)^\top \\ \nabla_x g(x) & & \\ \nabla_x h(x) & \mathbf{0} & \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda^+ \\ \mu^+ \end{bmatrix} = -\begin{bmatrix} \nabla_x f(x)^\top \\ g(x) \\ h(x) \end{bmatrix} \tag{2.19}$$

となり，式(2.11)と等しくなる．ただし，SQP 法と Newton 法が異なる点は，SQP 法ではヘッセ行列 $\nabla_{xx}^2 L$ を近似行列とすることと，解の更新量を求めた後に 1 次元探索を行うことである．

本項での内容をまとめると次のようになる．

(1) SQP 法のアルゴリズムに含まれる QP 問題を解くことは，変数の次元と制約関数の次元の和に等しい数の次元をもつ連立 1 次方程式を解くことに等しい．よって，大規模最適化問題である全体最適化問題を SQP 法で解こうとすると，多くの計算時間とメモリ量がこの大規模連立1次方程式を解くことに費やされる．

(2) SQP 法は Kuhn-Tucker 条件から得られる非線形方程式を満たす変数とラグランジュ乗数を Newton 法によって求める方法に基本的に等しい．SQP 法の QP 問題を解くための連立1次方程式は，Kuhn-Tucker 条件から得られる非線形方程式を Newton 法で解くときに未知変数の更新量を求める連立1次方程式に等しい．

2.4 部分最適化問題の定義

2.4.1 並列最適化法の必要性

大規模な全体最適化問題を一括して最適化していくのではなく，各サブシステムから

なる複数の最適化問題を並列処理により解き，それらをまとめて全システムの最適解を得ようとする方法を並列最適化法と呼ぶ．また，各サブシステムで定義される最適化問題を部分最適化問題，あるいは単に部分問題と呼ぶことにする．どのように部分最適化問題を定義したら良いのか，また部分最適化問題の解をどのようにまとめていったら良いのか，それは最適解を効率的に得るための大きな問題点である．

あるサブシステム i における未知変数を y_i とすると，

$$y_i \equiv \left(x_i^{\mathrm{T}}, \lambda_i^{\mathrm{T}}, \mu_i^{\mathrm{T}}\right)^{\mathrm{T}} \tag{2.20}$$

となる．ここで，λ_i は式(2.11)で定義した活性な制約条件のうちサブシステム i に関する制約条件に対するラグランジュ乗数である．また，サブシステム i に関わる接続条件を式(2.4)のように 2 つに分けたが，そのうちの一方を集めた接続条件

$$h_i(x_1, x_2, \ldots, x_N) \equiv \begin{bmatrix} h_{1i}(x_1, x_i) \\ \vdots \\ h_{Ni}(x_N, x_i) \end{bmatrix} = 0 \tag{2.21}$$

に関するラグランジュ乗数を μ_i とした．よって，

$$\lambda = \left(\lambda_1^{\mathrm{T}}, \lambda_2^{\mathrm{T}}, \ldots, \lambda_N^{\mathrm{T}}\right)^{\mathrm{T}}, \quad \mu = \left(\mu_1^{\mathrm{T}}, \mu_2^{\mathrm{T}}, \ldots, \mu_N^{\mathrm{T}}\right)^{\mathrm{T}} \tag{2.22}$$

のようにラグランジュ乗数を分け，全体問題の未知変数 y はサブシステムごとの変数 y_i に分けられたことになる．

$$y = \left(y_1^{\mathrm{T}}, y_2^{\mathrm{T}}, \ldots, y_N^{\mathrm{T}}\right)^{\mathrm{T}} \tag{2.23}$$

これにより，式(2.14)のように表されていた全システムの最適性の条件は，各サブシステムごとの最適性の条件に分割される．

$$\nabla_y L(y)^{\mathrm{T}} = \begin{bmatrix} \nabla_{y_1} L(y)^{\mathrm{T}} \\ \nabla_{y_2} L(y)^{\mathrm{T}} \\ \vdots \\ \nabla_{y_N} L(y)^{\mathrm{T}} \end{bmatrix} = 0 \tag{2.24}$$

ここで，サブシステム i に関する最適性の条件を抜き出してみると，

$$\nabla_{y_i} L(y)^{\mathrm{T}} = \begin{bmatrix} \nabla_{x_i} L(y)^{\mathrm{T}} \\ g(x_i) \\ h_i(x_1, x_2, \ldots, x_N) \end{bmatrix} = \mathbf{0} \tag{2.25}$$

となっている．上のサブシステム i に関する最適性の条件はサブシステム i の未知変数 y_i に依存するのはいうまでもないが，他のサブシステムの変数 x_j とラグランジュ乗数 μ_j（$j \neq i$）にも弱いながらも依存することに注意する．他のサブシステムとの接続条件が存在するためである．よって，各サブシステムごとに完全に独立して最適化計算を行い，全システムの最適解を得ることは決してできない．

また，式(2.15)で示した Newton 法（SQP 法）の更新量 Δy を求める連立 1 次方程式は

$$\begin{bmatrix} \nabla^2_{y_1 y_1} L & \nabla^2_{y_1 y_2} L & \cdots & \nabla^2_{y_1 y_N} L \\ \nabla^2_{y_2 y_1} L & \nabla^2_{y_2 y_2} L & \cdots & \nabla^2_{y_2 y_N} L \\ \vdots & \vdots & & \vdots \\ \nabla^2_{y_N y_1} L & \nabla^2_{y_N y_2} L & \cdots & \nabla^2_{y_N y_N} L \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_N \end{bmatrix} = - \begin{bmatrix} \nabla_{y_1} L^{\mathrm{T}} \\ \nabla_{y_2} L^{\mathrm{T}} \\ \vdots \\ \nabla_{y_N} L^{\mathrm{T}} \end{bmatrix} \tag{2.26}$$

となる．なお，Δy_i は変数 y_i の更新量であり，全システムの更新量 Δy の一部でもある．この左辺の行列の中で，非対角成分 $\nabla_{y_i y_j} L$（$i \neq j$）は接続関数によって存在する．

並列計算によりこの大規模連立 1 次方程式を解くことを考える．

2.4.2 Newton-like 法による部分最適化問題

式(2.24)の大規模非線形方程式の解を効率良く得るために，式(2.26)の連立 1 次方程式を正確に求めるのではなく並列計算によって近似解を求め，その近似解で変数値を更新していく方法が考えられる．その 1 つとして，式(2.25)の最適性の条件は変数 y_i に強く依存する点に注目すると，式(2.26)左辺の行列の対角要素である $\nabla_{y_i y_i} L$ に比べ，非対角要素の $\nabla_{y_i y_j} L$（$i \neq j$）のノルムは小さいと判断し，この非対角要素を省略して，

$$\begin{bmatrix} \nabla^2_{y_1 y_1} L & & & \mathbf{0} \\ & \nabla^2_{y_2 y_2} L & & \\ & & \ddots & \\ \mathbf{0} & & & \nabla^2_{y_N y_N} L \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_N \end{bmatrix} = - \begin{bmatrix} \nabla_{y_1} L^{\mathrm{T}} \\ \nabla_{y_2} L^{\mathrm{T}} \\ \vdots \\ \nabla_{y_N} L^{\mathrm{T}} \end{bmatrix} \tag{2.27a}$$

から変数の更新量 Δy を求める方法が考えられる．このとき，個々の更新量 Δy_i を並列計

算により求めることができる．すなわち，変数 y を y_i^+ へと次のように更新する．

$$\left[\nabla^2_{y_i y_i} L\right]\Delta y_i = -\nabla_{y_i} L^{\mathrm{T}} \tag{2.28a}$$

$$y_i^+ = y_i + \Delta y_i \tag{2.28b}$$

この N 個の計算はサブシステムごとに並列計算により処理することができる．

式(2.28a)を書き下してみると，

$$\begin{bmatrix} \nabla^2_{x_i x_i} L & \nabla_{x_i} g_i(x_i)^{\mathrm{T}} & \nabla_{x_i} h(x_i)^{\mathrm{T}} \\ \nabla_{x_i} g_i(x_i) & & \\ \nabla_{x_i} h_i(x_i) & \mathbf{0} & \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta\lambda_i \\ \Delta\mu_i \end{bmatrix} = -\begin{bmatrix} \nabla_{x_i} L^{\mathrm{T}} \\ g_i(x_i) \\ h_i(x_1,\ldots,x_N) \end{bmatrix} \tag{2.29}$$

となるため，サブシステム i は QP 問題

$$\text{variable} \quad \Delta x_i \tag{2.30a}$$

$$\text{minimize} \quad \nabla_{x_i} f \Delta x_i + \frac{1}{2}\Delta x_i{}^{\mathrm{T}} \nabla^2_{x_i x_i} L \Delta x_i \tag{2.30b}$$

$$\text{subject to} \quad g_{Ei} + \nabla_{x_i} g_{Ei} \Delta x_i = 0 \tag{2.30c}$$

$$g_{Ii} + \nabla_{x_i} g_{Ii} \Delta x_i \le 0 \tag{2.30d}$$

$$h_i + \nabla_{x_i} h_i \Delta x_i = 0 \tag{2.30e}$$

をもつ最適化問題を解くことになる．つまり，この解法はサブシステムの部分最適化問題

$$\text{variable} \quad x_i \tag{2.31a}$$

$$\text{minimize} \quad f(x_1, x_2, \ldots, x_N) \tag{2.31b}$$

$$\text{subject to} \quad g_{Ei}(x_i) = 0 \tag{2.31c}$$

$$g_{Ii}(x_i) \le 0 \tag{2.31d}$$

$$h_i(x_1, x_2, \ldots, x_N) = \begin{bmatrix} h_{1i}(x_1, x_i) \\ \vdots \\ h_{Ni}(x_N, x_i) \end{bmatrix} = 0 \tag{2.31e}$$

を並列計算により解き，得られた変数値 x_i （$i = 1,2,\ldots,N$）をサブシステム間で交換することを繰り返して最適解を得る方法に等しい．この部分最適化問題の定式化法と変数の更新法を Newton-like 法と呼ぶことにする．

Newton-like 法はサブシステムごとに関連した制約条件を集め，システム間の接続条件の感度情報を考慮することなく，各サブシステムを独立して最適化していく最も基本的な部分最適化問題の定義法である．しかし，この定式化では形式的に式(2.31b)の目的関数 f は全サブシステムの変数の関数であるとしたが，現実の問題では一部のサブシステムの変数にしか依存しない場合が多い．この場合，目的関数をもたないサブシステムが存在することになり，そのサブシステムは何を目的に最適化を行うのか不明となる．また，解が収束したとき，$\Delta y = \mathbf{0}$ が成り立つため，

$$\begin{bmatrix} \left[\nabla^2_{y_1y_1} L(y)\right]^{-1} \nabla_{y_1} L(y)^{\mathrm{T}} \\ \left[\nabla^2_{y_2y_2} L(y)\right]^{-1} \nabla_{y_2} L(y)^{\mathrm{T}} \\ \vdots \\ \left[\nabla^2_{y_Ny_N} L(y)\right]^{-1} \nabla_{y_N} L(y)^{\mathrm{T}} \end{bmatrix} = \mathbf{0} \tag{2.32}$$

をみたす変数値 y が求められる．これが式(2.24)の解とは限らないため，得られた最適解が全システムの最適解と一致する保証はない．

次に，この解法の収束性を調べてみる．式(2.28a)，(2.28b)より，

$$y^+ = y - \begin{bmatrix} \left[\nabla^2_{y_1y_1} L(y)\right]^{-1} \nabla_{y_1} L(y)^{\mathrm{T}} \\ \left[\nabla^2_{y_2y_2} L(y)\right]^{-1} \nabla_{y_2} L(y)^{\mathrm{T}} \\ \vdots \\ \left[\nabla^2_{y_Ny_N} L(y)\right]^{-1} \nabla_{y_N} L(y)^{\mathrm{T}} \end{bmatrix} \tag{2.33}$$

に従って変数 y を y^+ へと逐次更新していくが，変数 y の関数である行列 $\boldsymbol{H}$ を

$$\begin{aligned} \boldsymbol{H}(y) &\equiv \frac{\partial}{\partial y} \left\{ y - \begin{bmatrix} \left[\nabla^2_{y_1y_1} L(y)\right]^{-1} \nabla_{y_1} L(y)^{\mathrm{T}} \\ \left[\nabla^2_{y_2y_2} L(y)\right]^{-1} \nabla_{y_2} L(y)^{\mathrm{T}} \\ \vdots \\ \left[\nabla^2_{y_Ny_N} L(y)\right]^{-1} \nabla_{y_N} L(y)^{\mathrm{T}} \end{bmatrix} \right\} \\ &= -\begin{bmatrix} \mathbf{0} & \left[\nabla^2_{y_1y_1} L\right]^{-1} \left[\nabla^2_{y_1y_2} L\right] & \cdots & \left[\nabla^2_{y_1y_N} L\right]^{-1} \left[\nabla^2_{y_1y_N} L\right] \\ \left[\nabla^2_{y_2y_2} L\right]^{-1} \left[\nabla^2_{y_2y_1} L\right] & \mathbf{0} & \cdots & \left[\nabla^2_{y_2y_2} L\right]^{-1} \left[\nabla^2_{y_2y_N} L\right] \\ \vdots & \vdots & \ddots & \vdots \\ \left[\nabla^2_{y_Ny_N} L\right]^{-1} \left[\nabla^2_{y_Ny_1} L\right] & \left[\nabla^2_{y_Ny_N} L\right]^{-1} \left[\nabla^2_{y_Ny_2} L\right] & \cdots & \mathbf{0} \end{bmatrix} \end{aligned} \tag{2.34}$$

と定義すると，変数 y が最適解 y^* に収束する必要十分条件は，行列 $\boldsymbol{H}(y^*)$ のスペクトル

半径が1未満（行列$H(y^*)$の固有値の絶対値の最大値が1未満）である．すなわち，関数$\nabla_{y_i}L$が変数y_iに強く依存するときのみ最適解を得ることができる．関数$\nabla_{y_i}L$に含まれる変数y_j（$i\neq j$）は接続関数に含まれるため，接続条件がサブシステムの最適解に及ぼす影響が小さい．すなわちカップリングが弱い場合のみしか変数値は収束しない．

2.4.3 Jacobi-Newton 法による部分最適化問題

並列最適化法によって正確な最適解を求めるためには，式(2.26)の行列の非対角成分を考慮し，システム間の接続条件の感度情報を考慮する必要がある．よって，式(2.26)の連立1次方程式に Jacobi 法[23]を応用した変形を行い，

$$\begin{bmatrix} \nabla^2_{y_1y_1}L & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \nabla^2_{y_2y_2}L & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \nabla^2_{y_Ny_N}L \end{bmatrix} \begin{bmatrix} \Delta y_1^+ \\ \Delta y_2^+ \\ \vdots \\ \Delta y_N^+ \end{bmatrix} = -\begin{bmatrix} \nabla_{y_1}L^{\mathrm{T}} \\ \nabla_{y_2}L^{\mathrm{T}} \\ \vdots \\ \nabla_{y_N}L^{\mathrm{T}} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \nabla^2_{y_1y_2}L & \cdots & \nabla^2_{y_1y_N}L \\ \nabla^2_{y_2y_1}L & \mathbf{0} & \cdots & \nabla^2_{y_2y_N}L \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^2_{y_Ny_1}L & \nabla^2_{y_Ny_2}L & \cdots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_N \end{bmatrix} \tag{2.35}$$

とする繰り返し計算により正確な更新量Δy_iを求める．すなわち次のような反復計算が行われる．

Step 1.

初期の変数量yを適当に仮定する．

Step 2.

更新量Δyを適当に仮定する．

Step 3.

ある更新量Δyから，次の更新量Δy^+を

$$\Delta y^+ = H(y)\Delta y + c(y) \tag{2.36a}$$

$$H(y) \equiv -\begin{bmatrix} \mathbf{0} & \left[\nabla^2_{y_1y_1}L\right]^{-1}\left[\nabla^2_{y_1y_2}L\right] & \cdots & \left[\nabla^2_{y_1y_1}L\right]^{-1}\left[\nabla^2_{y_1y_N}L\right] \\ \left[\nabla^2_{y_2y_2}L\right]^{-1}\left[\nabla^2_{y_2y_1}L\right] & \mathbf{0} & \cdots & \left[\nabla^2_{y_2y_2}L\right]^{-1}\left[\nabla^2_{y_2y_N}L\right] \\ \vdots & \vdots & \ddots & \vdots \\ \left[\nabla^2_{y_Ny_N}L\right]^{-1}\left[\nabla^2_{y_Ny_1}L\right] & \left[\nabla^2_{y_Ny_N}L\right]^{-1}\left[\nabla^2_{y_Ny_2}L\right] & \cdots & \mathbf{0} \end{bmatrix} \tag{2.36b}$$

$$c(y) \equiv -\begin{bmatrix} \left[\nabla^2_{y_1 y_1} L(y)\right]^{-1} \nabla_{y_1} L(y)^{\mathrm{T}} \\ \left[\nabla^2_{y_2 y_2} L(y)\right]^{-1} \nabla_{y_2} L(y)^{\mathrm{T}} \\ \vdots \\ \left[\nabla^2_{y_N y_N} L(y)\right]^{-1} \nabla_{y_N} L(y)^{\mathrm{T}} \end{bmatrix} \tag{2.36c}$$

によって求める．上式を各更新量 Δy_i （$i = 1,2,\ldots,N$）の更新式ごとに分けて表すと，

$$\left[\nabla^2_{y_i y_i} L\right] \Delta y_i^{+} = -\nabla_{y_i} L^{\mathrm{T}} - \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \nabla^2_{y_i y_j} L \Delta y_j \tag{2.37}$$

になるため，更新量 Δy の計算は並列処理が可能である．

Step 4.

Δy^{+} と Δy が等しいと見なせるならば，Jacobi 法は収束したと判断して次の Step 5.に進む．さもなければ，更新量 Δy を Δy^{+} に変更して Step 3.に戻る．

Step 5.

Δy が十分に $\mathbf{0}$ に近いならば，変数 y は収束したと判断して反復計算を停止する．さもなければ，

$$y^{+} = y + \Delta y \tag{2.38}$$

より得られる変数値 y^{+} に変数 y の値を更新して Step 2.へ戻る．

この解法では，更新量 Δy を求める Step 3～4.までの Jacobi 法の反復計算と，変数 y を更新して最適解へ収束させる Step 2～5.の Newton 法の反復計算からなる合わせて 2 つの繰り返し計算を行っている．前者の反復計算を inner loop，後者の反復計算を outer loop と呼ぶことにする．本論文では非線形方程式に対する Newton 法アルゴリズムに含まれる連立 1 次方程式を Jacobi 法で求めたことから，この解法を Jacobi-Newton 法と呼ぶことにする．

ここで，

$$\nabla_{y_i y_i} L = \begin{bmatrix} \nabla^2_{x_i x_i} L & \nabla_{x_i} g_i(x_i)^{\mathrm{T}} & \nabla_{x_i} h_i(x_i)^{\mathrm{T}} \\ \nabla_{x_i} g_i(x_i) & \mathbf{0} & \\ \nabla_{x_i} h_i(x_i) & & \end{bmatrix} \tag{2.39a}$$

$$\nabla_{y_i y_j} L = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \nabla_{x_i} \boldsymbol{h}_{ij}(\boldsymbol{x}_i)^{\mathrm{T}} \\ \mathbf{0} & & \mathbf{0} \\ \nabla_{x_j} \boldsymbol{h}_i(\boldsymbol{x}_j) & & \end{bmatrix} \tag{2.39b}$$

であるから，式(2.37)を書き下してみると，

$$\begin{bmatrix} \nabla^2_{x_i x_i} L & \nabla_{x_i} \boldsymbol{g}_i(\boldsymbol{x}_i)^{\mathrm{T}} & \nabla_{x_i} \boldsymbol{h}_i(\boldsymbol{x}_i)^{\mathrm{T}} \\ \nabla_{x_i} \boldsymbol{g}_i(\boldsymbol{x}_i) & \mathbf{0} & \\ \nabla_{x_i} \boldsymbol{h}_i(\boldsymbol{x}_i) & & \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_i \\ \Delta \boldsymbol{\lambda}_i \\ \Delta \boldsymbol{\mu}_i \end{bmatrix} = - \begin{bmatrix} \nabla_{x_i} L^{\mathrm{T}} + \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \nabla_{x_i} \boldsymbol{h}_{ij}(\boldsymbol{x}_i)^{\mathrm{T}} \Delta \boldsymbol{\mu}_{ij} \\ \boldsymbol{g}_i(\boldsymbol{x}_i) \\ \boldsymbol{h}_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) + \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \nabla_{x_j} \boldsymbol{h}_i(\boldsymbol{x}_j) \Delta \boldsymbol{x}_j \end{bmatrix} \tag{2.40}$$

となるため，サブシステム i は次に示す QP 問題を解くこととなる．

$$\text{variable} \quad \Delta \boldsymbol{x}_i \tag{2.41a}$$

$$\text{minimize} \quad \nabla_{x_i} \left(f + \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \boldsymbol{\mu}_{ij}{}^{\mathrm{T}} \boldsymbol{h}_{ij} \right) \Delta \boldsymbol{x}_i + \frac{1}{2} \Delta \boldsymbol{x}_i{}^{\mathrm{T}} \nabla^2_{x_i x_i} L \Delta \boldsymbol{x}_i \tag{2.41b}$$

$$\text{subject to} \quad \boldsymbol{g}_{Ei} + \nabla_{x_i} \boldsymbol{g}_{Ei} \Delta \boldsymbol{x}_i = \mathbf{0} \tag{2.41c}$$

$$\boldsymbol{g}_{Ii} + \nabla_{x_i} \boldsymbol{g}_{Ii} \Delta \boldsymbol{x}_i \leq \mathbf{0} \tag{2.41d}$$

$$\boldsymbol{h}_i + \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \nabla_{x_j} \boldsymbol{h}_i \Delta \boldsymbol{x}_j + \nabla_{x_i} \boldsymbol{h}_i \Delta \boldsymbol{x}_i = \mathbf{0} \tag{2.41e}$$

つまり，Jacobi-Newton 法とは，各サブシステムの部分最適化問題が

$$\text{variable} \quad \boldsymbol{x}_i \tag{2.42a}$$

$$\text{minimize} \quad f(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) + \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \boldsymbol{\mu}_{ij}{}^{\mathrm{T}} \boldsymbol{h}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2.42b}$$

$$\text{subject to} \quad \boldsymbol{g}_{Ei}(\boldsymbol{x}_i) = \mathbf{0} \tag{2.42c}$$

$$\boldsymbol{g}_{Ii}(\boldsymbol{x}_i) \leq \mathbf{0} \tag{2.42d}$$

$$\boldsymbol{h}_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) = \begin{bmatrix} \boldsymbol{h}_{1i}(\boldsymbol{x}_1, \boldsymbol{x}_i) \\ \vdots \\ \boldsymbol{h}_{Ni}(\boldsymbol{x}_N, \boldsymbol{x}_i) \end{bmatrix} = \mathbf{0} \tag{2.42e}$$

で表される問題を並列計算により解き，得られた変数値 x_i （$i=1,2,\ldots,N$）と接続条件に対するラグランジュ乗数 μ_i をサブシステム間で交換することを繰り返して最適解を得る方法である．

なお，式(2.5)の接続関数の定義において，変数 x_i と x_j を分離した理由は，

$$\nabla_{x_i x_j} L = 0 \quad (i \neq j) \tag{2.43}$$

を成り立たせるためである．上式が成り立たないと，式(2.42a)～(2.42e)のように部分最適化問題を単純な形で表すことができない．

ここで最適化を行う変数に注意する．サブシステム i の部分最適化問題において最適化を行う変数は x_i のみであり，同時に制約条件の一部である接続条件に対するラグランジュ乗数 μ_i を求めることが要求されている．また，接続関数の現れ方に注意が必要である．サブシステム i に関わる接続関数のうち h_{ji} （$j=1,2,\ldots,N$，$j \neq i$），すなわち沿え字の i が後に置かれている接続関数は制約関数として定義されている．また，サブシステム i に関わる残りの接続関数 h_{ij}，すなわち沿え字の i が前に置かれている接続関数はその接続条件に関するラグランジュ乗数 μ_{ij} と掛け合わされて目的関数に加えられている．サブシステム i の目的関数に現れる接続関数は接続相手のもう一方のサブシステムでは制約条件として扱われており，そこでラグランジュ乗数値が得られる．

この部分最適化問題の定式化は，サブシステム間の接続関係をラグランジュ乗数によって目的関数にした点で，第 1 章と補遺Bで述べられているゴール調整法の定式化に似ている．しかし，それらラグランジュ乗数値の求め方が異なる．

本解法ではラグランジュ乗数という接続条件に対する感度情報を積極的に利用していく点が特徴といえる．もし，あるサブシステムの部分最適化問題（式(2.42a)～(2.42e)）を解く解法がラグランジュ乗数を求められない方法であるならば，拡張ラグランジュ乗数法（augmented Lagrange multiplier method）に見られるラグランジュ乗数の推定法[34]を使うことが可能であると思われる．

また，Jacobi 法により式(2.26)を正確に解いているため，この並列最適化法で最適解が得られれば，それが全システムの最適解となっている．さらに，前項で定式化された Newton-like 法による部分最適化問題では式(2.31b)の目的関数が存在しない場合があったが，ここでの問題設定法では接続条件を式(2.4)のように適当に分ければ，必ず目的関数を定義することができ，各部分問題の意味がはっきりする．

2.5 並列最適化法

2.5.1 Aitken-Jacobi-Newton 法

Jacobi-Newton 法の収束条件について考える．2 つの反復計算のうち，outer loop は Newton 法すなわち SQP 法による解の更新と一致し，変数値の最適解への収束性は保証される．一方，inner loop が局所収束性をもつための必要十分条件は，式(2.36b)の行列 $\boldsymbol{H}(y)$ のスペクトル半径が 1 未満となることである．つまり，Jacobi-Newton 法の収束条件は Newton-like 法の収束条件と同じであり，カップリングが強い場合，収束性は保証されない．

そこで，変数 Δy が式(2.36a)の更新公式に従って自然に収束するのを期待するのではなく，Aitken の加速法[25] を応用した更新法を適用することで収束性を改善することにする．式(2.36a)の Δy^+ を $\Delta\tilde{y}$ に改め，

$$\Delta\tilde{y} = H\Delta y + c \tag{2.44}$$

とし，関数 Ψ を

$$\begin{aligned}\Psi(\Delta y) &\equiv \Delta y - \Delta\tilde{y} \\ &= (I - H)\Delta y - c\end{aligned} \tag{2.45}$$

と定義すると，Δy が inner loop の収束値であれば，

$$\Psi(\Delta y) = \Delta y - \Delta\tilde{y} = 0 \tag{2.46}$$

が成り立たなければならない．そこで，

$$\Delta y^+ = \Delta y - \kappa B\,\Psi(\Delta y) \tag{2.47}$$

という更新法により Δy を Δy^+ へ更新する．ここで，κ は $0 < \kappa \leq 1$ を満たす定数である．また，行列 $\boldsymbol{B}$ は，

$$\left[\frac{\partial\Psi(\Delta y)}{\partial\Delta y}\right]^{-1} = (I - H)^{-1} \qquad (I：単位行列) \tag{2.48}$$

の近似行列である．もし，$\kappa = 1$ として，行列 $\boldsymbol{B}$ が正確に上式の行列と一致するならば，inner loop は 1 回の計算で収束して Δy を得ることができる．

行列 $I-H$ の構造を調べてみると，

$$I-H=\begin{bmatrix} I & \left[\nabla^2_{y_1y_1}L\right]^{-1}\left[\nabla^2_{y_1y_2}L\right] & \cdots & \left[\nabla^2_{y_1y_N}L\right]^{-1}\left[\nabla^2_{y_1y_N}L\right] \\ \left[\nabla^2_{y_2y_2}L\right]^{-1}\left[\nabla^2_{y_2y_1}L\right] & I & \cdots & \left[\nabla^2_{y_2y_2}L\right]^{-1}\left[\nabla^2_{y_2y_N}L\right] \\ \vdots & \vdots & \ddots & \vdots \\ \left[\nabla^2_{y_Ny_N}L\right]^{-1}\left[\nabla^2_{y_Ny_1}L\right] & \left[\nabla^2_{y_Ny_N}L\right]^{-1}\left[\nabla^2_{y_Ny_2}L\right] & \cdots & I \end{bmatrix} \tag{2.49}$$

となっている．$\left[\nabla_{y_iy_i}L\right]$ に比べ $\left[\nabla_{y_iy_j}L\right]$ $(i\neq j)$ のノルムは小さいため，行列 $I-H$ の非対角要素は零行列に近い疎行列となっている．よって，行列 $I-H$ は単位行列 I に近い行列である．また，行列 H の成分は式(2.44)の計算をするとき求められるため，行列 $I-H$ を保存するメモリ量は少なく済み，逆行列の計算時間もそれほどかからない．

行列 $I-H$ は式(2.26)の左辺の行列と同じ大きさをもつ大規模な行列である．そのため，式(2.26)を直接計算して Δy を求めたほうが速いように思える．しかし，実際の計算は，並列計算による効果，行列 $I-H$ の疎性により，ここで説明した過程のほうが少ないメモリ量，短い計算時間で Δy を求められる．また，式(2.47)で求める変数 Δy^+ には Δx^+ に加えて全てのラグランジュ乗数の更新値 $\Delta\lambda^+$，$\Delta\mu^+$ が含まれている．しかし，部分最適化問題の定式化において必要なラグランジュ乗数は接続条件に対するラグランジュ乗数 $\Delta\mu^+$ のみである．よって，Δy^+ から $\Delta\lambda^+$ を除いて式(2.47)の行列 B，すなわち行列 $I-H$ の大きさを縮小することが可能である．

もし行列 $(I-H)^{-1}$ が求められないならば，行列 B に行列 $(I-H)^{-1}$ の近似行列を用いれば良い．例えば，ある inner loop での値 Δy と次の繰り返し回数での値 Δy^+ から

$$v=\Psi(\Delta y^+)-\Psi(\Delta y) \tag{2.50a}$$

$$u=\Delta y^+-\Delta y \tag{2.50b}$$

の2つのベクトルを定義したとき，secant 条件と呼ばれる

$$Bv=u \tag{2.51}$$

を満足する行列 B を式(2.47)で用いれば，この行列 B は式(2.48)の良い近似になっているはずである．そこで，inner loop の開始時には $B=I$ として，次の least change secant update[26,27]で式(2.51)が成り立つ行列 B を逐次近似していくことができる．

$$B^{+} = B + \frac{(u - Bv)v^{\mathrm{T}}}{v^{\mathrm{T}}v} \tag{2.52}$$

また，行列 B は疎行列であり特徴的な構造をしているため，行列 B の特定の成分のみに least change secant update を適用することで収束性を高めることが可能である．なお，後の章の問題を解く際には，行列 $I-H$ が求められるために逆行列を直接求める方法を採用している．

この Aitken 法は前章で述べたシステム全体の情報を利用する coordination に相当するが，従来の解法のような最適化問題として定義されず，単純な行列計算が行われるにすぎない．本章で示した部分最適化問題の定式化法とそれらの解法をまとめた並列最適化法を Aitken-Jacobi-Newton 法と呼ぶことにする．

2.5.2 1 次元探索

通常，SQP 法では QP 問題から変数の更新量を求めた後で一次元探索を行い，真の更新量を求める．補遺Aで述べられているように，1 次元探索を行うことは解の大域的収束性を保証するために重要である．本章における並列最適化法において，1 次元探索は部分最適化問題により更新量 Δx_i が求められた後にサブシステムごとに並列に行い，更新量 Δx_i に対するステップ幅 α_i（$i=1,2,\ldots,N$）を決定することにする．1 次元探索をサブシステムごとに分かれて行う理由は，システム全体で一括して 1 次元探索を行った場合，関数解析は各サブシステムごとに行われるため，ステップ幅，各サブシステムの関数解析の開始信号とその結果についてのやり取りが計算プロセス間で頻繁に行われ，並列化効率が悪化するためである．また，後の章で述べるが，各サブシステムごとに独自のステップ幅を設定したほうが優れた最適解が得られるという特性があるためでもある．

一次元探索をサブシステムごとに分かれて行うと，行列 $I-H$ を正確に計算すること難しい．なぜなら，サブシステム i の一次元探索の結果であるステップ幅 α_i は変数 y_i に依存するためである．しかし，ここではそのような依存は存在しないと仮定して，

$$\left[\nabla_{y_i y_i} L\right]^{-1} = \begin{bmatrix} \alpha_i I & 0 & 0 \\ 0 & & I \\ 0 & & \end{bmatrix} \begin{bmatrix} \nabla_{x_i x_i} L & \nabla_{x_i}^{\mathrm{T}} g_i(x_i) & \nabla_{x_i}^{\mathrm{T}} h(x_i) \\ \nabla_{x_i} g_i(x_i) & & 0 \\ \nabla_{x_i} h_i(x_i) & & \end{bmatrix}^{-1} \tag{2.53}$$

により行列 $I-H$ を近似することにする．この場合，inner loop の計算ごとにステップ幅

α_iが変化すると，Δyが振動してしまい$\Psi(\Delta y)$が$\mathbf{0}$に向かって収束しなくなる．そこで，式(2.47)のパラメータκの値を変化させる．inner loopの開始時には$\kappa=1$とし，Δyが振動したときパラメータκ値を小さくする．inner loopの回数は計算時間に大きく関わるため，inner loopをある程度繰り返しても更新量Δyが収束しないときにはそのままouter loopに進むことにする．さらに，最適解の近傍では変数の更新量，またステップ幅は小さな値となり，inner loopは必ず収束する．

2.5.3 不等式制約条件

不等式制約条件が存在すると，inner loopの間，幾つかの不等式制約条件の活性と不活性が移り変わり，理論的には正確な行列$\boldsymbol{I}-\boldsymbol{H}$を求めることはできない．しかしながら，各inner loopで活性な不等式制約条件のみを考慮した行列$\boldsymbol{H}$から行列$(\boldsymbol{I}-\boldsymbol{H})^{-1}$を計算して行列$\boldsymbol{B}$にする．そして，前項で述べたように式(2.47)のパラメータκを変化させて収束させるが，多くの繰り返し回数を経ても収束しないときはそのままouter loopに進む．すなわち，不等式制約条件の活性と不活性の状態が移り変わることによってΔyが振動してしまったとき，パラメータκの値を小さくする．最適解の近傍では活性な不等式制約条件が変化することは少ないため，inner loopは必ず収束する．

1次元探索と不等式制約条件が存在すると，inner loopに多くの繰り返し計算が必要となるが，後章の例題への適用によってinner loopとouter loopの収束性に問題がないことは確認されている．

2.6 並列最適化アルゴリズム

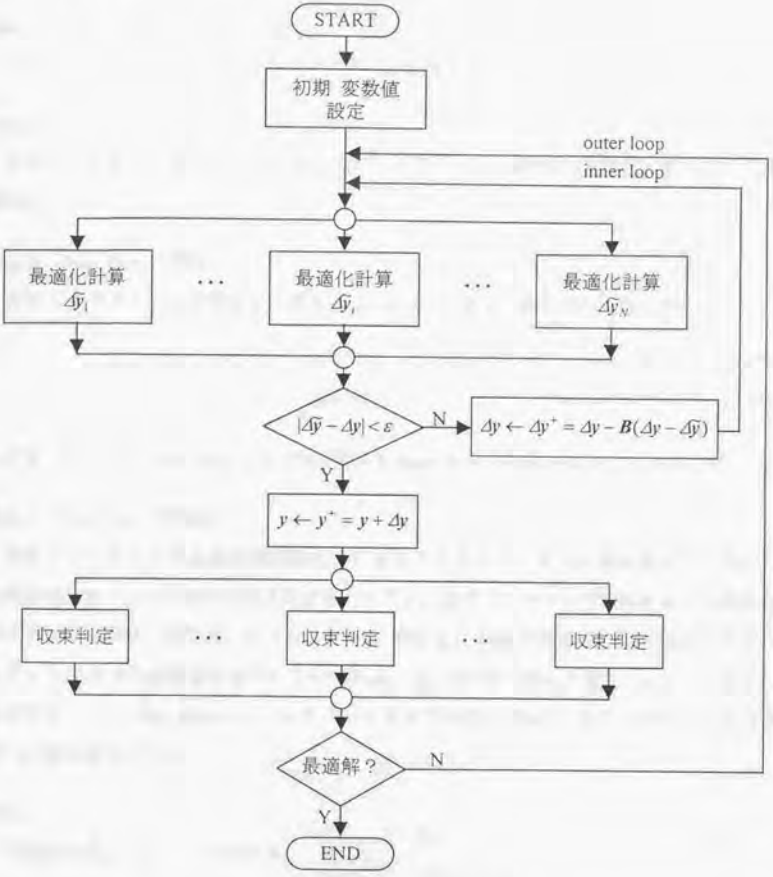


図2.2 並列最適化アルゴリズム

並列最適化法は部分最適化問題を解くサブシステムレベルのプロセスと，それらの解をまとめて変数値を更新するシステムレベルのプロセスに分けられる．したがって，サブシステムが N 個であれば，N 個のサブシステムプロセスと 1 個のシステムプロセスからなる合計 $N+1$ 個のプロセスが同時に存在する．

並列最適化法のアルゴリズムは図2.2に示す通りである．簡単にまとめる．

Step 1.

システムプロセスとサブシステムプロセスを開始する．

Step 2.

各サブシステムの変数を入力する．なお，ラグランジュ乗数の初期値は **0** としても問題ない．

Step 3.（outer loop の開始）

変数 $\boldsymbol{x}_i$ とラグランジュ乗数 μ_i の更新量 $\Delta\boldsymbol{x}_i$，$\Delta\mu_i$ に適当な値を代入する．例えば，

$$\Delta\boldsymbol{x}_i = \boldsymbol{0} \tag{2.54a}$$

$$\Delta\mu_i = 0 \tag{2.54b}$$

とする．始めての outer loop でなければ前回の outer loop の結果を使うことができる．

Step 4.（inner loop の開始）

各サブシステムの部分最適化問題に合わせてシステムプロセスは変数値とラグランジュ乗数値をサブシステムプロセスに送る．ただし，各サブシステムで最適化される変数とラグランジュ乗数，例えば，サブシステム i では $\boldsymbol{x}_i$，μ_i はそのまま転送するが，サブシステム i に含まれる最適化を行わない変数 $\boldsymbol{x}_j$，μ_j（$j \neq i$）は $\boldsymbol{x}_j + \Delta\boldsymbol{x}_j$，$\mu_j + \Delta\mu_j$ として転送する．これ以降，Step 8 まではサブシステムプロセスであり，サブシステムごとに並列に計算が進められる．

Step 5.

各関数の値 f，$\boldsymbol{h}_{ij}$，$\boldsymbol{h}_{ji}$ を求める．

Step 6.

もし outer loop の後，始めての inner loop であるならば，関数値 g_{E_i}，g_{I_i} と勾配値 $\nabla_{x_i} f$，$\nabla_{x_i} g_{E_i}$，$\nabla_{x_i} g_{I_i}$，$\nabla_{x_i} \boldsymbol{h}_{ij}$，$\nabla_{x_i} \boldsymbol{h}_{ji}$ を求める．2 回目以降の inner loop であるならば，これらの値は変わらないので次の Step 7 へ進む．値が変化しない理由は，inner loop では各サブシステムで最適化を行う変数の値（例えば，サブシステム i では $\boldsymbol{x}_i$）は変化せず，かつ接続関数が式(2.5)の形式をしているためである．

Step 7.

各サブシステムの部分最適化問題（式(2.42a)～(2.42e)）を解いて，変数と接続条件のラグランジュ乗数の更新量 $\Delta\tilde{x}_i$，$\Delta\tilde{\mu}_i$ を求める．実際には式(2.41a)～(2.41e)の QP 問題を解き，その後に 1 次元探索を行う．QP 問題の解法として Goldfarb-Indnani（GI）法（補遺A）を用いると良い．この解法は繰り返し計算によって QP 問題の解を求めるが，その初期解を求める行列計算は outer loop 後の始めての inner loop で 1 回計算をしておけば，以降 inner loop のたびに同じ計算を省略することができる．この行列計算は GI 法の計算時間の大きな割合を占めているため，2 回目以降の inner loop はかなり速く処理が進む．これは，全体最適化法に比べて並列最適化法の繰り返し計算の回数は多いが，全計算時間が短いことの理由の 1 つとなっている．また同時に式(2.49)の行列 $I-H$ の成分である $[\nabla_{y_i y_i} L]^{-1}[\nabla_{y_i y_j} L]$（$j=1,2,\ldots,N$、$j\neq i$）を求めておく．GI 法によれば，この行列は更新量 $\Delta\tilde{x}_i$，$\Delta\tilde{\mu}_i$ を求めるのと同時に得ることができる．

Step 8.

各サブシステムはシステムプロセスに対して，更新量 $\Delta\tilde{x}_i$ と $\Delta\tilde{\mu}_i$，行列 $[\nabla_{y_i y_i} L]^{-1}[\nabla_{y_i y_j} L]$ の成分を転送する．

Step 9.（inner loop の終点）

各サブシステムで計算された全ての更新量 $\Delta\tilde{x}_i$，$\Delta\tilde{\mu}_i$ が Δx_i，$\Delta\mu_i$ にほぼ等しければ，inner loop は収束したと判断して，変数値とラグランジュ乗数値を新しい値

$$x^+ = x + \Delta x \tag{2.55a}$$

$$\mu^+ = \mu + \Delta\mu \tag{2.55b}$$

に更新して Step 10 へ進む．図2.2にあるパラメータ ε は inner loop の収束を判断する閾値である．また，収束していなければ，式(2.49)の行列 $I-H$ を構成し，その逆行列を計算して行列 B とし，式(2.47)より更新量 Δx^+ と $\Delta\mu^+$ を求め，それを新たな Δx，$\Delta\mu$ にして Step 4 に戻る．

Step 10.（outer loop の終点）

各サブシステムごとに最適性の判定を並列的に行う．それらをまとめて，最適解が得られたと判断するならば計算を停止する．さもなければ Step 3 へ戻る．

2.7 軌道最適化問題に対する並列最適化の適用法

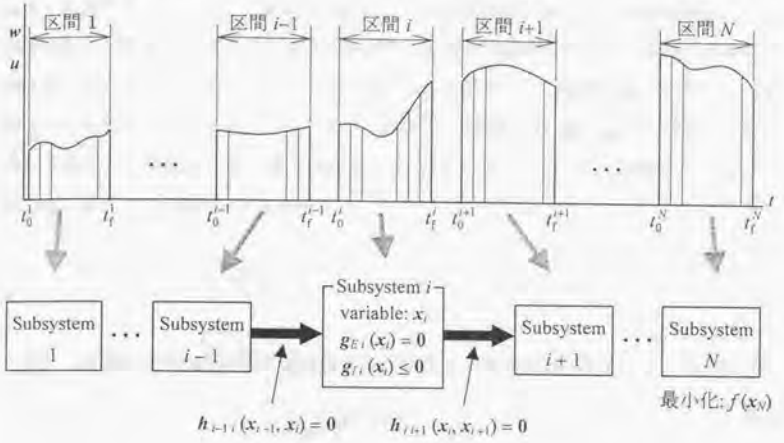


図2.3 最適制御問題の分割化

本節では軌道最適化問題に並列最適化法を適用する方法を考える．軌道最適化問題とはいわゆる最適制御問題である．最適制御問題は時間の関数である状態変数と制御変数を最適化するため，静的な変数を最適化する本章の非線形計画問題と呼ばれている最適化問題とは異なる．ここでは時間の関数である動的変数を離散化して，最適制御問題を非線形計画問題へ変換する collocation 法（補遺C）を考える．この方法の利点は様々な形式の拘束条件を扱うことができる点にある．しかし，状態変数，制御変数の数が多く，また制御時間が長い最適制御問題（大規模最適制御問題）は，非線形計画問題に変換されたとき，大規模最適化問題となる欠点があった．そこで本章で提案した並列最適化法を適用して大規模最適制御問題を解くことにする．

制御時間を N 個の区間に分け 1 つの区間を 1 つのサブシステムと見なす．このとき，サブシステム1は初期時間を含む区間からなるサブシステムとし，以降，時間の進みにあわせてサブシステムの番号を付け，終端時間を含むサブシステムをサブシステム N とする．よって，図2.3のように最適制御問題はサブシステムが train 構造で接続した形である．なお，ここでいう区間は状態方程式，拘束条件の不連続性によって必然的に生じる他に，長い制御時間を意図的に分割して構成することもできる．

ある i 番目の区間（区間 i）の時間 $t \in [t_0^i, t_f^i]$ と，最適化される状態変数 $w(t)$，制御変数 $u(t)$ を定義する．その上で，各区間において，微分方程式で表される状態方程式または運動方程式と，等式拘束条件，不等式拘束条件，区間の初期時間での初期条件，終端時間での終端条件が与えられる．ここで，collocation 法の特徴として柔軟に様々な形式の拘束条件を与えられることが挙げられる．次に，区間 i の時間を M_i 個の要素に分割し，要素間の点（節点）の時間で，状態量，制御量を w_k^i，u_k^i（$k = 0,1,\ldots,M_i$）と離散化する．これにより，区間 i で最適化される変数 x_i は

$$x_i \equiv \left({w_0^i}^{\mathrm{T}}, {u_0^i}^{\mathrm{T}}, \ldots, {w_{M_i}^i}^{\mathrm{T}}, {u_{M_i}^i}^{\mathrm{T}}, t_0^i, t_f^i \right)^{\mathrm{T}} \tag{2.56}$$

となり，各種拘束条件も同様に離散化され，等式と不等式の制約条件にまとめられる．

$$g_{Ei}(x_i) = 0 \tag{2.57a}$$

$$g_{Ii}(x_i) \le 0 \tag{2.57b}$$

加えて，ある区間 i の初期時間 t_0^i とその時間の状態量 w_0^i と制御量 u_0^i は 1 つ前の区間 $i-1$ における終端時間 t_f^{i-1} とその時間の状態量 $w_{M_{i-1}}^{i-1}$ と制御量 $u_{M_{i-1}}^{i-1}$ に等しくなければならない．これが隣り合うサブシステム間に存在する接続条件

$$h_{i-1,i}(x_{i-1}, x_i) \equiv \begin{bmatrix} w_{M_{i-1}}^{i-1} - w_0^i \\ u_{M_{i-1}}^{i-1} - u_0^i \\ t_f^{i-1} - t_0^i \end{bmatrix} = 0 \quad (i = 2,3,\ldots,N) \tag{2.58}$$

になる．ただし，問題によっては区間が変わる時間で制御量が不連続に変化しても良い場合がある．その場合，式(2.58)から制御量の接続条件を除く．

また，最適制御問題の評価関数は終端時間の状態で表されることが多い．すなわち，

$$\text{minimize} \quad f(w_{M_N}^N, u_{M_N}^N, t_f^N) \tag{2.59}$$

となる場合がある．このとき，最適化問題の目的関数は区間 N の変数 x_N の関数として表され，N 番目以外の区間では目的関数が定義できないことになる．

よって，本論文で提案する並列最適化法を最適制御問題に適用し，各サブシステムの部分最適化問題を以下のように定義する．

サブシステム1（区間 1）

$$\text{variable} \quad x_1 \tag{2.60a}$$

$$\text{minimize} \quad \mu_{I2}{}^{\mathrm{T}} h_{I2}(x_1, x_2) \tag{2.60b}$$

$$\text{subject to} \quad g_{E1}(x_1) = 0 \tag{2.60c}$$

$$g_{I1}(x_1) \le 0 \tag{2.60d}$$

サブシステム i（区間 i）　$(i = 2, \ldots, N-1)$

$$\text{variable} \quad x_i \tag{2.61a}$$

$$\text{minimize} \quad \mu_{ii+1}{}^{\mathrm{T}} h_{ii+1}(x_i, x_{i+1}) \tag{2.61b}$$

$$\text{subject to} \quad g_{Ei}(x_i) = 0 \tag{2.61c}$$

$$g_{Ii}(x_i) \le 0 \tag{2.61d}$$

$$h_{i-1i}(x_{i-1}, x_i) = 0 \tag{2.61e}$$

サブシステム N（区間 N）

$$\text{variable} \quad x_N \tag{2.62a}$$

$$\text{minimize} \quad f(x_N) \tag{2.62b}$$

$$\text{subject to} \quad g_{EN}(x_N) = 0 \tag{2.62c}$$

$$g_{IN}(x_N) \le 0 \tag{2.62d}$$

$$h_{N-1N}(x_{N-1}, x_N) = 0 \tag{2.62e}$$

接続関数を目的関数と制約条件に巧妙に振り分けることで，全てのサブシステムの部分最適化問題で目的関数が定義されるようにしている．

より詳細な collocation 法とその解法については補遺Cを参照されたい．

2.8 まとめ

本章では本論文で提案する並列最適化法の導出とアルゴリズムの説明を行った．

まず，大規模システムと大規模最適化問題の定義を行い，大規模システムをカップリングした複数のサブシステムに分割することを述べた．次に大規模最適化問題を一括して解く全体最適化法について説明した．ここで取り上げた最適化法は逐次 2 次計画（SQP）法である．SQP 法は現在最も有力な最適化手法であるが大規模最適化問題になると計算負荷が大きくなる．そこで，SQP 法にとって最も計算時間が要する大規模線形方程式の計算過程を並列処理することにした．これは第 1 章で述べた「解法の並列化」に基づく並列最適化法の導出である．しかし，この解法で並列に行われている個々の計算を見直してみると，ある最適化計算を行っていることに等しい．その最適化問題を部分最適化問題と呼ぶ．すなわち，1 つの大規模システムの大規模最適化問題を，工学的意味をもつサブシステムごとの部分最適化問題に分割して並列計算することができる．よって，本論文において並列最適化法と名付けた解法は大規模最適化問題を複数の部分最適化問題に分割して解く分割解法でありながら「解法の並列化」に等しい並列化効率をもつことが期待でき，次章からその確認が行われる．本章の最後に次章からの例題への適用に備え，軌道最適化問題に並列最適化法を適用する場合の定式化を行った．

第3章

軌道最適化問題に対する並列最適化法の適用例

本章では前章で示した並列最適化法を簡単な 2 つの軌道最適化問題に適用し，その有効性を確認する．まず並列計算の評価項目について簡単にまとめ，その後に例題に並列最適化法を適用する．例題は「不連続な潮流の中を進む船」「最速降下線問題」と呼ばれる問題である．前者の問題では，変数の数が少ない小規模の最適化問題から変数の数が多い最適化問題まで問題の規模を変化させることができ，変数の数が変化する大規模最適化問題を，部分最適化問題の変数の数が一定になるように適当な数に分割する．一方，後者の問題における全体問題の変数の数は常に一定であり，これを様々な数の部分問題に分割する．よって，前者と異なり後者の部分問題の変数の数は分割数によって変化する．

ところで，本論文で提案する並列最適化法は並列計算アルゴリズムの 1 つではあるが，一般的な並列計算に比べれば並列度（分割数）は小さく，本章の例題でもその数は 10 以下である．よって，提案された並列最適化法の計算のために専用の並列計算機を使用するまでもなく，ネットワーク接続されたパーソナルコンピュータで十分計算可能である．部分最適化問題を各計算機に割り当てることによって，1 台の計算機を 1 つのサブシステムとみなして解析と最適化を行うことができる．ただし，本章の例題に対しては 1 台の計算機上で複数のプロセスを起動し，各プロセスを切り替えながら実行することによって並列計算を模擬した．

3.1 並列最適化の評価法

一般的な並列アルゴリズムの評価方法[9]を参考にして，本論文の並列最適化法に対して考慮する評価項目を列挙してまとめておく．

3.1.1 評価項目

(1) 分割数：N

並列最適化における全体最適化問題の分割数（部分最適化問題の数）を N と表す．実際の並列計算に必要なプロセッサ数はサブシステムプロセスの N 個にシステムプロセスを合わせて $N+1$ 個になる．

(2) 実行時間：$T(N)$

分割数 N の問題を実行したときの計算開始から結果が得られるまでの計算時間を $T(N)$ と表す．並列計算が行われているサブシステムプロセスの計算時間のうち最も長く実行時間を要したプロセスの計算時間をシステムプロセスの計算時間に加えていく．

(3) 実行速度向上率：$T_R(N)$

全体最適化法での実行時間を $T(1)$ とする．実行速度向上率 $T_R(N)$ とは全体最適化法での実行時間 $T(1)$ と並列最適化法での実行時間 $T(N)$ の比で表される．

$$T_R(N)=\frac{T(1)}{T(N)} \tag{3.1}$$

(4) 実行速度効率：$T_E(N)$

並列最適化法の実行時間 $T(N)$ に分割数 N を掛けた値をコストと呼ぶ．実行速度効率とは全体最適化法の実行時間と並列最適化法のコストの比で表される．

$$T_E(N)=\frac{T(1)}{T(N)\times N}=\frac{T_R(N)}{N} \tag{3.2}$$

(5) メモリ量：$M(N)$

並列最適化を実行するプログラム及びデータ格納に必要な 1 プロセス当たりの最大メモリ量を $M(N)$ とする．

(6) メモリ量比：$M_R(N)$

メモリ量比 $M_R(N)$ とは全体最適化法のメモリ量 $M(1)$ と並列最適化法のメモリ量 $M(N)$

の比を表す．

$$M_R(N)=\frac{M(1)}{M(N)} \tag{3.3}$$

(7) メモリ量効率：$M_E(N)$

実行速度効率$T_E(N)$と同様に，メモリ量についてもメモリ量効率が定義される．

$$M_E(N)=\frac{M(1)}{M(N)\times N}=\frac{M_R(N)}{N} \tag{3.4}$$

3.1.2 並列化効率の限界

一般的に，逐次アルゴリズム（例えば N 回のループが存在するアルゴリズム）を，そのまま N 個に分割して並列に実行すると，実行速度向上率は$T_R(N)<N$となり，実行速度効率は$T_E(N)<1$になる．よって，並列アルゴリズムには$T_R(N)>N$，$T_E(N)>1$を実現することが求められる．すなわち，全体問題を N 個の部分問題に分けたとき，計算時間が $1/N$ 未満になることが要求される．ただし，分割数を多くすればするほど実行時間が短くなり速度向上率が増加し続けるわけではなく必ず限界が存在する．なぜなら，サブシステム間（プロセッサ間）でデータが転送される過程や，本論文の並列最適化法では式(2.47)においてΔy^+を計算する過程など，分割できない部分が必ず存在するからである．例えばあるアルゴリズムにおいて並列化できない部分の割合を$x(\le 1)$とする．このアルゴリズムをNプロセッサで並列処理すると，大雑把に，実行時間$T(N)$，実行速度向上率$T_R(N)$は

$$T(N)=\frac{(1-x)T(1)}{N}+xT(1) \tag{3.5a}$$

$$T_R(N)=\frac{N}{Nx+1-x} \tag{3.5b}$$

と求まる．よって，プロセッサ数を無限に増やしても，

$$\lim_{N\to\infty} T(N)=xT(1) \tag{3.6a}$$

$$\lim_{N\to\infty} T_R(N)=\frac{1}{x} \tag{3.6b}$$

となり，並列化できない部分の影響を受けて，実行時間，実行速度向上率には限界が存在する．これはアムダールの法則（Amdahl's law）[28]といわれている．また，メモリ量，メモリ量比にも同じく限界が存在する．

3.2 例題1「不連続な潮流の中を進む船」

3.2.1 問題設定

図3.1のように不連続な潮流がある海域を船が目標地点まで最短時間で航行する航路を求める[29]．ここで船は質点とし，時間$t \in [t_0, t_f]$に対し，状態変数は座標 x，y の 2 個，制御量は経路角γとする．運動方程式を次のように定義する．

$$\dot{x} = V\cos\gamma + u \tag{3.7a}$$

$$\dot{y} = V\sin\gamma \tag{3.7b}$$

ここで，船の速さ$V=1$とし，uは潮流の大きさを表す．船は初期条件

$$x(t_0)=0,\quad y(t_0)=0,\quad t_0=0 \tag{3.8}$$

から，終端条件

$$x(t_f)=N,\quad y(t_f)=N \qquad (N\text{は自然数}) \tag{3.9}$$

まで航行する．この間の状態量空間をy軸に沿ってN個に分割する．その中の領域iは

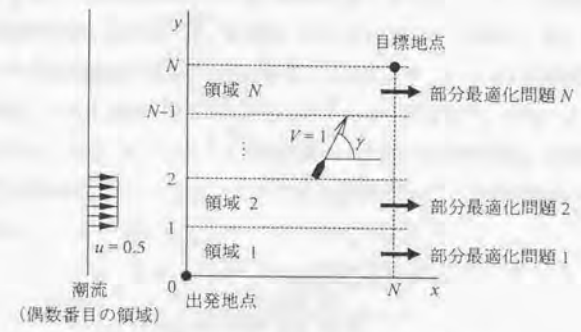


図3.1 不連続な潮流を受ける船の最短時間経路

$$\{(x,y)\,|\,x\in R,\ y\in[i-1,i]\} \tag{3.10}$$

で示される．この領域のうち，偶数番目の領域にのみ潮流があるとする．すなわち，

$$u=0 \quad (\text{領域 } 1,3,4,\ldots) \tag{3.11a}$$

$$u=0.5 \quad (\text{領域 } 2,4,6,\ldots) \tag{3.11b}$$

とする．以上の拘束条件のもとで，出発地点から目標地点まで最短時間でたどり着ける経路を求めることにする．よって最小化する評価関数 J は

$$J=t_f \tag{3.12}$$

である．

軌道最適化問題の解法に BDH 法（補遺C）を用いる．出発地点から目的地点までの状態量空間を運動方程式が異なる N 個の領域に分けたことから，初期時間 $t=t_0$ より終端時間 $t=t_f$ までの制御時間を同じ N 個の区間に分ける．そして，ある区間 i の初期時間を t_0^i，終端時間を t_f^i とすると，これまでの拘束条件に，各区間の終端条件

$$y(t_f^i)=i \tag{3.13}$$

を加え，N 個の区間からなる軌道最適化問題に対して BDH 法を適用する．このとき，各区間の時間を 30 個の等幅要素で離散化した．

軌道最適化問題に対する並列最適化法の適用は2.7節に従い，各区間をサブシステムと見なし，N 個の部分最適化問題をもつ並列最適化法を考える．よって，区間数が増えて全体最適化問題の変数の数が増すが，区間数と同じ数の分割数で問題を分割するため変数の数が常に一定な部分最適化問題が構成される．これにより，いかなる区間数の問題であっても実行時間，メモリ量は一定に保たれるはずである．あるサブシステム i と $i+1$ の間にある接続条件としては，サブシステム i の変数の一部である終端時間 t_f^i，終端状態量 $x(t_f^i)$，$y(t_f^i)$ が，サブシステム $i+1$ の変数の一部である初期時間 t_0^{i+1}，初期状態量 $x(t_0^{i+1})$，$y(t_0^{i+1})$ と等しくなる

$$\boldsymbol{h}_{i+1}\equiv\begin{bmatrix} x(t_f^i)-x(t_0^{i+1}) \\ y(t_f^i)-y(t_0^{i+1}) \\ t_f^i-t_0^{i+1} \end{bmatrix}=\boldsymbol{0} \tag{3.14}$$

という条件が定義される．ここで，接続条件には制御量を含めないため制御量は連続しない可能性があることに注意する．問題の意味を考えれば制御量が不連続に変化することは明らかである．

最適化を行うに当たり，初期解として

$$x(t)=Vt,\quad y(t)=Vt,\quad \theta(t)=45\,[\mathrm{deg}] \tag{3.15a}$$

$$t_0=0,\quad t_f=\frac{\sqrt{2}N}{V} \tag{3.15b}$$

とする軌道を与える．また，全体最適化と並列最適化による繰り返し計算の停止条件は，理論解と比較した目的関数値の誤差が 0.0001 %以下になり，かつ制約条件

$$\boldsymbol{g}_E\equiv\begin{bmatrix} g_{E1}\\ \vdots\\ g_{Em_E}\end{bmatrix}=\boldsymbol{0},\quad \boldsymbol{g}_I\equiv\begin{bmatrix} g_{I1}\\ \vdots\\ g_{Im_I}\end{bmatrix}\le\boldsymbol{0} \tag{3.16}$$

の誤差の和

$$error\equiv\sum_{i=1}^{m_E}\left|g_{Ei}\right|+\sum_{i=1}^{m_I}\max\left[0,g_{Ii}\right] \tag{3.17}$$

がそれ以上計算を繰り返しても 10^{-10} を超えなくなったときとする．

本節では，分割数 N を 2 から 10 まで変化させて全体最適化と並列最適化法を適用したときに，計算負荷である計算時間と必要メモリ量がどう変化するか調べてみる．なお，計算機には Visual Technology 社製 VT-Alpha 667(CPU: Alpha 21164A 667 [MHz], OS: Windows NT 4) を一台使用する．このコンピュータは並列計算機ではない．複数のプロセスに対して CPU の処理能力を順次振り分けて並列計算を模擬する．詳細は補遺Fで説明する．

3.2.2 数値計算結果

(1) 数値解

この問題は単純な軌道最適化問題であり，容易に理論解を得ることができる[29]．図3.2, 3.3に，$N=4$（区間数 4）としたときの数値解と理論解の比較を示す．いかなる区間数（$N=2,3,\ldots,10$）の問題設定においても両者はよく一致した．

(2) 繰り返し回数

表3.1では初期解から最適解を得るまでの繰り返し回数を比較している．まず，部分問題数が増加するにつれて繰り返し回数は増加するが，初期解と最適解の差が大きくなるためと思われる．並列最適化法には inner loop と outer loop の 2 つの繰り返し回数が示されている．前章で説明した通り，並列最適化法の outer loop における計算は全体最適化法の繰り返し計算とほぼ同じであるため，両者の繰り返し回数はほぼ一致した．また，この表によると，inner loop の回数は outer loop の回数の 2 倍になっている．ここでの問題は不等式制約条件が存在しないため，式(2.47)の行列 $\boldsymbol{B}$ を式(2.48)の行列 $(\boldsymbol{I}-\boldsymbol{H})^{-1}$ とすれば，inner loop は 1 回の計算で収束して outer loop に進むことができる．しかし，ここでは inner loop の収束を確認するために，さらに 1 回の計算を繰り返しているため，outer loop 1 回当たり 2 回の inner loop の計算が行われている．

(3) 実行時間

図3.4は実行時間を示している．区間数を多くすると全体問題の規模は大きくなり，1.3 節で示した通り，全体最適化の実行時間は比例を超える関係で増大していく．それを区間数と同数の部分問題に分割して並列最適化を行うと，実行時間は繰り返し回数に合わせて多少の増加はするものの，非常に少ない時間で最適解を得ることができる．そのため，2 つの最適化法の実行時間の比である実行速度向上率（図3.5）は分割数の増加につれて全体問題の実行時間と同じように増加し，また，実行速度効率（図3.6）も分割数と共に増加する．

(4) メモリ量

図3.7，3.8，3.9はメモリ量とその向上率，効率を示している．メモリ量にはプロセスの変数を格納する領域に加え，そのプログラムの格納領域も含まれている．実行時間と同様に，全体最適化法におけるメモリ量は問題の規模が大きくなると比例を超える関係で増大することが分かる．しかし，並列最適化を行えば 1 つの部分問題当たりの問題の規模はほぼ変化しないため，必要なメモリ量を一定に抑えることができる．

表3.1 繰り返し回数

部分問題数 N	全体最適化法	並列最適化法	
		inner loop	outer loop
2	24	48	24
3	24	48	24
4	30	60	30
5	32	54	27
6	32	64	32
7	42	84	42
8	52	104	52
9	54	108	54
10	37	76	38

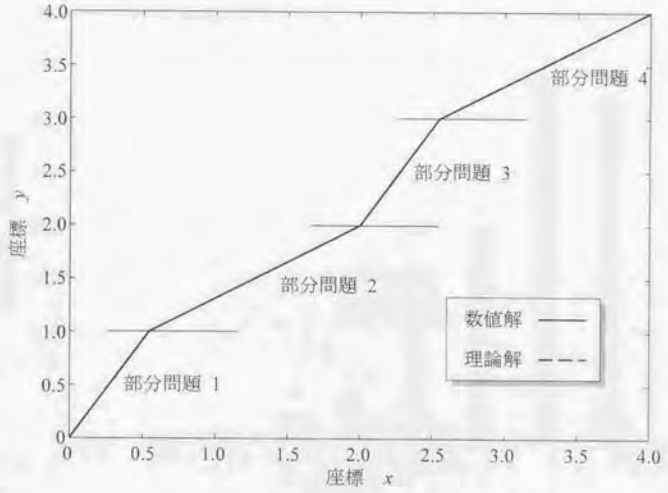


図3.2 軌道（部分最適化問題数4）

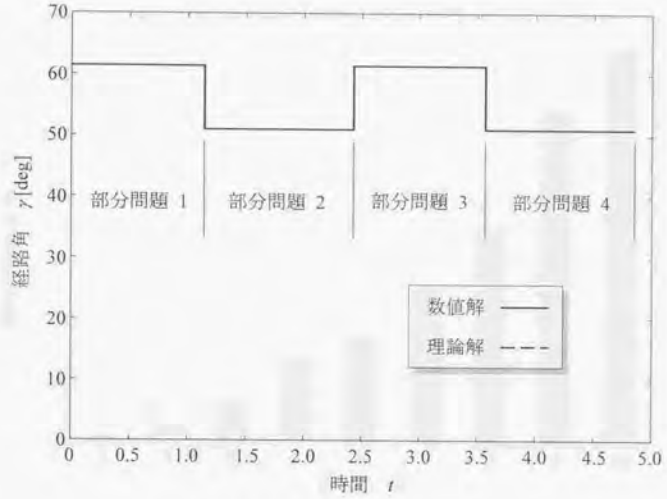


図3.3 制御量（部分最適化問題数4）

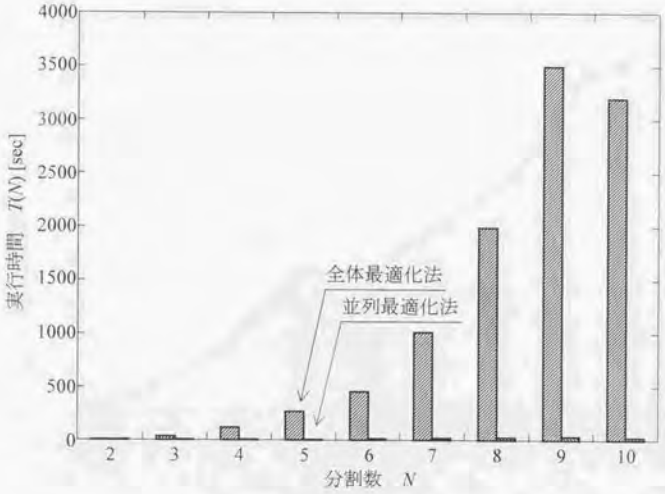


図3.4 実行時間

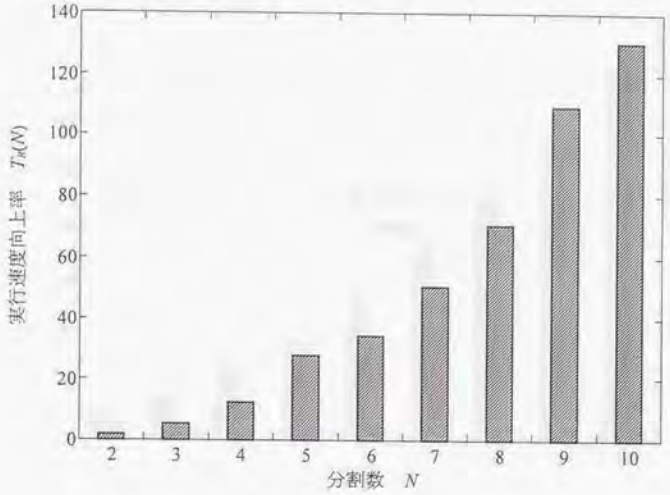


図3.5 実行速度向上率

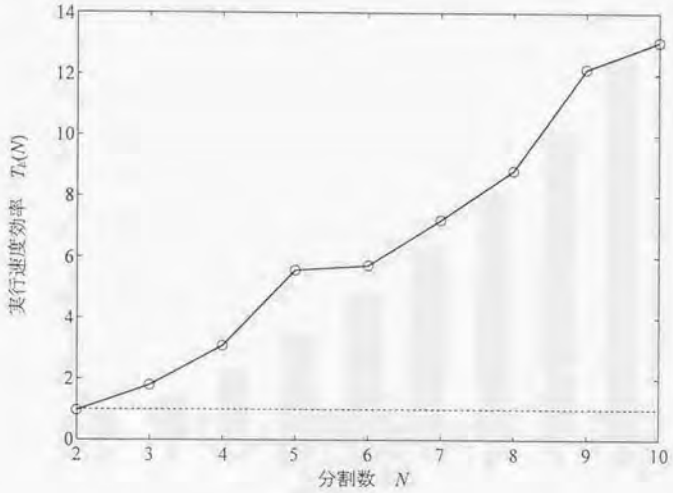


図3.6 実行速度効率

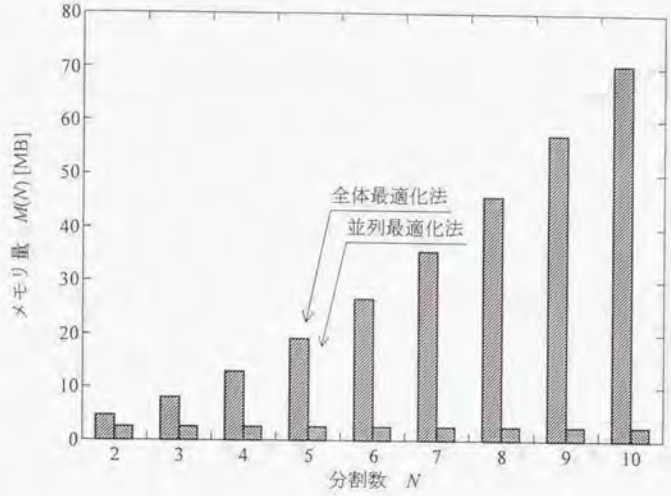


図3.7 メモリ量

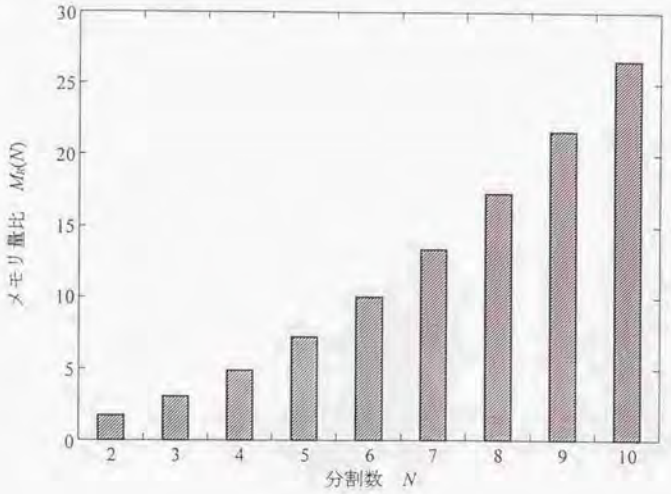


図3.8 メモリ量比

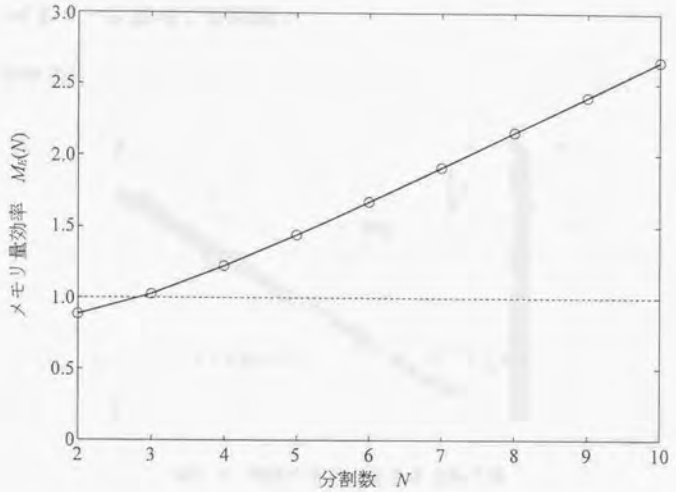


図3.9 メモリ量効率

3.3 例題2「最速降下線問題」

3.3.1 問題設定

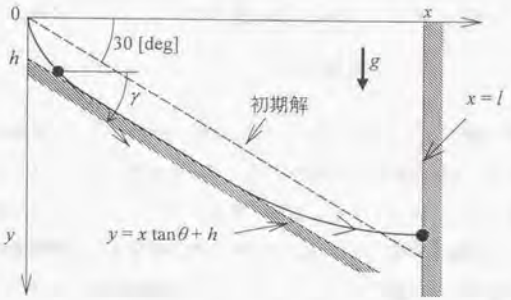


図3.10 拘束のある場合の最速降下線

変分法の問題として有名な最速降下線問題[29]を取り上げる．

原点で静止している質点が一様な重力を受けて，鉛直方向に伸びる直線上まで鉛直面内を落下する時の最短時間経路を求める．時間 $t \in [t_0, t_f]$ に対し，状態変数は質点の位置を表す座標 x, y の 2 つ，制御変数は経路角を表す γ である．運動方程式は次のように定義される．

$$\dot{x} = \sqrt{2gy}\cos\gamma \tag{3.18a}$$

$$\dot{y} = \sqrt{2gy}\sin\gamma \tag{3.18b}$$

初期条件，

$$x(t_0) = 0, \quad y(t_0) = 0, \quad t_0 = 0 \tag{3.19}$$

終端条件，

$$x(t_f) = l \tag{3.20}$$

として，最小化する評価関数 J は

$$J = t_f \tag{3.21}$$

と定義する．また，軌道に状態量不等式拘束条件

$$y \le x\tan\theta + h \tag{3.22}$$

が存在しない場合と存在する場合の2例を調べる．なお，パラメータ値は以下に定める．

$$g=1,\quad l=1,\quad h=0.1,\quad \theta=30\,[\mathrm{deg}] \tag{3.23}$$

この軌道最適化問題にも先の例題と同様にBDH法を適用し，その際に時間を200個の等幅要素で離散化した．本稿の数値計算では初期時間から終端時間までの200個の要素を等数に近い要素数をもった2から10の区間に分割してサブシステムとし部分最適化問題を構成した．先の問題設定では変数の数の増加に合わせて分割数も増加し，部分問題の変数の数は分割数によらず一定に保たれていた．一方，この問題では分割数が増加しても全体問題の変数の数はほぼ一定であり，分割数の増加につれて部分問題の変数の数は減少していく．したがって，一定規模の大規模最適化問題に対して，分割数によって並列最適化法の実行時間とメモリ量がどう変化するか調べる．

各区間の終端条件として，

$$x(t_f^i) = i\frac{l}{N} \qquad (N\text{は分割数}) \tag{3.24}$$

を加える．あるサブシステム i と $i+1$ の間にある接続条件として，サブシステム i の変数の一部である終端時間 t_f^i，終端状態量 $x(t_f^i)$，$y(t_f^i)$，終端制御量 $\gamma(t_f^i)$ と，サブシステム $i+1$ の変数の一部である初期時間 t_0^{i+1}，初期状態量 $x(t_0^{i+1})$，$y(t_0^{i+1})$，終端制御量 $\gamma(t_0^{i+1})$ から

$$\boldsymbol{h}_{i+1} \equiv \begin{bmatrix} x(t_f^i) - x(t_0^{i+1}) \\ y(t_f^i) - y(t_0^{i+1}) \\ \gamma(t_f^i) - \gamma(t_0^{i+1}) \\ t_f^i - t_0^{i+1} \end{bmatrix} = \boldsymbol{0} \tag{3.25}$$

という条件が定義される．ここでは先の例題と異なり制御量の接続条件が含まれる．

初期解には30 [deg]の経路角で落下する軌道を与える．すなわち，ある部分問題 i の初期解は次のようになる．

$$x(t)=\frac{\sqrt{3}}{8}gt^2,\quad y(t)=\frac{1}{8}gt^2,\quad \gamma(t)=30\,[\mathrm{deg}] \tag{3.26a}$$

$$t\in\left[\ 2\sqrt{\frac{2(i-1)l}{\sqrt{3}gN}},\quad 2\sqrt{\frac{2il}{\sqrt{3}gN}}\ \right] \tag{3.26b}$$

また，全体最適化と並列最適化による繰り返し計算の停止条件は，理論解と比較した目的関数値の誤差が 0.02 %以下になり，制約条件の誤差の和がそれ以上計算を繰り返しても 10^{-5} を超えなくなったときとする．

3.3.2 数値計算結果

(1) 数値解

この問題に対する理論解は容易に得ることができる[29]．図3.11～3.14に，部分問題数 $N=5$ としたときの数値解と理論解の比較を示す．不等式拘束条件の有無，部分問題数に関わらず，数値解は理論解とよく一致した．図3.11，3.12に見られる終端時間近くでのわずかな両最適解の軌道のずれは並列最適化法によるものではなく，動的変数の離散化に起因する．

(2) 繰り返し回数

表3.2，3.3は不等式拘束条件がない場合，ある場合について，初期解から最適解を得るまでの繰り返し回数を比較した表である．部分問題数が増加するにつれて繰り返し回数が減少する原因は，区間の終端に置かれた制約条件（式(3.24)）が増えて解が制約を満たすようにすばやく収束するためと思われる．全体最適化法における繰り返し回数は不等式拘束条件がある場合のほうが少ない理由も制約条件の増加による．また，前節の例題と同様に，並列最適化法の outer loop における繰り返し回数は全体最適化法の繰り返し回数とほぼ同じである．不等式拘束条件がない場合の inner loop の回数は outer loop の回数の 2 倍になっている結果も先の例題と同じである．しかし，不等式拘束条件があると，活性な不等式制約条件が定まるまで式(2.47)の行列 $\boldsymbol{B}$ を正確に求めることができないため，inner loop をより多く繰り返すことになる．なかなか収束しないとき，2.5.3項で説明した通り，式(2.47)のパラメータ κ を徐々に減少させる．どうしても収束しないときはそのまま outer loop に進むことも説明した．inner loop に多くの反復回数が必要なのは，活性となる不等式制約条件が不明な最適化計算開始時である．計算が進むにつれ解が最適解に近付けば活性な不等式制約条件も決まり，inner loop は少ない回数で収束するようになる．inner loop はおよ

そ2〜30回，平均5回程度で収束する．

(3) 実行時間

図3.15〜3.20は実行時間とその向上率，効率を示している．前節の問題では全体最適化問題の変数の数に合わせて分割数も増加していたので，全体最適化法の実行時間は分割数と共に増加した．しかし，本節の問題設定では全体最適化問題の変数の数はほぼ一定であり，分割数が増えても区間の終端条件と接続条件が増えるだけであるので，繰り返し回数の増減に比例した変化が見られるだけである．前節との分割に対する考え方が異なるため，分割数に対する傾向の違いは実行速度向上率，実行速度効率にも見られる．

分割数に対する実行速度向上率の推移を見てみると，分割数を増やし部分問題の数を多くするにつれ実行速度向上率は増加するが，ある程度以上に分割数を増やすとむしろ減少し始める．これは分割数の増加により部分問題間のデータの転送とシステムプロセスの作業量が増加したためである．同様に，実行速度効率も不等式拘束条件がない場合には分割数 4〜6，不等式拘束条件がある場合には分割数 3 辺りで最大値をとり，それ以上分割数を増やすと減少する．多くの分割数によって並列計算を行っても効率が上がらないことは3.1.2項で説明したアムダールの法則に一致する．一方，不等式拘束条件の有無による実行速度効率の違いを比較してみると，不等式拘束条件があると inner loop に多くの繰り返し回数が必要となるため，不等式拘束条件がない場合の効率よりは悪くなる．しかし，どちらの場合も実行速度効率は1を大きく上回っている．

(4) メモリ量

図3.21〜3.26にメモリ量に関する結果を示す．前節との分割に対する考え方が異なるため，分割数に対する傾向の違いがこれらにも見られる．ここでは，必要メモリ量は分割を増やせば一様に減少する．しかし，メモリ量効率には限界があり，分割数4を最大値としてそれ以上分割数を増やすと効率は悪くなる．これはプログラムのためのメモリ領域等，分割することができないメモリ量を含むからである．

表3.2 繰り返し回数（不等式拘束なし）

部分問題数 N	全体最適化法	並列最適化法	
		inner loop	outer loop
2	450	900	450
3	409	818	409
4	281	562	281
5	250	500	250
6	261	522	261
7	261	522	261
8	246	492	246
9	247	494	247
10	281	562	281

表3.3 繰り返し回数（不等式拘束あり）

部分問題数 N	全体最適化法	並列最適化法	
		inner loop	outer loop
2	397	1360	391
3	338	1372	336
4	290	1215	291
5	232	1185	234
6	234	1575	234
7	236	1765	236
8	228	1592	228
9	234	1614	235
10	221	1603	221

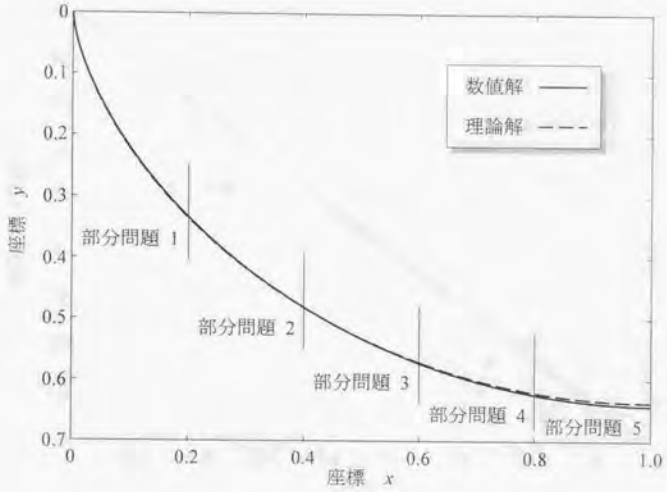


図3.11 軌道（不等式拘束なし）（部分最適化問題数 5）

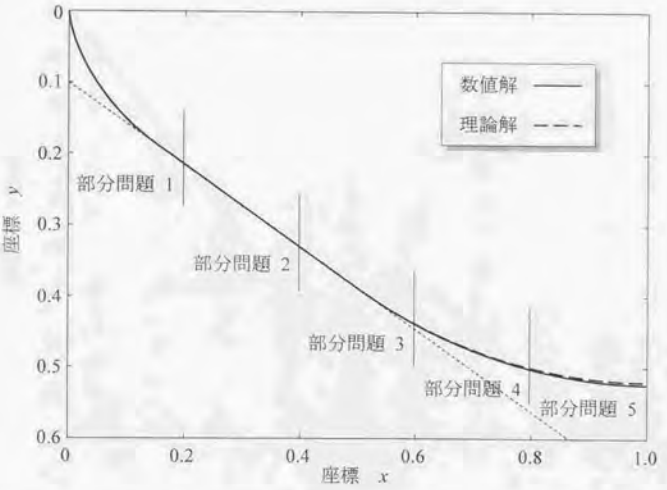


図3.12 軌道（不等式拘束あり）（部分最適化問題数 5）

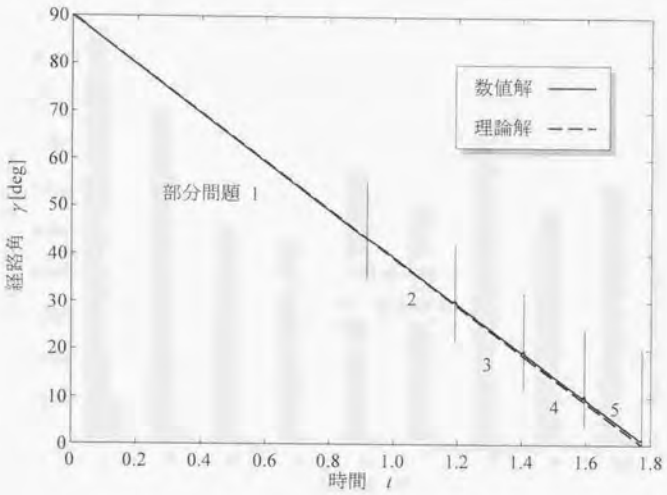


図3.13 制御量（不等式拘束なし）（部分最適化問題数 5）

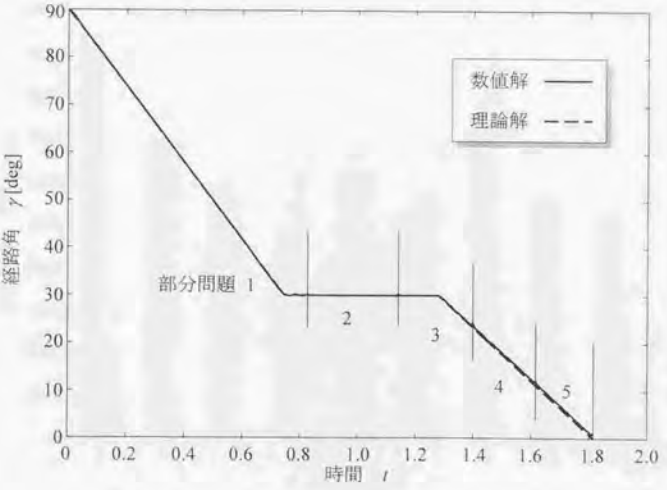


図3.14 制御量（不等式拘束あり）（部分最適化問題数 5）

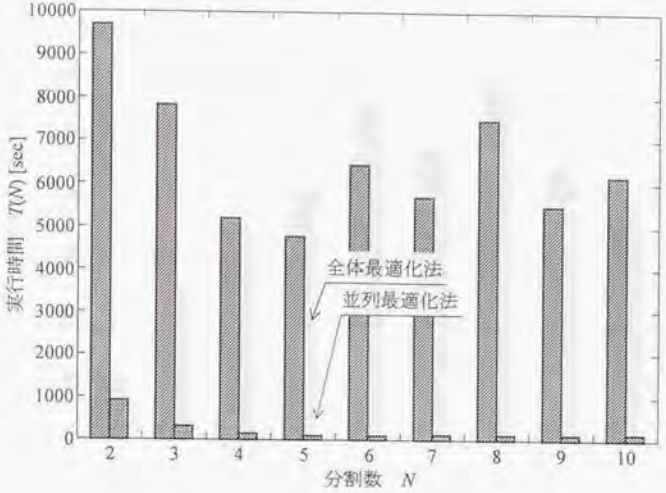


図3.15 実行時間（不等式拘束なし）

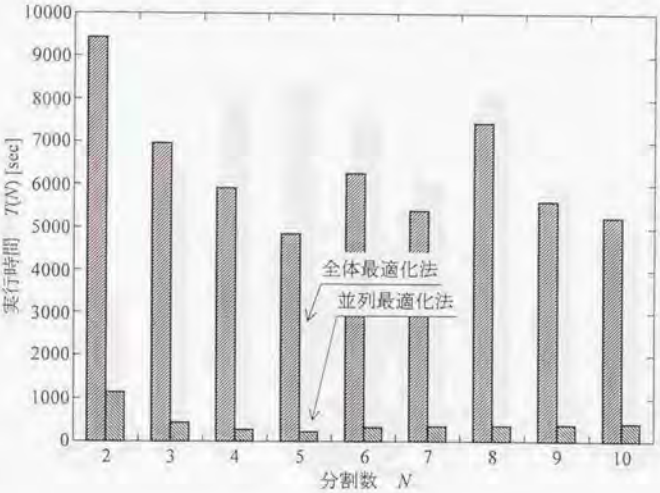


図3.16 実行時間（不等式拘束あり）

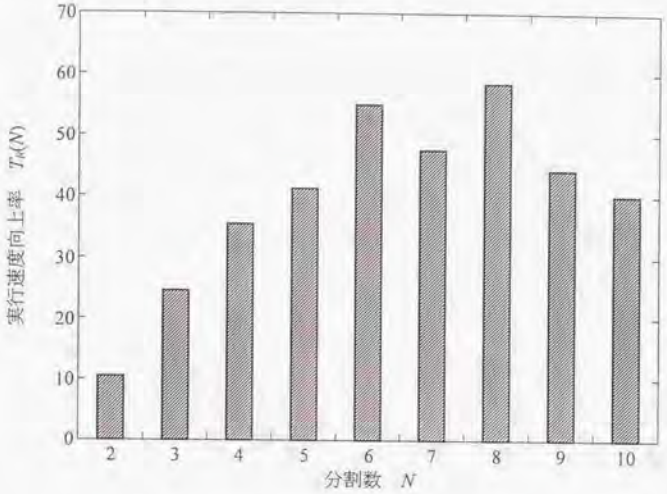


図3.17 実行速度向上率（不等式拘束なし）

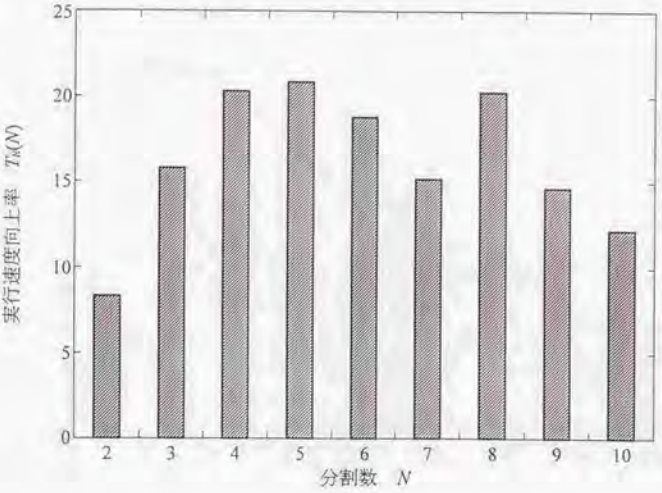


図3.18 実行速度向上率（不等式拘束あり）

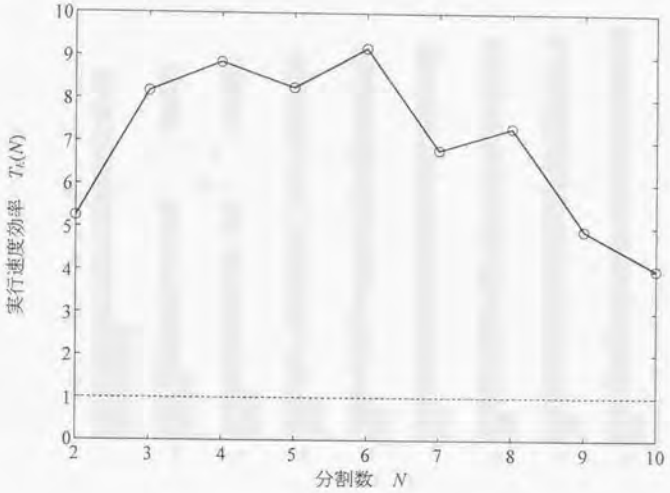


図3.19 実行速度効率（不等式拘束なし）

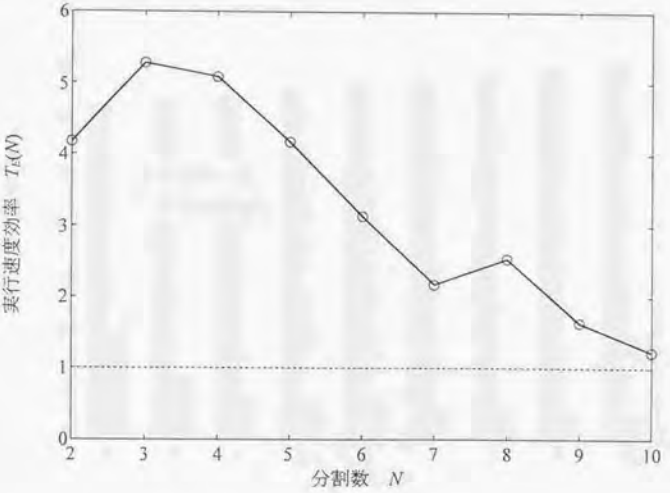


図3.20 実行速度効率（不等式拘束あり）

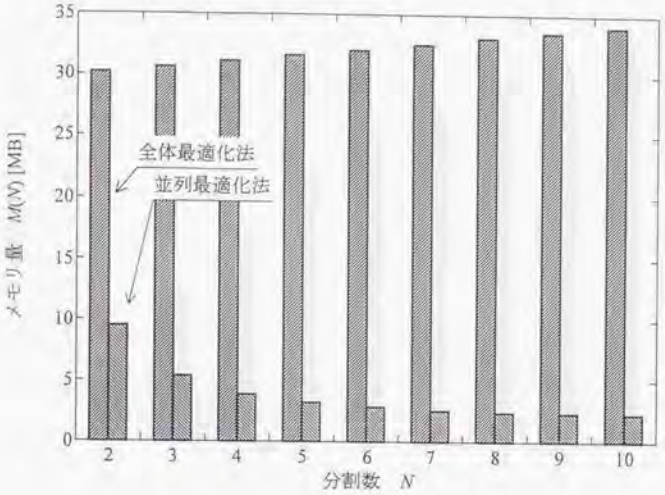


図3.21 メモリ量（不等式拘束なし）

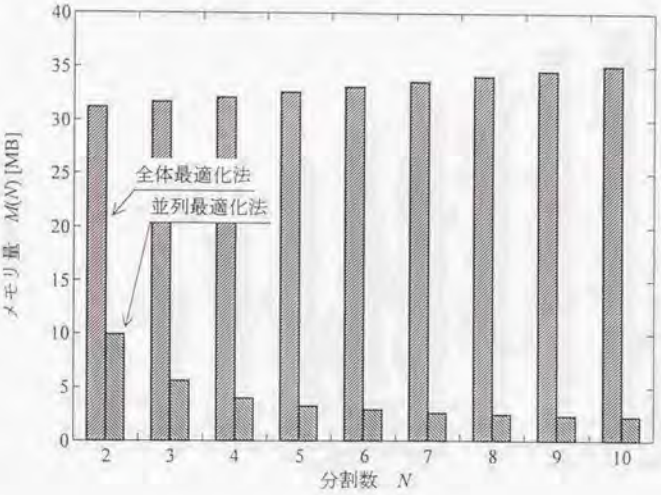


図3.22 メモリ量（不等式拘束あり）

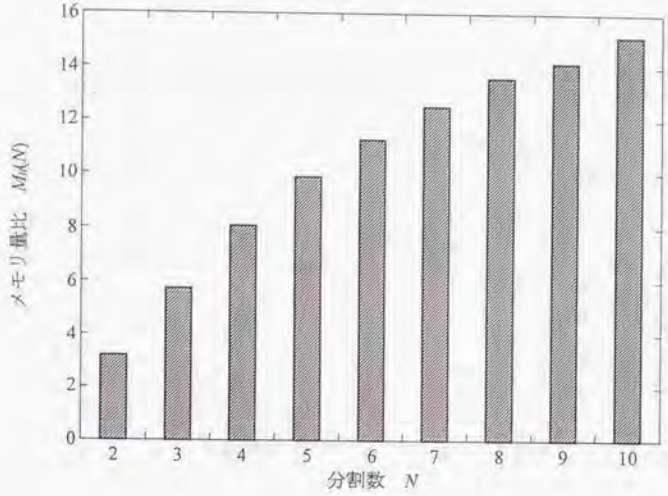


図3.23 メモリ量比（不等式拘束なし）

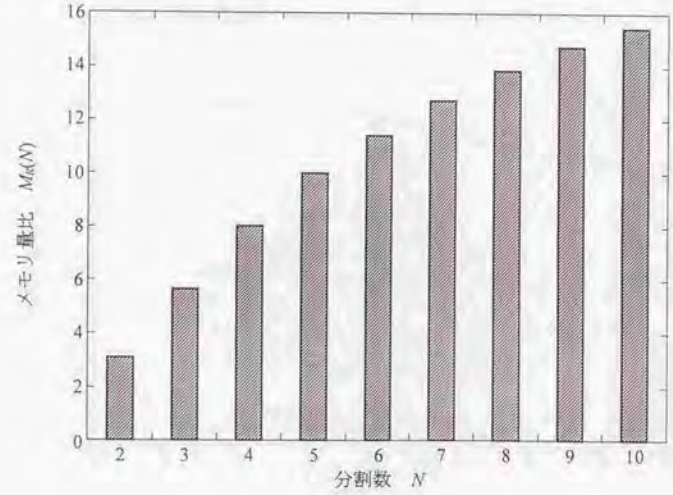


図3.24 メモリ量比（不等式拘束あり）

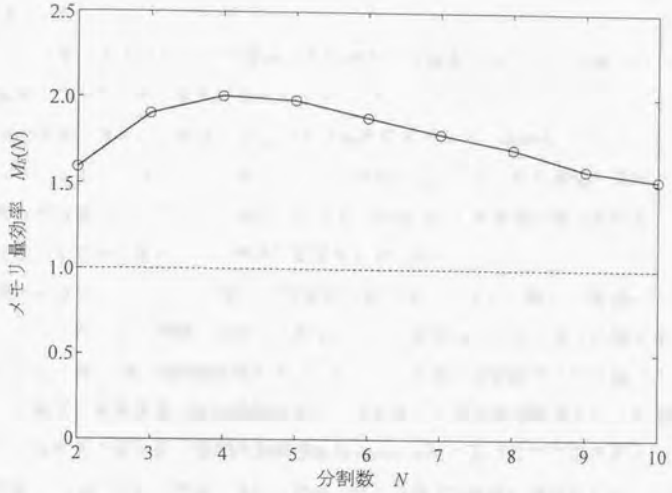


図3.25 メモリ量効率（不等式拘束なし）

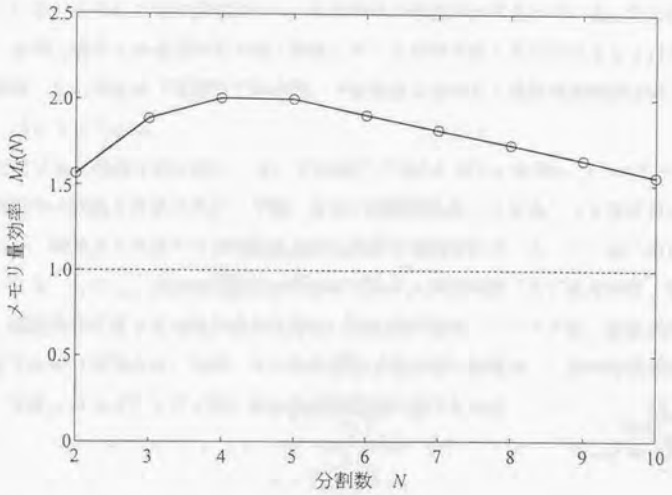


図3.26 メモリ量効率（不等式拘束あり）

3.4 まとめ

本章では前章で提案された並列最適化法の有効性を確認するために簡単な軌道最適化問題に適用してみた．その結果，次のようなことが分かった．

本論文で提案された並列最適化法は，実行速度効率とメモリ量効率が 1 を上回り，並列アルゴリズムとして望ましい特性をもつことがわかる．また，部分最適化問題が不等式制約条件を含む場合は，含まない場合と比べて，inner loop の収束に明らかに多くの反復計算を必要とするが，最終的には確実に最適解が得られる．

また過去の研究によると，一般に，問題を分割することなく一括して最適化する全体最適化法に比べて，部分問題に分割して最適化する分割解法はかなり多くの繰り返し計算を必要とする．第 1 章，補遺Bで取り上げられている従来の分割解法との比較は行っていないが，本論文の並列最適化法の反復回数は全体最適化法の反復回数の多くても数倍程度であることは注目に値する．従来の分割解法も inner loop のようなサブシステムレベルの最適化計算と outer loop に相当するシステムレベルの最適化計算に分けられる．しかしこれらの方法は inner loop が行われている間，他のサブシステムと変数の交換をしていない．本論文で提案する解法では inner loop の反復計算の間も常に他のサブシステムと変数値を交換している．このために本解法は少ない反復計算で最適解が得られていると思われる．ただし，並列に進められるプロセス間で頻繁にデータのやり取りが行われることは負担になる．実際，この負担は「最速降下線問題」で分割数を増やすと実行速度効率が低下した理由の 1 つとなっている．

本章では平易な問題を等分割し，それぞれのプロセスが等しい負荷になるようにした．本章の問題では最適化計算が実行不可能，あるいは数値計算上の問題から計算が停止することはなく，本論文で提案する並列最適化法は良好な並列化効率，ロバスト性と収束性をもつといえる．しかし，現実の問題はより複雑であり，部分問題ごとに変数の数，制約条件の数，関数評価に要する時間が異なるので，常に高い効率，ロバスト性，収束性が得られるかどうかは不明である．実際，後の章の問題では本章の例題ほどの並列化効率は得られない．また，カップリングが強い場合の影響は後に調べられる．

第5章
スペースプレーンの上昇軌道最適化

本章ではスペースプレーンの上昇軌道最適化問題を並列最適化法によって解く．この問題は最適制御問題であり，2.7節において説明した方法で並列最適化法を適用する．

現在，世界中の研究機関から様々な完全再使用型の次世代宇宙輸送システムが提案されている[32, 33]．それらは単段式（Single Stage To Orbit: SSTO）か多段式か[34]，また離陸と着陸の形態によって分類される．垂直離着陸型の再使用ロケット[35]，翼を利用して水平離着陸可能なスペースプレーン[2]と呼ばれる宇宙輸送機がその代表である．本論文では，空気吸い込み式（air-breathing）エンジンとロケットエンジンの両者を推進器として使用するスペースプレーンを考える．スペースプレーンが従来のロケットと大きく異なる特徴の一つは水平離着陸フェーズが存在することである．スペースプレーンの過酷なミッションを成功させるためには，空力，推進，構造，制御等の各技術間のトレードオフが従来の航空機あるいはロケットに比べ非常に重要である[36]．これまでのスペースプレーンの概念設計に関する研究では，スペースプレーンの重量解析を中心に，設計，上昇軌道の最適化が行われている．

以上を踏まえ，本章では，後章でスペースプレーンの概念設計を視野に入れた異分野どうしの統合的最適化問題を解く前に，上昇飛行において消費される推進剤重量を最小とする飛行経路を求める最適制御問題を解くことにする．従来の研究と比べて，解の精度を上げるためにより多くの変数を最適化することにし，並列最適化法を用いることで計算負荷の軽減を行う．本章の最後に，並列最適化法と従来の解法，すなわち全体最適化法の収束性の比較が行われ，並列最適化法の特徴が説明される．

5.1 問題設定

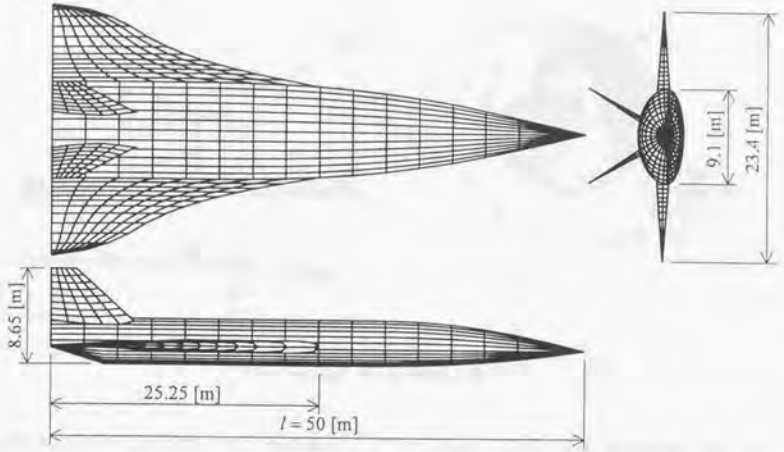


図5.1 スペースプレーンNAL0次形状

ここで考慮するスペースプレーンは図5.1のような形状をしている．この形状はNAL0次形状と呼ばれ，科学技術庁航空宇宙技術研究所にて提案された．亜音速から極超音速までの揚力係数と抗力係数が文献37の中で風洞試験によって求められ関数として与えられていることから，このデータを使い軌道を求めることにする．離陸重量$W_{takeoff}$は100 [ton]，機体全長lは50 [m]と決める．

スペースプレーンは作動条件が異なる3種類のエンジン，エアターボラムジェット（air-turboramjet: ATR）[38]，スクラムジェット（scramjet: SCR）[39]，ロケット（rocket: ROC）エンジンを持っている．3種類のエンジンは作動条件が異なっており，ATRはマッハ0からマッハ6程度まで，SCRはマッハ5からマッハ15程度まで，ROCはあらゆる条件で使用可能である．

スペースプレーンは始めATRを使用して離陸する．その後マッハ6に到達したところで，ATRを停止し，SCRを点火する．ここで，ATRとSCRの作動可能マッハ数は重なっており，およそマッハ5からマッハ6の幅で切り替えマッハ数を選ぶことができる．しかし，文献36によれば，切り替えマッハ数の違いにより燃料消費量に大きな差は生じないことから，切り替えマッハ数は6で一定とした．SCRで飛行するスペースプレーンは作動

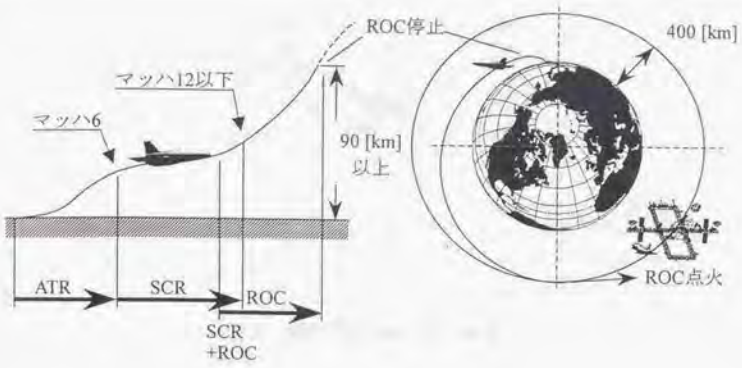


図5.2 上昇軌道計画

限界前に ROC に切り替える必要がある．ここで，ROC はあらゆる状態で使用可能であり，また構造上 SCR とは全く異なるエンジンであることに注目する．SCR の使用限界であるマッハ 12 に近付き，また高度を上げるにつれ，SCR の比推力と推力は低下する．そこで，SCR 使用区間の後半に ROC を併用することを考える．SCR はマッハ 12 以下で停止し，最後は ROC のみで加速，上昇を続ける．その後，高度 90 [km]以上で遠地点高度 400 [km]の楕円軌道に入った時点で一旦 ROC を停止する．そして，高度 400 [km]で再び ROC を点火し増速して，スペースプレーンは高度 400 [km]の円軌道に投入される．

軌道を最適化するにあたり，最小化する評価関数は円軌道に到達するまでの推進剤消費重量とする．

5.2 運動方程式

機体を質点モデルで表現し，鉛直軌道面内の運動のみを考える．また，地球を球で近似し，地球の自転の影響は無視する．図5.3のように幾つかの記号を設定すると，運動方程式はスペースプレーンの高度 h，速度 v，経路角γ，機体質量 m を状態変数，迎角αを制御変数として以下のように書ける．

$$\frac{\mathrm{d}h}{\mathrm{d}t} = v\sin\gamma \tag{5.1a}$$

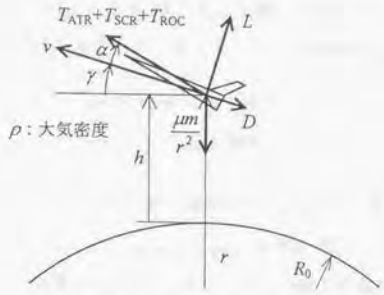


図5.3 運動方程式に必要な記号

$$\frac{dv}{dt}=\frac{(T_{ATR}+T_{SCR}+T_{ROC})\cos\alpha-D}{m}-\frac{\mu\sin\gamma}{r^2} \tag{5.1b}$$

$$\frac{d\gamma}{dt}=\frac{(T_{ATR}+T_{SCR}+T_{ROC})\sin\alpha+L}{mv}+\left(\frac{v}{r}-\frac{\mu}{vr^2}\right)\cos\gamma \tag{5.1c}$$

$$\frac{dm}{dt}=-\left(\frac{T_{ATR}}{I_{SP\,ATR}}+\frac{T_{SCR}}{I_{SP\,SCR}}+\frac{T_{ROC}}{I_{SP\,ROC}}\right)\frac{1}{g_0} \tag{5.1d}$$

ここで，μは重力定数であり，地表（$r=R_0$）での重力加速度g_0と

$$g_0=\frac{\mu}{R_0^2} \tag{5.2}$$

の関係にある．また，空気密度ρにより

$$q=\frac{1}{2}\rho v^2 \tag{5.3}$$

と表せる動圧qと，基準面積S_{ref}，NAL0次形状の揚力係数C_L，抗力係数C_D（補遺D，図D.8，図D.10）によって，揚力L，抗力Dは，

$$L=qC_LS_{ref} \tag{5.4a}$$

$$D=qC_DS_{ref} \tag{5.4b}$$

と計算される．ATR 推力T_{ATR}，SCR 推力T_{SCR}に関しては，それぞれのエンジンの推力係数$C_{T\mathrm{ATR}}$，$C_{T\mathrm{SCR}}$を図5.4にある通りに，またインテーク面積をS_{ATR}，S_{SCR}とすると，

$$T_{\mathrm{ATR}} = S_{\mathrm{ATR}} C_{T\mathrm{ATR}} \ [\mathrm{tonf}] \tag{5.5a}$$

$$T_{\mathrm{SCR}} = S_{\mathrm{SCR}} C_{T\mathrm{SCR}} \ [\mathrm{tonf}] \tag{5.5b}$$

と書ける．ただし，S_{ATR}，S_{SCR}の単位は[m²]とする．ROC の推力T_{ROC}は作動時には一定であるとする．式(5.1d)の$I_{\mathrm{SP\,ATR}}$，$I_{\mathrm{SP\,SCR}}$，$I_{\mathrm{SP\,ROC}}$はそれぞれ ATR，SCR，ROC の比推力である．$I_{\mathrm{SP\,ATR}}$，$I_{\mathrm{SP\,SCR}}$は図5.5に示す．なお，マッハ数の計算に必要な気温，空気密度ρは“U. S. Standard Atmosphere 1976”[40]より高度hの関数として与える．計算に用いた諸数値を表5.1にまとめる．

ここで，スペースプレーンは複数のエンジンを使用するため，エンジンを点火，停止するときに運動方程式が不連続に変化することに注意する．よって，使用するエンジンに従って制御時間を ATR 区間（$t_0^{\mathrm{ATR}} \le t \le t_{\mathrm{f}}^{\mathrm{ATR}}$），SCR 区間（$t_{\mathrm{f}}^{\mathrm{ATR}} = t_0^{\mathrm{SCR}} \le t \le t_{\mathrm{f}}^{\mathrm{SCR}}$），SCR+ROC 区間（$t_{\mathrm{f}}^{\mathrm{SCR}} = t_0^{\mathrm{SCR+ROC}} \le t \le t_{\mathrm{f}}^{\mathrm{SCR+ROC}}$），ROC 区間（$t_{\mathrm{f}}^{\mathrm{SCR+ROC}} = t_0^{\mathrm{ROC}} \le t \le t_{\mathrm{f}}^{\mathrm{ROC}}$）と分ける．

表5.1　スペースプレーンの軌道最適化問題の諸数値

諸元	値
機体全長　l [m]	50
基準面積　S_{ref} [m²]	325
ATR インテーク面積　S_{ATR} [m²]	14
SCR インテーク面積　S_{SCR} [m²]	14
ROC 推力　T_{ROC} [tonf]	70
ROC 比推力　$I_{\mathrm{SP\,ROC}}$ [sec]	450
離陸重量　W_{takeoff} [ton]	100
重力定数　μ [km³/sec²]	3.986×10^{5}
地球半径　R_0 [km]	6378

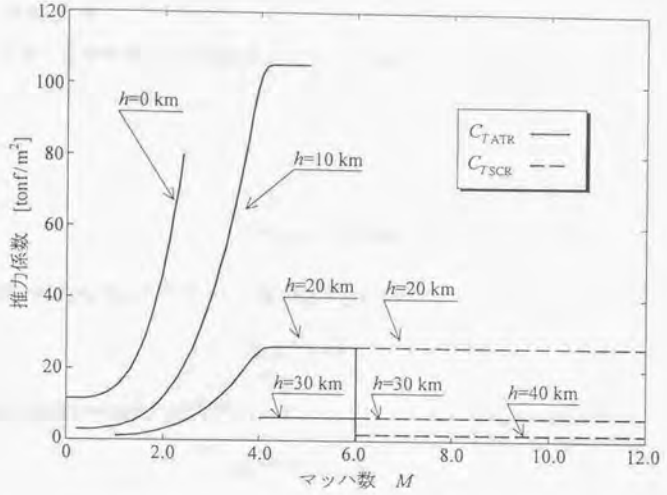


図5.4 ATR と SCR の推力係数 $C_{T\,ATR}$, $C_{T\,SCR}$ [2)]

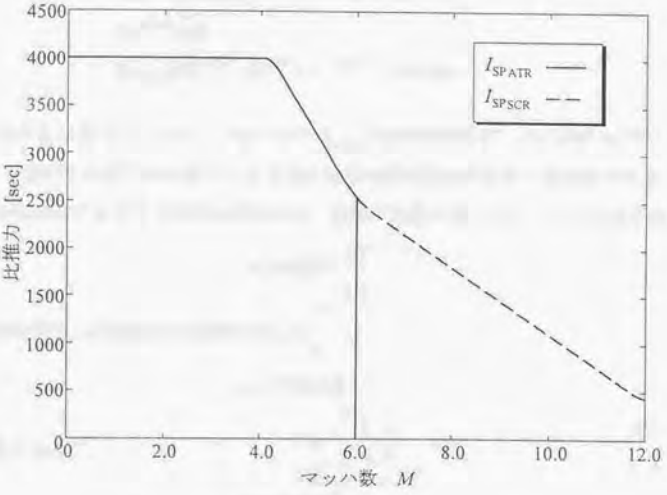


図5.5 ATR と SCR の比推力 $I_{SP\,ATR}$, $I_{SP\,SCR}$ [2)]

5.3 拘束条件

ATR 区間の初期時間 t_0^{ATR} を離陸時とすると，初期条件として，

$$t_0^{\mathrm{ATR}} = 0 \tag{5.6a}$$

$$h(t_0^{\mathrm{ATR}}) = 0 \tag{5.6b}$$

$$v(t_0^{\mathrm{ATR}}) = 300\,[\mathrm{m/sec}] \tag{5.6c}$$

$$m(t_0^{\mathrm{ATR}}) = W_{\mathrm{takeoff}} = 100\,[\mathrm{ton}] \tag{5.6d}$$

ATR 区間の終端時間 t_f^{ATR} でのマッハ数 $M(t_f^{\mathrm{ATR}})$ に対して，

$$M(t_f^{\mathrm{ATR}}) = 6 \tag{5.7}$$

SCR+ROC 区間の終端条件 $t_f^{\mathrm{SCR+ROC}}$ として，

$$M(t_f^{\mathrm{SCR+ROC}}) \le 12 \tag{5.8}$$

ROC 区間の終端時間 t_f^{ROC} を高度 90 [km]以上で ROC を停止し楕円軌道投入時点とすると，

$$h(t_f^{\mathrm{ROC}}) \ge 90\,[\mathrm{km}] \tag{5.9a}$$

$$\gamma(t_f^{\mathrm{ROC}}) \ge 0 \tag{5.9b}$$

$$H_{\mathrm{apogee}}\left[h(t_f^{\mathrm{ROC}}), v(t_f^{\mathrm{ROC}}), \gamma(t_f^{\mathrm{ROC}})\right] = 400\,[\mathrm{km}] \tag{5.9c}$$

の終端条件を設定する．なお，式(5.9c)の H_{apogee} は終端時間 t_f^{ROC} での高度 $h(t_f^{\mathrm{ROC}})$，速度 $v(t_f^{\mathrm{ROC}})$，経路角 $\gamma(t_f^{\mathrm{ROC}})$ から移行できる楕円軌道の遠地点高度を求める関数である．

制御時間全体にかかる不等式拘束条件は，機体が地面に潜らないように高度に対し

$$h \ge 0\,[\mathrm{km}] \tag{5.10}$$

また，機体構造上の理由から動圧に関して

$$q \le 100\,[\mathrm{kPa}] \tag{5.11}$$

失速を防ぐために，

$$\alpha \le 10\,[\mathrm{deg}] \tag{5.12}$$

を考慮することとする．

5.4 並列最適化法の適用

最小化される推進剤消費重量$W_{\text{propellant}}$は，離陸重量W_{takeoff}から高度 400 [km]円軌道投入後の重量を引けば求められる．円軌道投入時の速度増分をΔvとすると，

$$W_{\text{propellant}} = W_{\text{takeoff}} - m(t_f^{\text{ROC}}) \exp\left(- \frac{\Delta v\left[h(t_f^{\text{ROC}}), v(t_f^{\text{ROC}}), \gamma(t_f^{\text{ROC}})\right]}{g_0 I_{\text{SP ROC}}} \right) \tag{5.13}$$

と記述される．ここで，Δvは ROC 区間終端時間t_f^{ROC}での高度$h(t_f^{\text{ROC}})$，速度$v(t_f^{\text{ROC}})$，経路角$\gamma(t_f^{\text{ROC}})$の関数として表現される．なお，式(5.13)の第 1 項目W_{takeoff}は定数であり最小化に際し何の影響も与えないため除いてもかまわない．

本章で定式化された最適化問題は最適制御問題であり，ここでは BDH 法（補遺C）に従って非線形計画問題に変換する．離散化要素数は初期時間t_0^{ATR}から終端時間t_f^{ROC}まで 200 個とする．また，先にも述べた通り，制御時間は ATR 区間，SCR 区間，SCR+ROC 区間，ROC 区間の 4 つの区間に分割される．また，並列最適化法を適用するに当たり，2.7 節に従い，各区間から部分最適化問題を構成し，合計 4 つの部分最適化問題をもつ並列最適化を行う．最適化される各区間の要素数，各部分最適化問題の変数の数をまとめると，表5.2のようになる．

計算機はこれまでと同じ，Visual Technology 社製 VT-Alpha 667（CPU: Alpha 21164A 667 [MHz]，OS: Windows NT 4）を使用する．

表5.2 部分最適化問題の構成

部分最適化問題	要素数	変数の数
ATR 区間	37	192
SCR 区間	66	337
SCR+ROC 区間	20	107
ROC 区間	77	392
合計	200	1028

5.5 最適解

図5.6～5.14に得られた最適解を示す．評価関数である推進剤消費重量$W_{\text{propellant}}$は 75.38 [ton]となる．離陸時の迎角はさほど大きくない．翼の大きさはスペースプレーンを離陸させるのに十分な大きさである．離陸後，機体は動圧制限まで急激に加速，上昇するが，制限に到達した後は高度を上げることで空気密度を下げて動圧制限を越えないようにする．SCR に切り替えた後は一旦上昇を止め，高度を 30 [km]付近に保ちながら加速を続ける．SCR 区間の後半部分は加速が鈍るため，経路角を下げ，動圧制限に到達するまで高度を下げて加速を続ける．マッハ 10.51 に達した時点で ROC を点火する．その後，迎角が最大迎角になるまで，機首を引き起こし，加速，上昇をする．SCR は最大動作マッハ数であるマッハ 12.00 まで使用して停止する．ROC の推力角は迎角に対応している．よって，動圧が小さくなってからは推力角を調整して，最終的に高度 90 [km]，経路角 0 [deg]，速度 7.945 [km/sec]で楕円軌道の近地点に投入される．

得られた最適解の特徴は従来の研究報告と類似している．

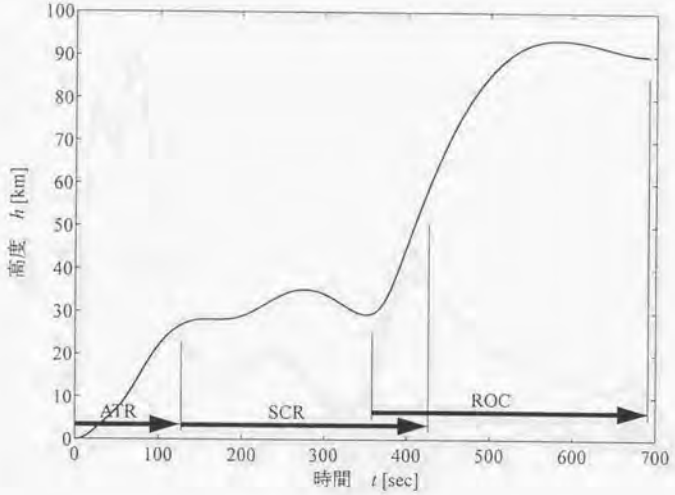


図5.6 高度の時間履歴

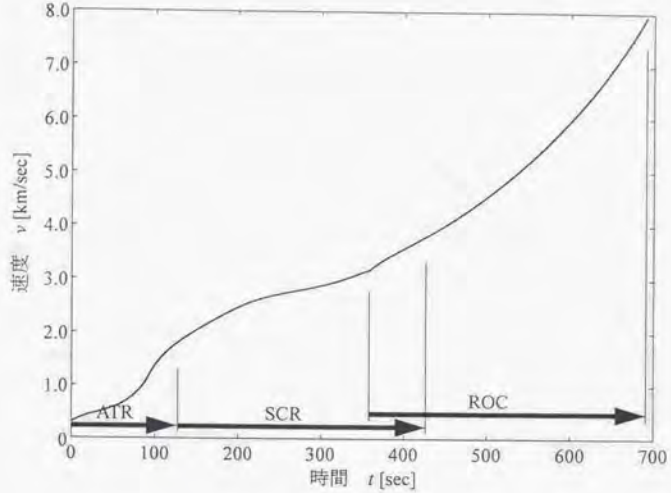


図5.7 速度の時間履歴

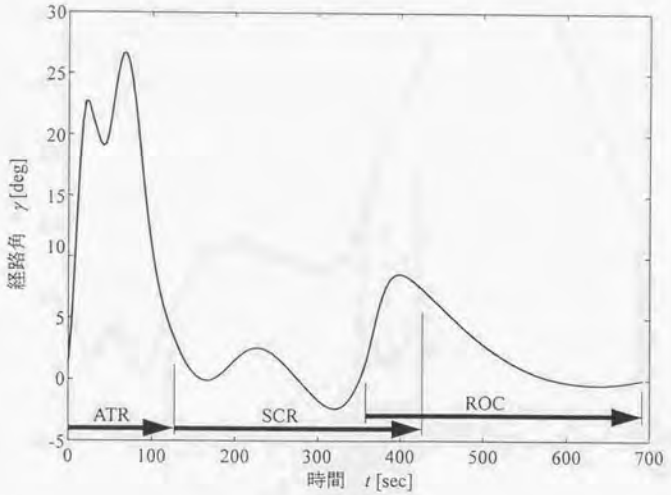


図5.8 経路角の時間履歴

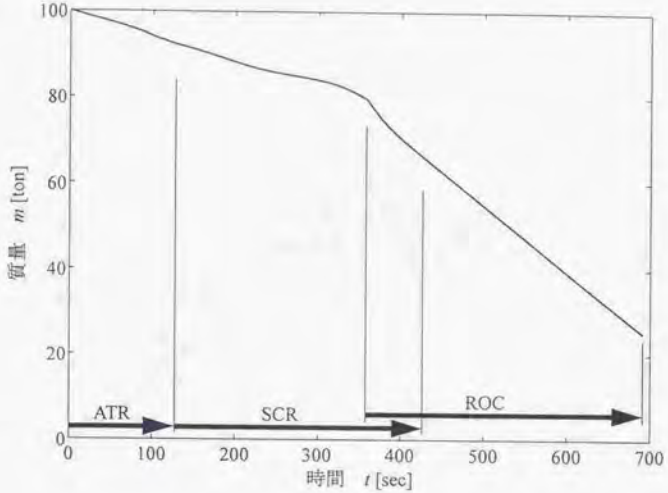


図5.9 質量の時間履歴

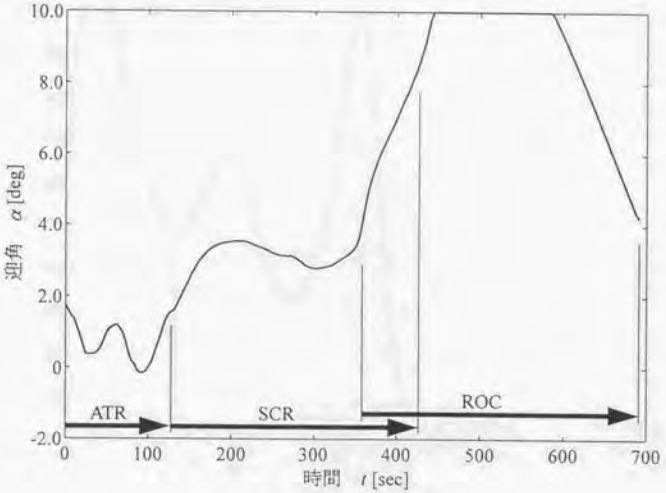


図5.10 迎角の時間履歴

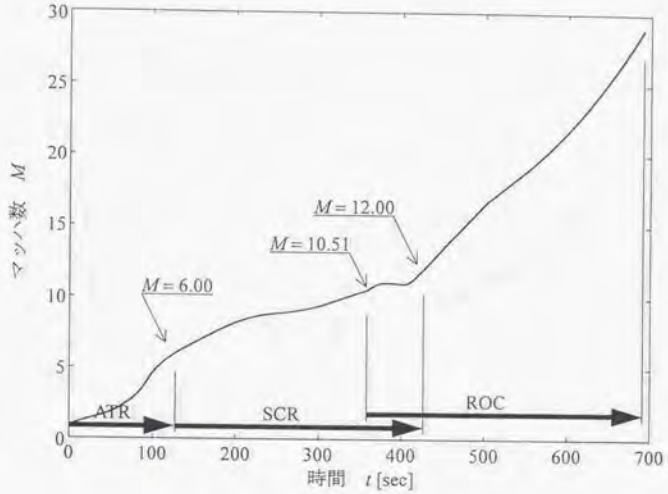


図5.11 マッハ数の時間履歴

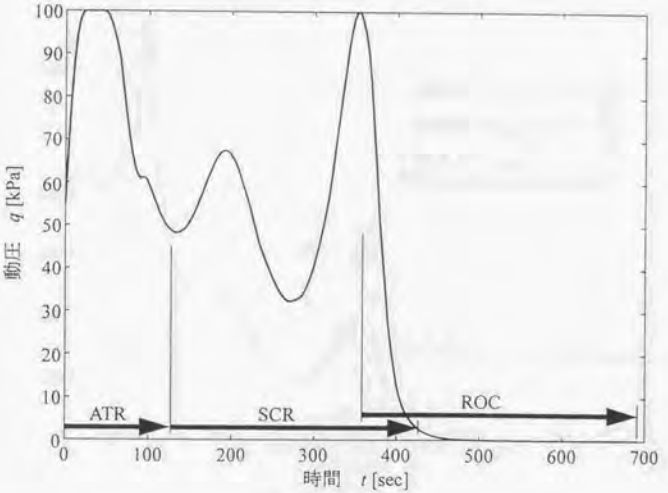


図5.12 動圧の時間履歴

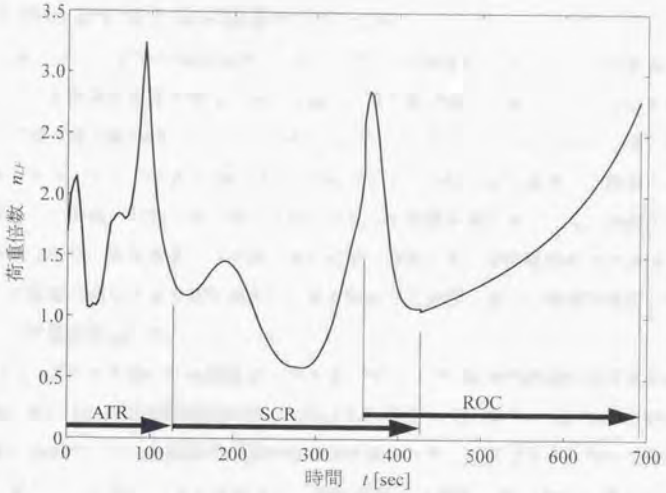


図5.13 荷重倍数の時間履歴

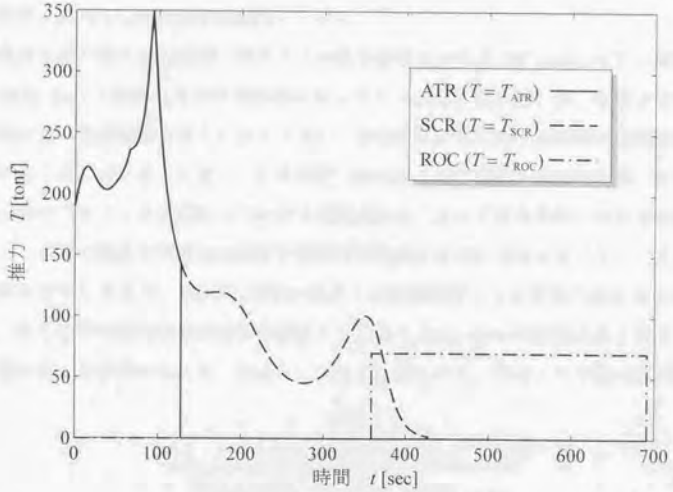


図5.14 推力重量比の時間履歴

5.6 全体最適化法と並列最適化法の比較

並列最適化法と，全ての制約条件，変数を一括して最適化を行っていく全体最適化法を比較する．本問題の変数の数は 1000 を超え，また制約条件の数もそれとほぼ同数存在するが，全体最適化法で解くことは不可能ではない．ただし，かなりのメモリ量と計算時間を必要とする．ここではROC 停止高度を 90 [km]から 100 [km]に変更し，前節の最適解を初期解として最適化計算を再び実行する．それ以上計算を繰り返しても，制約条件誤差が10^{-5}未満であり，全体最適化では繰り返し計算 1 回あたり，並列最適化では outer loop 1 回あたりの変数の最も大きな変化量が10^{-6}を上回ることが無くなった時点で解は収束したと判断して計算を停止した．

表5.3に 1 プロセス当たりに必要なメモリ量，図5.15に計算実行時間に対する目的関数値の推移，図5.16に計算実行時間に対する制約条件誤差の推移を示す．ここで制約条件誤差とは満たされていない制約条件の関数値の絶対値を合計した値であり，大きな値ほど制約条件を満たしていないことを意味する．制約関数は正規化されており，例えば，ROC 停止高度に関する制約条件では誤差 10 [km]が制約関数値 1 となっている．繰り返し計算を開始した時点では，ROC 停止高度以外の制約条件は全て満たされているため，図5.16の実行時間 0 [sec]での制約条件誤差は 1 となる．

全体最適化法の繰り返し計算 1 回当たりの計算時間はおよそ 160 [sec]に対し，並列最適化法の outer loop 1 回当たりの計算時間はおよそ 7 [sec]と非常に短いが，収束までの繰り返し回数は並列最適化法のほうがかなり多い．結果的に計算に要する時間は並列最適化法のほうが短く済んでいる．ただし，計算時間（特に繰り返し回数）は収束判定，最適化プログラム内のパラメータ値によってかなり左右される．よって計算時間に対する優劣は付けづらい．本章の最適化問題では前章の簡単な例題ほど並列化効率は良くない．その理由は，問題を分割したとき，表5.3に見られるように部分問題ごとに変数の数に偏りがあることと，多くの不等式制約条件が存在することにより inner loop の収束に多くの繰り返し回数を要することが挙げられる．なお，1 プロセス当たりに必要なメモリ量は並列最適化

表5.3 1プロセス当たりに必要なメモリ量

最適化手法	必要メモリ量 [kB]
全体最適化法	83832
並列最適化法	15160

法のほうが変数の数に応じて少なくすむ．

一方，図5.15によると，全体最適化法に比べて並列最適化法は最小化する目的関数がより小さい値に収束する．また，図5.16によると，並列最適化法の変数は全体最適化法よりも常に制約条件誤差が小さい値で推移する．これは並列最適化法の特徴の1つといえる．その理由として1次元探索のあり方が挙げられる．全体最適化法の1次元探索では問題全体で1つのペナルティ関数を考え，1つのステップ幅で変数全てを常に良い方向に更新していく．一方，並列最適化法では全ての変数を部分問題ごとに分け，各部分問題ごとに1次元探索を行い，システム全体からみると複数のステップ幅によって変数が更新されていく．よって，並列最適化法では1次元探索により更新されて取り得る変数値の範囲が全体最適化法の変数値の範囲よりも広い．したがって，全体最適化法では変数がそれ以上改善されないような値をとったとしても，並列最適化法によれば，さらに解を改善する方向を見つけ出す可能性がある．以上より，本章の問題に対し，並列最適化法によって全体最適化法よりも優れた解が得られた．これは並列最適化法を使用して最適化を行うことの大きな利点である．

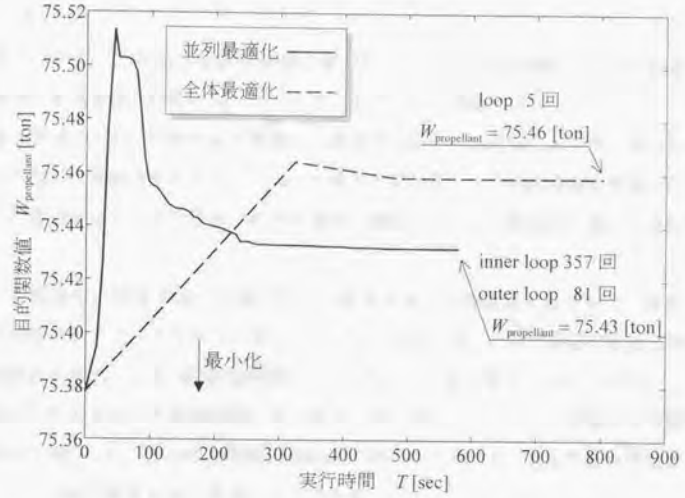


図5.15 計算実行時間に対する目的関数値（推進剤消費重量$W_{propellant}$）の推移

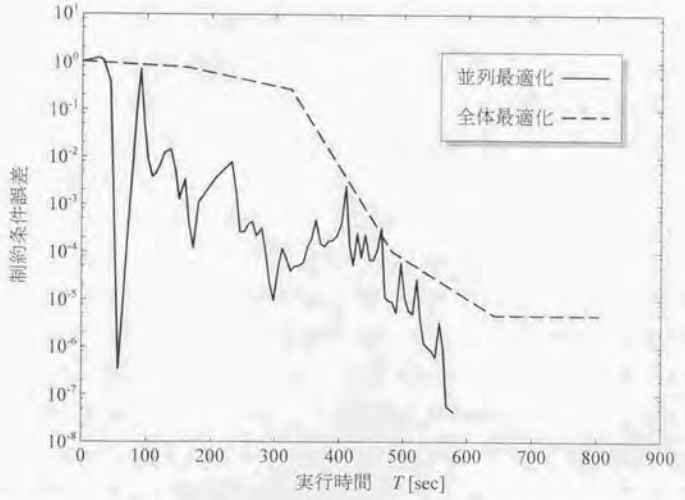


図5.16 計算実行時間に対する制約条件誤差の推移

5.7 まとめ

本章では前章まで単純な最適化問題に適用してその有効性を確認した並列最適化法をより複雑な軌道最適化問題に適用してみた．取り上げた問題はスペースプレーンの上昇軌道最適化問題であり，問題自体が複雑で非線形性が強く，また多くの要素で動的変数を離散化したため，複雑な制約条件と 1000 を超える変数をもつ大規模最適化問題となった．得られた最適解による飛行経路は従来の報告と類似しており，最適解に対して妥当性が認められた．

また，複雑な大規模最適化問題に対して並列最適化法を適用することで，前章までの簡単な問題を解くだけでは分からなかった解法の性質が得られた．複雑な最適化問題に並列最適化法を適用しても，簡単な問題ほどよい並列化効率を得ることはできない．しかし，最適解が予想できない本章の問題に並列最適化法を適用したところ，従来の全体最適化法を適用した場合に比べ，目的関数値，制約条件誤差の点で，より優れた解を得ることができた．これは並列最適化法の特徴の 1 つである．

第6章

スペースプレーンの機体設計／軌道の統合的最適化

本章では第 2 章で提案した並列最適化法をスペースプレーンの機体設計と軌道の統合的最適化問題に適用することによって，より実際に実在する工学問題に近い複雑な大規模最適化問題に対する並列最適化法の有効性を検証する．それに合わせてスペースプレーンの実現性についても考察する．

前章ではスペースプレーンの離陸重量を 100 [ton]と仮定して推進剤重量を最小とする上昇軌道を求めた．そこでは離陸重量を 100 [ton]と仮定した正当性，重量 100 [ton]の内訳，また推進剤が機体に収まるのか等，まったく考慮しなかった．実際，過去の研究[36]から，離陸重量の内訳を考慮してスペースプレーンの各構造部材の重量を求め，それを合計してみると離陸重量には収まらないという矛盾が生じることが分かっている．この矛盾が解消されない限り残念ながらスペースプレーンは実現されない．そこで，離陸重量からスペースプレーンを構成する上で最小限必要な重量を引いて求められるペイロード重量を最大化される目的関数とし，機体設計と上昇軌道の統合的最適化問題を定義する．最適化問題は，スペースプレーンを設計する際に関わる専門分野に応じて，大きく機体設計問題，空力解析問題，軌道計画問題の 3 つの部分最適化問題に分けられる．目的関数値の計算に必要な機体重量は機体形状から統計的推算法によって求めることにする．また，揚力係数と抗力係数の計算は横断流によるスレンダーボディ理論とニュートニアン理論，軸流による摩擦力，造波抗力，底面抗力の計算から簡易的に求めることにする．

加えて本章の最後に全体最適化法と並列最適化法の比較を行う．

6.1 問題設定

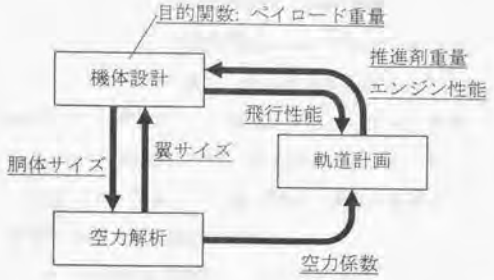


図6.1 本論文で考慮する専門分野とその接続条件，および目的関数

本章ではスペースプレーンの設計に関わる分野を図6.1に示すように，機体設計分野，空力解析分野，軌道計画分野の3つに分け，各分野から部分最適化問題を構成する．また，軌道計画分野は扱う変数の数が多く，運動方程式がエンジン切り替え時に不連続に変化するため，使用しているエンジンに従ってさらに細かく2〜4個の部分最適化問題に分ける．その結果，全部で4〜6個の部分最適化問題が定義され並列最適化される．

各分野にはその分野で最適化される変数とそれを制約する条件式が存在する．しかし，図6.1に示す胴体サイズ，翼サイズ，飛行性能，エンジン性能，推進剤重量，空力係数に関わる変数は複数の分野に含まれ，これらが最終的に得られる最適解において各分野ごとに異なる値を取ることは許されない．すなわち，各分野間で同じ意味をもつ変数の値は一致しなければならない．そこで，これら共通する変数の値が一致するという接続条件が定義される．前述した通り，接続関数はそこに含まれる変数をもつ一方の部分問題で制約条件として扱われ，他方の部分問題では目的関数に変換される．図6.1において矢印で表される接続関数を，矢先側の部分問題で制約条件に，矢尻側の部分問題で目的関数に含めることにする．なお，目的関数に含まれる接続関数に現れる変数と制約条件にある接続条件に現れる変数を比較した場合，1回の最適化計算で前者のほうがより大きな変化を取りうる，つまり，その部分問題で主に決めることができる変数である．よって，前者の変数を自由度が大きい変数と呼び，対する後者の変数を自由度が小さい変数と呼ぶことにする．

なお，2つの分野間にまたがる接続条件をどちらの分野で制約条件あるいは目的関数に変換するかを自動的に決定する方法は存在せず，経験または試行錯誤によるヒューリステ

ィックな判断が求められる．当然，第2章で述べたように，各部分問題が必ず目的関数をもつようにすることに注意すれば任意に接続条件を制約条件あるいは目的関数に変換できる．しかし，それによって，しばしば最適化計算途中で部分問題が実行可能解を失う場合がある．実際，図6.1に示されている通り，例えば機体設計と空力解析の間に存在する機体形状に関わる接続条件を，胴体サイズと翼サイズについての2種類の接続条件に分け，それぞれ制約条件あるいは目的関数とする分野を異なるようにしたのはそのような理由による．このように2種類に分けたほうが，部分問題が実行可能解をもたなくなり最適化が進まなくなる等の問題に直面する可能性が少なくなる．

本問題ではペイロード重量を目的関数とし最大化することを考える．ここでペイロード重量は機体設計分野の変数のみに依存すると仮定する．そのため，他の部分問題の目的関数は接続関数の項のみから定義されることになる．

6.1.1 機体設計

機体設計分野では主にスペースプレーンの機体形状と性能諸元値を変数として決定する．本論文で扱われるスペースプレーンの機体形状を図6.2に示す．機体は楕円柱とそれに滑らかに接続するtangent ogiveのノーズからなる胴体とデルタ翼より構成される単純な形状をしている（補遺D）．このような形状と仮定したのは次に述べる空力解析の手法によるためである．スペースプレーンは作動条件が異なる3種類のエンジン，エアターボラムジェット（air-turboramjet: ATR），スクラムジェット（scramjet: SCR），ロケット（rocket: ROC）エンジンを持ち，推進剤に液体水素（LH_2），液体酸素（LOX）を使用する．その

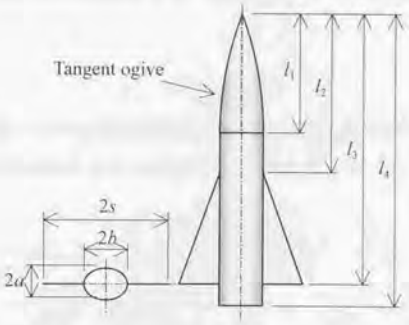


図6.2 スペースプレーンの機体形状モデル

混合比 LOX/LH_2 は，ATR と SCR では 0，ROC では 6 とする．密度は LOX が 1.149 [g/cm^3]，LH_2 が 0.071 [g/cm^3]である．以後，本分野からなる部分最適化問題に存在する変数と制約条件および目的関数を説明する．

- 変数

本分野では次の 15 個の変数が存在する．

胴体サイズ

l_1，l_4，a，b

翼サイズ

l_2，l_3，s

飛行性能

最大動圧 $q_{\max}$，最大荷重倍数 $n_{LF\max}$，最大推力 $T_{\max}$

エンジン性能

ATR インテーク面積 S_{ATR}，SCR インテーク面積 S_{SCR}，ROC 推力 T_{ROC}

推進剤重量

ATR と SCR のみで消費された LH_2 重量 $W_{\mathrm{ATR+SCR}}$，推進剤消費重量 $W_{\mathrm{propellant}}$

ここで推進剤重量を $W_{\mathrm{ATR+SCR}}$ と $W_{\mathrm{propellant}}$ に分けた理由は後に述べられる．これらの変数のうち，接続関数が目的関数にあり，比較的自由に決定することができる自由度の大きい変数に分類されるのが胴体サイズと飛行性能を表す変数である．また，制約条件にある接続関数に含まれる自由度の小さい変数として，翼サイズ，エンジン性能，推進剤重量に関する変数がある．

- 制約条件

自由度の大きい胴体サイズに関する変数に対して，胴体形状が矛盾を生じずに成立するために次のような不等式制約条件を定義する．

$$l_4 \geq l_1 \geq 0 \tag{6.1a}$$

$$a \geq 3\,[\mathrm{m}] \tag{6.1b}$$

$$b \geq 3\,[\mathrm{m}] \tag{6.1c}$$

また，$W_{\mathrm{ATR+SCR}}$ と $W_{\mathrm{propellant}}$ から求められる LH_2 と LOX タンクの容積の和が胴体体積の 70%

以下にする不等式制約条件を置く．タンク容積については補遺E，胴体体積のうちノーズ体積については補遺Dを参考にされたい．加えて，最大動圧，最大荷重倍数に関して，

$$q_{\max} \leq 100\,[\mathrm{kPa}] \tag{6.2a}$$

$$n_{LF\max} \leq 4\,[\mathrm{G}] \tag{6.2b}$$

で表される上限値を置く．

また，翼サイズ，エンジン性能，推進剤重量の変数に関する接続関数を等式制約条件とする．

- 目的関数

機体設計問題の目的関数には最大化するペイロード重量W_{payload}が定義される．ペイロード重量W_{payload}は，

$$W_{\mathrm{payload}} = W_{\mathrm{takeoff}} - \left(W_{\mathrm{structure}} + W_{\mathrm{propellant}} + W_{\mathrm{auxiliary}}\right) \tag{6.3}$$

より計算される．ここで，W_{takeoff}，$W_{\mathrm{structure}}$，$W_{\mathrm{propellant}}$，$W_{\mathrm{auxiliary}}$はそれぞれ離陸重量，構造重量，推進剤消費重量とその他の付加重量である．本問題において離陸重量W_{takeoff}はある仮定した一定値とするが，推進剤消費重量$W_{\mathrm{propellant}}$は本分野の変数の 1 つである．また構造重量$W_{\mathrm{structure}}$と付加重量$W_{\mathrm{auxiliary}}$は本分野の変数から求められ，これらの計算には重量推算プログラム WAATS を一部変更して使用する．WAATS についての詳細は補遺Eで説明されている．

さらに，胴体サイズ，飛行性能に関する接続関数とそのラグランジュ乗数を掛け合わせた項の和を上のペイロード重量W_{payload}に加えて機体設計問題の目的関数とする．

6.1.2 空力解析

空力解析には補遺Dに述べられている空力解析プログラム CRSFLW を多少変更して使用する．このプログラムによれば，図6.2に沿った任意の形状の機体について，任意の迎角とマッハ数下での空気力を推算することができる．なお，胴体下部にあるエンジンが空気力に及ぼす影響を，CRSFLW では解析ができないため考えない．

空力解析分野と軌道計画分野を分けるためには，マッハ数と迎角の連続な関数である揚力係数と抗力係数を離散化し，これら係数を求めるためのパラメータの値を軌道計画分野に伝達するようにしなければならない．そこで，空力解析分野では補遺Dに記述されて

いる通り 5 つのマッハ数をサンプリングし，各マッハ数ごとに 3 個づつの合計 15 個のある係数値 C_{A0}^{i}，C_{N1}^{i}，C_{N2}^{i}（$i=1,2,3,4,5$）を求める．ここではこれを空力係数と呼び，この空力係数を用いれば，任意の迎角とマッハ数の揚力係数と抗力係数を容易に求めることができる．

- 変数

本分野では次の 22 個の変数が存在する．

胴体サイズ

l_1，l_4，a，b

翼サイズ

l_2，l_3，s

空力係数

C_{A0}^{i}，C_{N1}^{i}，C_{N2}^{i}（$i=1,2,3,4,5$）

この中で，自由度が大きい変数は翼サイズと空力係数に関する変数である．そのため空力解析では，機体設計分野から与えられる胴体サイズを受け，翼サイズと空力係数を最適化する部分最適化問題を定義したといえる．

- 制約条件

胴体サイズと翼サイズの変数から空力係数を求める関数が等式制約条件になる．また，自由度が大きい翼サイズの変数に対して，翼形状に矛盾が生じないための不等式制約条件を定義する．

$$l_4 \geq l_3 \geq l_2 \geq l_1 \tag{6.4a}$$

$$s \geq a \tag{6.4b}$$

胴体サイズの変数に関する機体設計分野との間の接続関数が接続条件に加えられる．

- 目的関数

翼サイズと空力係数の変数に関する接続関数がそのラグランジュ乗数と掛け合わされ，目的関数として定義される．

6.1.3 軌道計画

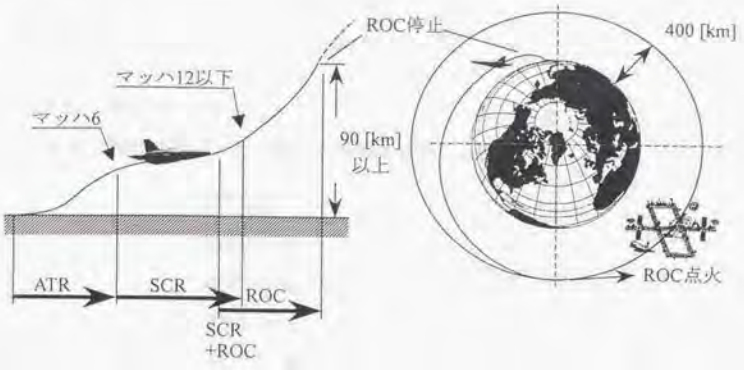


図6.3 上昇軌道計画

スペースプレーンの飛行計画は第 5 章とほぼ同じである．スペースプレーンは離陸後，マッハ 6 まで ATR，最大マッハ 12 まで SCR，および ROC エンジンを利用して加速上昇し高度 90 [km]以上に達する．ここで ROC を一旦停止した後，無推力で高度 400 [km]まで到達し ROC を再点火して円軌道に投入される．スペースプレーンは複数のエンジンを使用するため，運動方程式はエンジンを点火，停止するときに不連続に変化する．よって，使用するエンジンに従って制御時間を ATR 区間（$t_0^{\mathrm{ATR}} \le t \le t_f^{\mathrm{ATR}}$），SCR 区間（$t_0^{\mathrm{SCR}} \le t \le t_f^{\mathrm{SCR}}$），SCR+ROC 区間（$t_0^{\mathrm{SCR+ROC}} \le t \le t_f^{\mathrm{SCR+ROC}}$），ROC 区間（$t_0^{\mathrm{ROC}} \le t \le t_f^{\mathrm{ROC}}$）と分け，各区間から部分最適化問題を構成する．したがって，軌道計画問題は 4 つの部分問題に細分化される．しかし，後の結果に示されるように，問題設定によっては SCR を使用しない飛行が最適解となる場合が存在する．このため，上述のエンジン使用法を基本形として最適化計算を始めるが，そこから得られた解のエンジン使用法に応じて部分問題の構成を変化させる．

運動方程式は第 5 章の式(5.1a)～(5.1d)で定義されている．すなわち，スペースプレーンは質点として扱われ，状態変数は高度 h，速度 v，経路角 γ，質量 m，制御変数は迎角 α とする．ここで，スペースプレーンの揚力 L，抗力 D は空力係数 C'_{A0}，C'_{N1}，C'_{N2} によって補遺Dで述べる方法に従って求める．エンジンモデルは図5.4，図5.5で示され，エンジン性能について，

$$S_{\mathrm{ATR}} \geq 0 \tag{6.5a}$$

$$S_{\mathrm{SCR}} \geq 0 \tag{6.5b}$$

$$T_{\mathrm{ROC}} \geq 0 \tag{6.5c}$$

という不等式制約条件が置かれる．

ATR 区間の初期時間 t_0^{ATR} を ATR による離陸時とし，初期条件として，

$$t_0^{\mathrm{ATR}} = 0 \tag{6.6a}$$

$$h(t_0^{\mathrm{ATR}}) = 0 \tag{6.6b}$$

$$v(t_0^{\mathrm{ATR}}) \leq 150\ [\mathrm{m/sec}] \tag{6.6c}$$

$$\gamma(t_0^{\mathrm{ATR}}) = 0 \tag{6.6d}$$

$$m(t_0^{\mathrm{ATR}}) = W_{\mathrm{takeoff}} \tag{6.6e}$$

$$L\cos\alpha + (T - D)\sin\alpha \geq W_{\mathrm{takeoff}} g_0 \tag{6.6f}$$

を定義する．各区間の初期と終端条件は第 5 章の問題設定と同じであり，すなわち，

$$M(t_f^{\mathrm{ATR}}) = 6 \tag{6.7a}$$

$$M(t_f^{\mathrm{SCR+ROC}}) \leq 12 \tag{6.7b}$$

$$h(t_f^{\mathrm{ROC}}) \geq 90\ [\mathrm{km}] \tag{6.7c}$$

$$\gamma(t_f^{\mathrm{ROC}}) \geq 0 \tag{6.7d}$$

$$H_{\mathrm{apogee}}\left[h(t_f^{\mathrm{ROC}}), v(t_f^{\mathrm{ROC}}), \gamma(t_f^{\mathrm{ROC}})\right] = 400\ [\mathrm{km}] \tag{6.7e}$$

とする．

また，ATR と SCR のみで消費された LH_2 重量 $W_{\mathrm{ATR+SCR}}$，推進剤消費重量 $W_{\mathrm{propellant}}$ を，それぞれエアブリージング（空気吸い込み式）エンジンの最終区間と ROC 区間の終端状態量から求める式が等式制約条件として存在する．例えば，エアブリージングエンジンの最終区間が ATR 区間であれば，

$$W_{\mathrm{ATR+SCR}} - \left\{W_{\mathrm{takeoff}} - m(t_f^{\mathrm{ATR}})\right\} = 0 \tag{6.8}$$

SCR 区間であれば，

$$W_{\mathrm{ATR+SCR}} - \left\{W_{\mathrm{takeoff}} - m(t_f^{\mathrm{SCR}})\right\} = 0 \tag{6.9}$$

SCR+ROC 区間であるならば，

$$W_{\mathrm{ATR+SCR}} - \left\{ W_{\mathrm{takeoff}} - m(t_{\mathrm{f}}^{\mathrm{SCR+ROC}}) + \frac{T_{\mathrm{ROC}}}{g_0 I_{\mathrm{SP\,ROC}}} (t_{\mathrm{f}}^{\mathrm{SCR+ROC}} - t_0^{\mathrm{SCR+ROC}}) \right\} = 0 \tag{6.10}$$

とする．$W_{\mathrm{ATR+SCR}}$ に ROC が消費する LH_2 を含めないことに注意する．さらに，推進剤消費重量 $W_{\mathrm{propellant}}$ を求める条件式は，高度 400 [km]円軌道投入時の速度増分を Δv とすると，

$$W_{\mathrm{propellant}} - \left\{ W_{\mathrm{takeoff}} - m(t_{\mathrm{f}}^{\mathrm{ROC}}) \exp\left(- \frac{\Delta v\left[h(t_{\mathrm{f}}^{\mathrm{ROC}}), v(t_{\mathrm{f}}^{\mathrm{ROC}}), \gamma(t_{\mathrm{f}}^{\mathrm{ROC}})\right]}{g_0 I_{\mathrm{SP\,ROC}}} \right) \right\} = 0 \tag{6.11}$$

と記述される．ここで，Δv は ROC 区間終端時間 $t_{\mathrm{f}}^{\mathrm{ROC}}$ での高度 $h(t_{\mathrm{f}}^{\mathrm{ROC}})$，速度 $v(t_{\mathrm{f}}^{\mathrm{ROC}})$，経路角 $\gamma(t_{\mathrm{f}}^{\mathrm{ROC}})$ の関数である．

飛行経路全体にかかる不等式拘束条件は，高度，動圧，迎角について，

$$h \geq 0 \tag{6.12a}$$

$$q \leq q_{\max} \tag{6.12b}$$

$$\alpha \leq 20\,[\mathrm{deg}] \tag{6.12c}$$

が定義される．また，本章の問題では推力に関する制限

$$T_{\mathrm{ATR}} + T_{\mathrm{SCR}} + T_{\mathrm{ROC}} \leq T_{\max} \tag{6.13}$$

と，空気力と推力からなる荷重倍数 n_{LF} に関する制限

$$n_{LF} = \frac{\sqrt{\{(T_{\mathrm{ATR}} + T_{\mathrm{SCR}} + T_{\mathrm{ROC}})\cos\alpha - D\}^2 + \{(T_{\mathrm{ATR}} + T_{\mathrm{SCR}} + T_{\mathrm{ROC}})\sin\alpha + L\}^2}}{mg_0} \leq n_{LF\max} \tag{6.14}$$

を拘束条件に加える．なお，$q_{\max}$，$T_{\max}$，$n_{LF\max}$ も飛行性能に関して最適化される変数であることに注意する．

ここで定義された軌道計画問題は，時間の関数である状態変数と制御変数を決定する問題であり，ここでは補遺Cの BDH 法に従って両変数とこれらの変数の汎関数である運動方程式と拘束条件を離散化する．離散化要素数は初期時間 t_0^{ATR} から終端時間 $t_{\mathrm{f}}^{\mathrm{ROC}}$ まで 200 個とする．

以上より，各区間の部分最適化問題には，最適化される変数として，飛行するマッハ

数に応じた空力係数と飛行性能に関わる変数，使用するエンジンの性能に関する変数，推進剤重量に関する変数，離散化された状態変数と制御変数，初期と終端時間が存在する．しかし，軌道計画問題に含まれる複数の部分問題の間にも幾つかの同じ意味の変数が存在する．よって，機体形状問題と空力解析問題との間だけでなく，軌道計画問題内の部分問題の間でも接続条件が必要になり，自由度の大きい変数と小さい変数は各部分問題ごとに異なる．詳細は次節の図6.4，6.5，6.6に示される．

6.2 並列最適化法の適用

図6.4，6.5，6.6はスペースプレーンの機体設計，空力解析，軌道計画の統合的最適化問題において，各部分最適化問題に含まれる変数とそれらの接続条件を示した図である．エンジンの使用法として 3 つの問題設定を考えたため 3 つの図が示されている．図6.4の 3 つのエンジンを 4 段階で使用するのが基本であり，後に示される最適解は全てこの問題設定で計算を開始した．しかし，後述するように，問題設定によって，SCR を使用しない解，SCR を単独で使用せずに ROC と併用して使用する 2 種類の解が得られるため，問題設定をそれに合わせて変更し，再度計算を行った．

部分最適化問題は，図6.4の場合では 6 個，図6.5の場合では 5 個，図6.6の場合では 4 個存在する．部分問題の内部には変数が各図で示されるように存在し，矢印で結ぶ変数どうしの値が一致するという接続条件が定義される．そしてこの接続関数は矢先側の部分問題では制約条件として，矢尻側の部分問題では目的関数として扱われている．

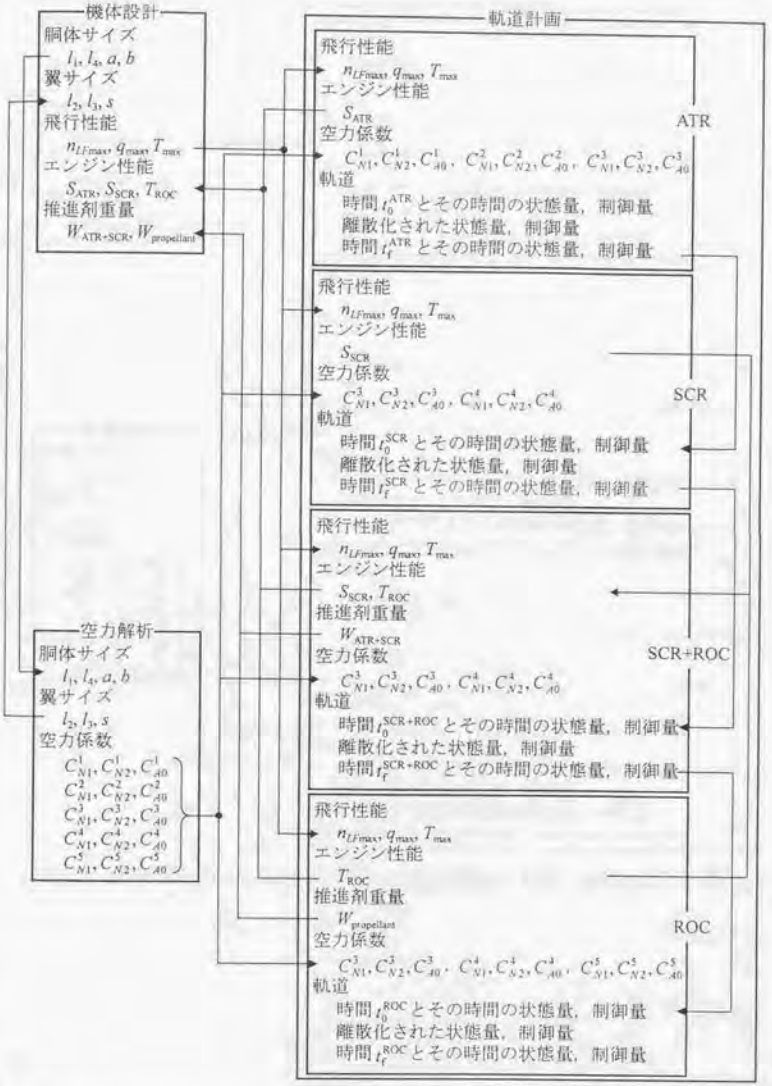


図6.4 スペースプレーンの統合的最適化問題における変数と接続条件（4段）

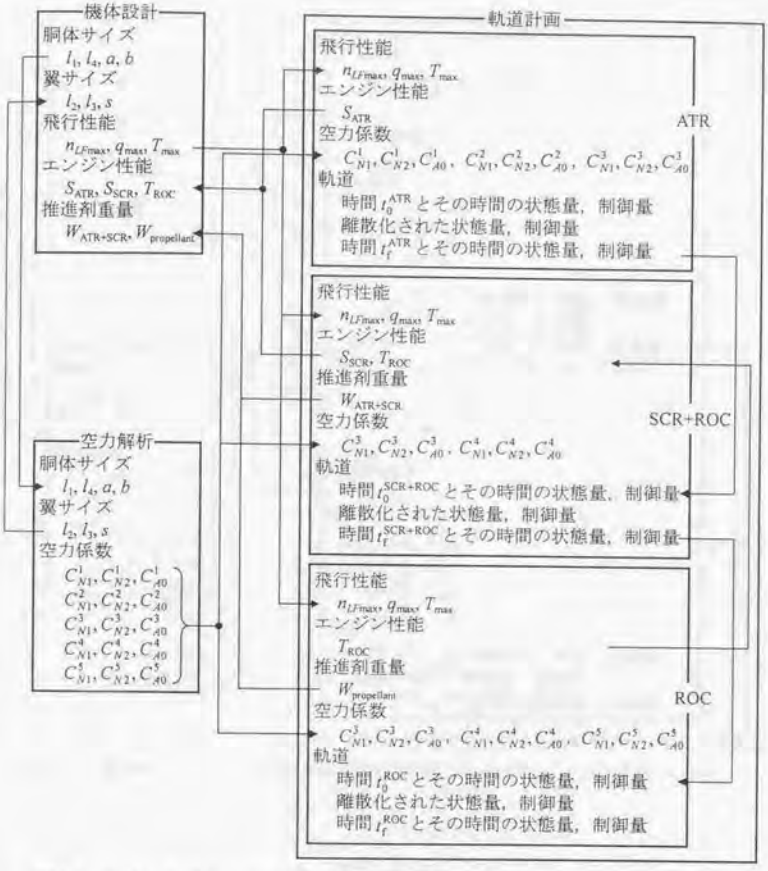


図6.5 スペースプレーンの統合的最適化問題における変数と接続条件（3段）

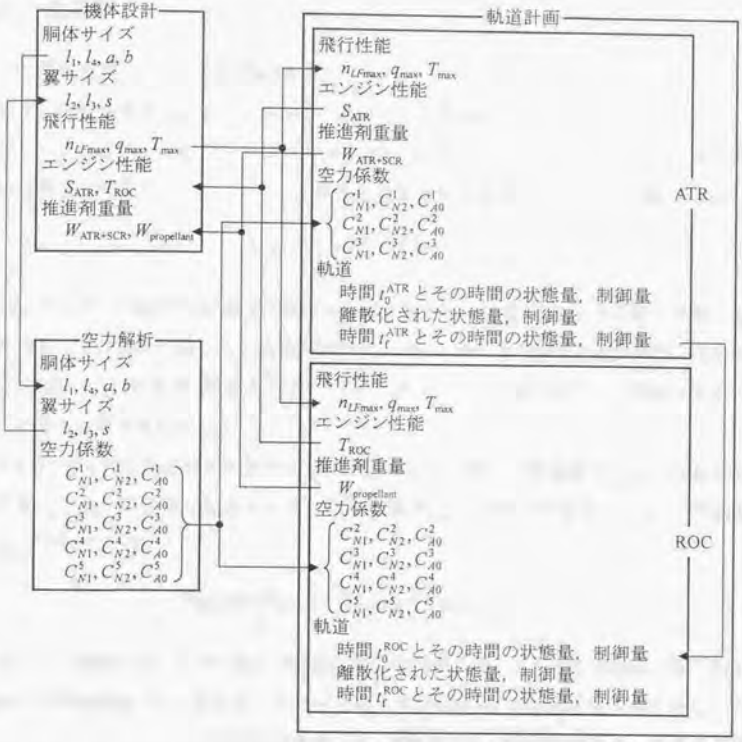


図6.6 スペースプレーンの統合的最適化問題における変数と接続条件（2段）

6.3 最適解

6.3.1 離陸重量 300 [ton]の計算結果

まず，離陸重量W_{takeoff}を 300 [ton]の一定値として最適解を求めてみる．はじめ，スペースプレーンは 3 つのエンジン ATR，SCR，ROC を使用すると仮定して，図6.4に示す問題設定で最適化計算を始めた．しかし，得られた最適解は SCR を使用しない解，すなわち，

$$S_{\text{SCR}} = 0, \quad t_0^{\text{SCR}} = t_f^{\text{SCR}} \tag{6.15}$$

となったため，問題設定を図6.6に示すエンジンを ATR と ROC のみの 2 段で使用する設定に変更して計算をし直した．軌道計画問題の BDH 法による離散化要素数は ATR 区間を 150 個，ROC 区間を 50 個とした．ATR 区間は大きく状態が変化し，複雑な軌道となるため多くの要素を配置した．

表6.1に得られた最適解の性能諸元を示す．ここで，最小必要重量W_{required}とは機体を構成する上で最小限必要な重量であり，構造重量$W_{\text{structure}}$，推進剤重量$W_{\text{propellant}}$，付加重量$W_{\text{auxiliary}}$の和である．

$$W_{\text{required}} = W_{\text{structure}} + W_{\text{propellant}} + W_{\text{auxiliary}} \tag{6.16}$$

すなわち，離陸重量からペイロード重量を引いた値である．この表によれば，得られた最適解の目的関数値である最大化されたペイロード重量は-13.75 [ton]と負の値である．これは現在スペースプレーンの重量推算を行うと一般的に現れる傾向[76]である．つまり，現在の技術では周回軌道に到達するまで飛行できるスペースプレーンを製作できない．この機体を実現するためには 313.75 [ton]ある最小必要重量を離陸重量の 300 [ton]以下に減らすために，5 %以上の重量軽減が必要である．表6.2，図6.7に詳細な機体構成要素の重量配分をまとめてある．構成要素については補遺Eを参照されたい．

図6.8は求められた機体形状を示している．翼スパンは翼の重量に大きな影響を及ぼすため，翼根翼弦長をできる限り大きくして翼スパンを小さく押さえるのが良い．また，胴体は抗力を軽減するために細長くなっている．図6.8にはほぼ同じ離陸重量である Boeing 747-400 の機体も比較のため示されている．両者はほぼ同じ大きさであることが分かる．

図6.9，6.10に揚力係数と抗力係数，図6.11～6.19に飛行履歴を示す．スペースプレーンは離陸時に最大迎角を取る．すなわち，翼の大きさは亜音速時，特に離陸条件から決まる．

超音速を超えれば揚力は充分に足りているため，翼の構造重量を小さく押さえる必要性から可能な限り小さな翼が望ましい．離陸後，スペースプレーンは ATR を使用して加速上昇するが，得られた解の ATR は遷音速時の抗力に打ち勝って加速できる限界のサイズであるため，しばらくは低高度をゆっくりと加速しながら飛行する．その後，動圧制限に達したところで高度を上げ，制限を超えないように飛行する．ATR 区間の後半はマッハ数が大きくなるため ATR 推力が増加し，推力制限に拘束されて飛行することになる．マッハ 6 に達したところで，ATR から ROC に切り替えるが，ROC の推力はその時点の重量より小さな値であるため，機体は一度大きく降下し加速してから引き起こされて上昇をする．高度 90 [km]に達した時，動圧はほぼ 0 になり，ちょうど遠地点高度が 400 [km]の軌道上にあるため ROC を停止する．

以上のように，胴体，翼，エンジンの大きさ等の諸元値と飛行経路には無駄が少ないことから，ここで得られた最適解は妥当であるといえる．なお，ATR と ROC を併用でき，ATR の停止マッハ数を 6 以下にする問題設定に変更して最適化計算を行ったが本項と同じ解が得られた．

表6.1 離陸重量 300 [ton]での諸元値の最適解

諸元	最適解
胴体全長 l_4 [m]	63.48
ノーズ長さ l_1 [m]	32.99
胴体高さ $2a$ [m]	6.00
胴体幅 $2b$ [m]	6.36
翼スパン $2s$ [m]	20.04
最大動圧 q_{max} [kPa]	100.0
最大荷重倍数 n_{LFmax} [G]	3.82
最大推力 T_{max} [tonf]	226.6
ATR インテーク面積 S_{ATR} [m^2]	12.59
SCR インテーク面積 S_{SCR} [m^2]	0.00
ROC 推力 T_{ROC} [tonf]	226.6
LH_2 重量 W_{LH2} [ton]	60.46
LOX 重量 W_{LOX} [ton]	181.50
構造重量 $W_{structure}$ [ton]	63.30
付加重量 $W_{auxiliary}$ [ton]	8.50
最小必要重量 $W_{required}$ [ton]	313.75
ペイロード重量 $W_{payload}$ [ton]	-13.75

表6.2 離陸重量300 [ton]のスペースプレーンの重量内訳

機体構成要素	重量 [kg]
body structure	16163
wing	7397
vertical fin	0
horizontal stabilizer	0
landing gear	5961
thrust structure	2322
ATR engine	14476
SCR engine	0
ROC engine	2946
liquid hydrogen tank	7229
liquid oxygen tank	3163
liquid hydrogen system	822
liquid oxygen system	349
pressurization system	2467
avionics	1900
aerodynamic control system	1667
electrical system	1635
hydraulic/pneumatic system	387
OMS	220
OMS tank	22
OMS propellant	1500
crew, crew life support	1171
胴体/翼	29521
エンジン	19744
タンク/配管系	14030
制御/動力系	5589
その他	2913
LH_2	60459
LOX	181498
構造重量	63295
付加重量	8502
推進剤重量	241957
ペイロード重量	-13754

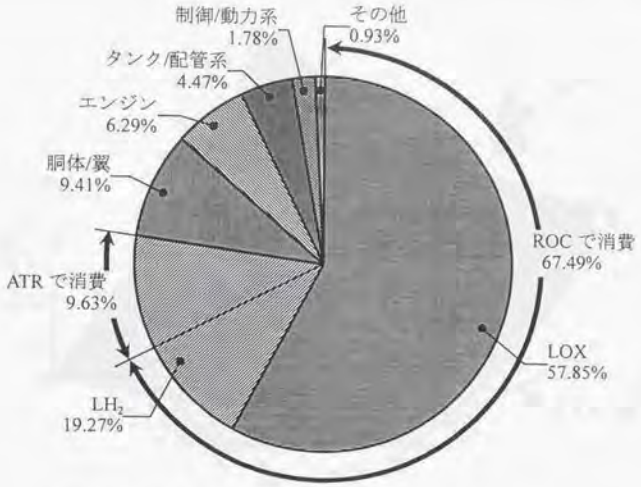


図6.7 離陸重量 300 [ton]のスペースプレーンの重量内訳

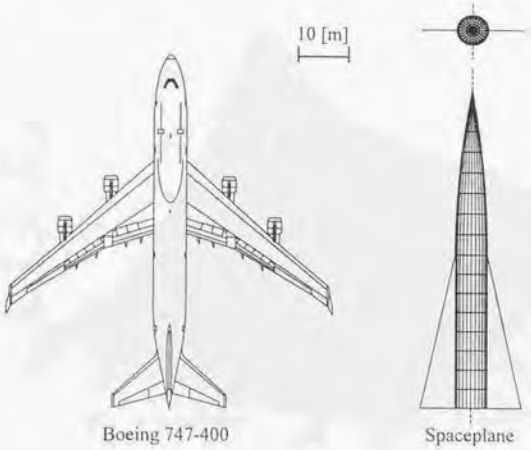


図6.8 離陸重量 300 [ton]のスペースプレーン最適機体形状と Boeing 747-400

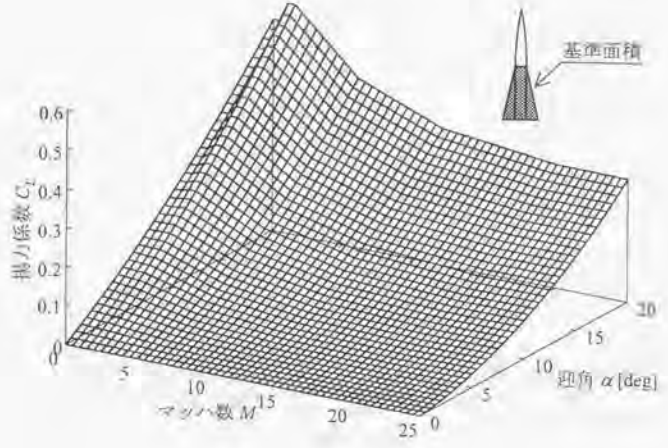


図6.9　揚力係数

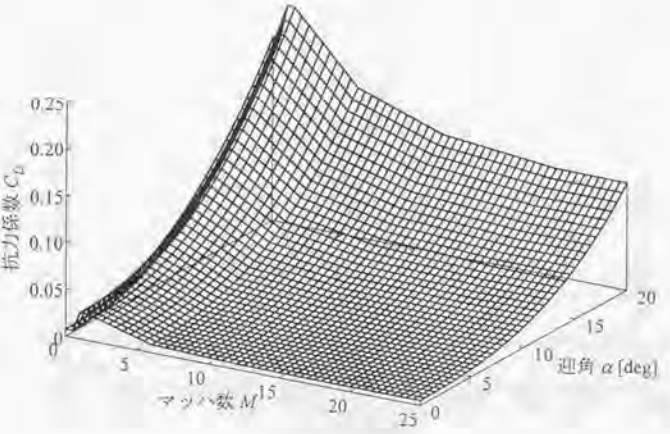


図6.10　抗力係数

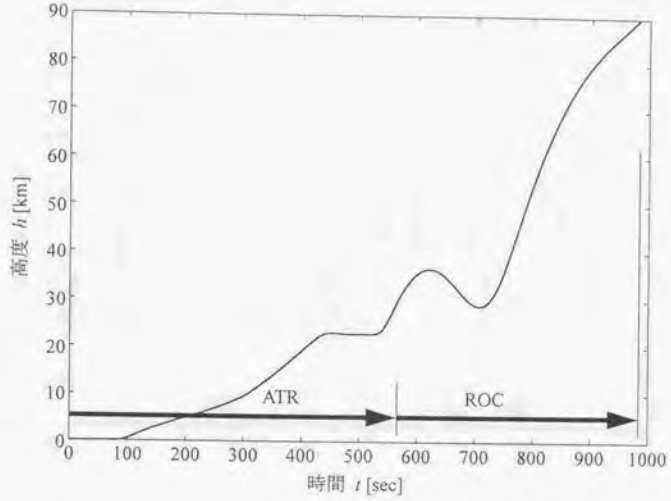


図6.11　高度の時間履歴

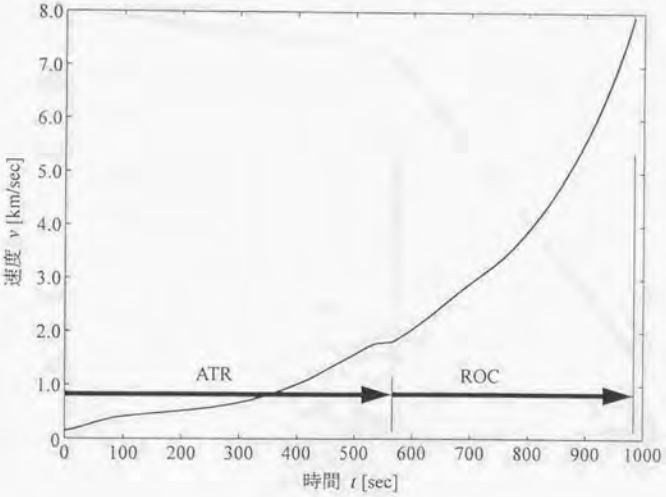


図6.12　速度の時間履歴

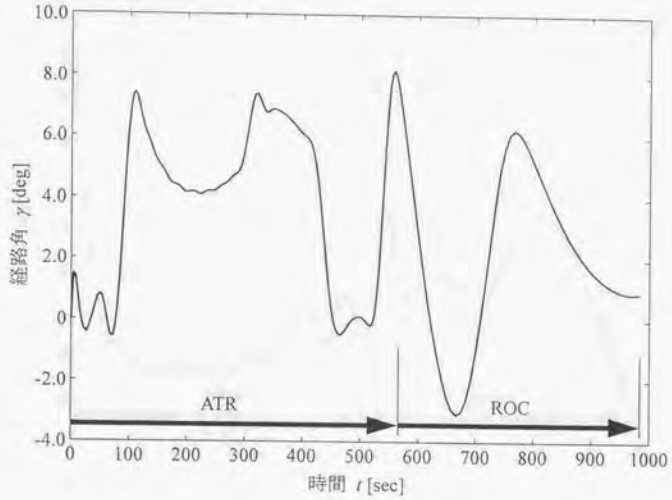


図6.13　経路角の時間履歴

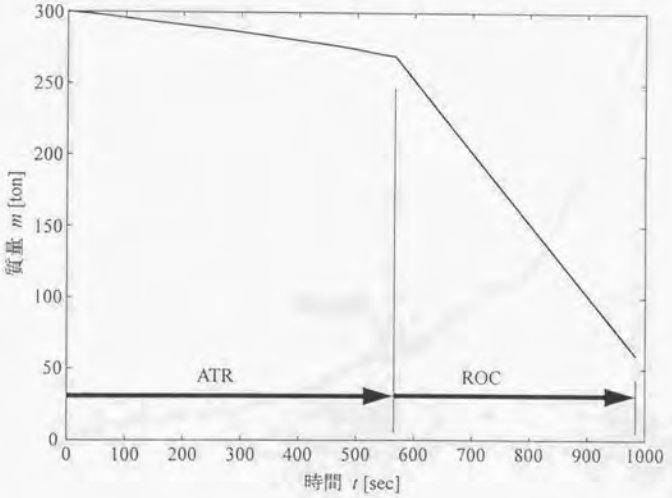


図6.14　質量の時間履歴

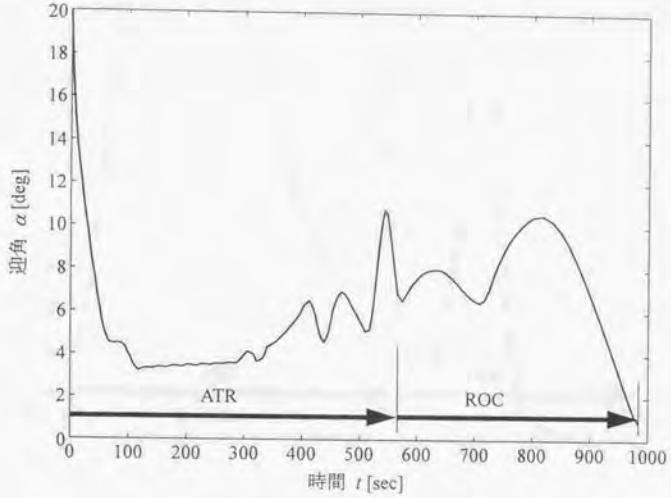


図6.15 迎角の時間履歴

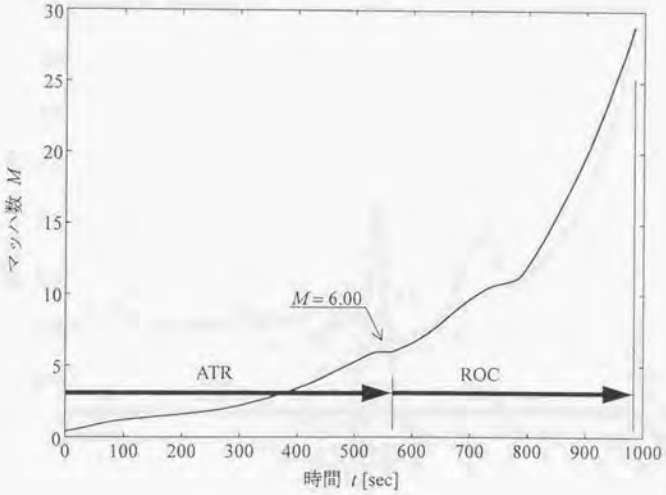


図6.16 マッハ数の時間履歴

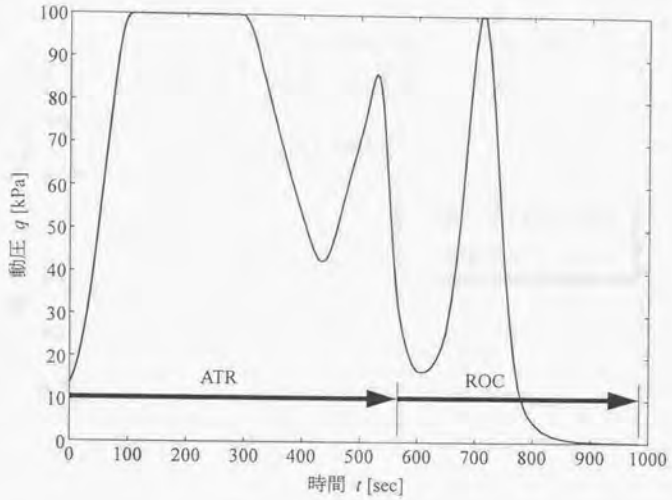


図6.17 動圧の時間履歴

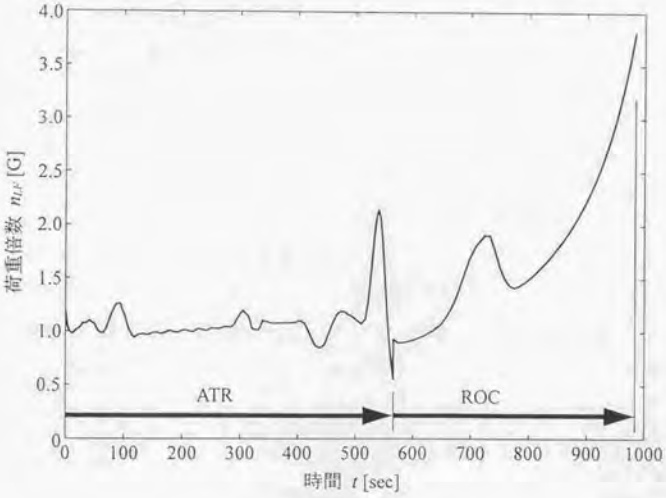


図6.18 荷重倍数の時間履歴

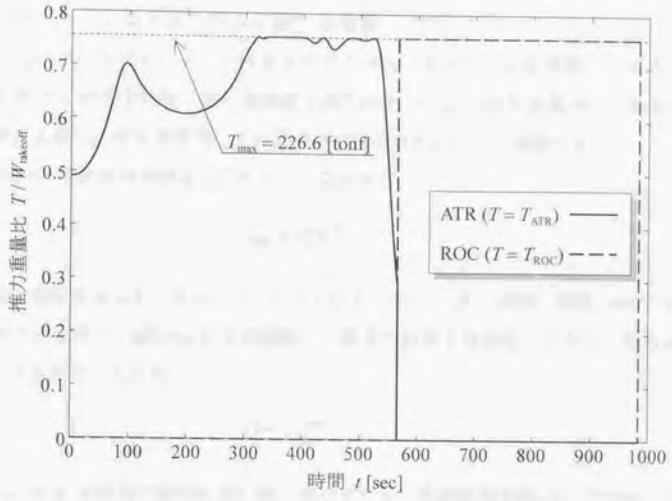


図6.19 推力重量比の時間履歴

6.3.2 スクラムジェットエンジンに関する考察

本項ではスペースプレーンにおけるスクラムジェットエンジンを考察してみる．前項の問題設定ではSCRを使用しない最適解が得られた．そこでSCRを使用した場合の解と前項の解を比較し，SCRを使用しないほうが有利な理由について考察する．

式(6.5b)の不等式制約条件を以下のように変更する．

$$S_{\mathrm{SCR}} \geq 5[\mathrm{m}^2] \tag{6.17}$$

前項の解を初期解として，スペースプレーンは3つのエンジンATR，SCR，ROCを4段で使用すると仮定し，図6.4に示す問題設定で最適化計算を始めた．しかし，得られた最適解は以下を満たしていた．

$$t_0^{\mathrm{SCR}} = t_f^{\mathrm{SCR}} \tag{6.18}$$

すなわち，SCRを単独で使用しない解となったため，問題設定を図6.5に示すエンジンを3段で使用する設定に変更して計算をし直した．軌道計画問題の離散化要素数はATR区間を95個，SCR+ROC区間を45個，ROC区間を60個とした．

当然，この場合の最適解のSCRインテーク面積S_{SCR}は最小値の5 [m^2]になる．得られた最適解の性能諸元，重量，形状をSCRなしの最適解と比較して表6.3，表6.4，図6.20に示す．図6.21，6.22に揚力係数と抗力係数，図6.23～6.31に飛行履歴を示す．

飛行履歴によると，SCRを単独で使用することはなく，マッハ6でATRからSCRに切り替えると同時にROCも点火する．このとき，推力が最大推力制限を越えないようにしている．その後，作動限界に達するよりも少し早めのマッハ11.15でSCRを停止する．

表6.3，表6.4によると，SCRを使用するとペイロード重量が2.56 [ton]ほど減少してしまう．まず，SCRを使用すると燃料であるLH_2消費量は増加するが，ROCの負担が減るためLOXの消費量は減少し，結果的にLH_2とLOXを合わせた推進剤消費重量は減少する．もし推進剤消費重量を最小化する目的関数に変更するならば，SCRを使用する最適解が得られると思われる．また，LOXの密度1.1496 [g/cm^3]に対して，LH_2の密度は0.071 [g/cm^3]と非常に小さい．そこで，LOXの減少，LH_2の増加は全タンク容積の増加を招き，それにつれて図6.20から分かるように機体が大きくなり胴体の重量は微増する．その一方で，ROCの推力が減少して最大荷重倍数が30 %以上減少するため，翼の重量は20 %近く軽くなる．エンジン重量に及ぼす影響を考えてみると，SCRのためにATRとROCは必要な

推力が減り両者の重量は減少する．しかし，SCR の重量はインテーク面積 1 [m²]当たり 1 [ton]と重いため，エンジン全体の重量は 3.56 [ton]と大幅に増加する．したがって，SCR を使用すると，推進剤消費重量の減少に比べ構造重量の増加が大きく，結果的にペイロード重量が減少する．この結果から概算すると，SCR の存在がペイロード重量を増加させる作用を持つためには，その重量をおよそ半分以下にしなければならない．

本論文における問題設定ではエンジンモデルが非常に簡単であり，また飛行中のエンジンのスロットルは常に全開状態にある．さらに，重量推算法，または目的関数の定義によっても結果が変わりうる可能性があるため，この結果から即座に今後のスペースプレーン開発において SCR は不要であるとは断言できず，さらなる検討が必要である．

表6.3 SCR 不使用の場合と使用の場合の諸元値

諸元		SCR なし	SCR あり	変化割合
胴体全長	l_4 [m]	63.48	70.95	11.8 %
ノーズ長さ	l_1 [m]	32.99	37.64	14.1 %
胴体高さ	$2a$ [m]	6.00	6.00	0.0 %
胴体幅	$2b$ [m]	6.36	6.01	-5.4 %
翼スパン	$2s$ [m]	20.04	19.91	-0.6 %
胴体体積	[m³]	1442.13	1513.99	4.9 %
LH_2 タンク体積	[m³]	851.53	906.01	6.4 %
LOX タンク体積	[m³]	157.96	153.78	-2.7 %
最大動圧	q_{max} [kPa]	100.0	100.0	0.0 %
最大荷重倍数	n_{LFmax} [G]	3.82	2.55	-33.2 %
最大推力	T_{max} [tonf]	226.6	226.2	-0.2 %
ATR インテーク面積	S_{ATR} [m²]	12.59	12.16	-3.4 %
SCR インテーク面積	S_{SCR} [m²]	0.00	5.00	—
ROC 推力	T_{ROC} [tonf]	226.6	153.3	-32.3 %
LH_2 重量	W_{LH2} [ton]	60.46	64.33	6.4 %
LOX 重量	W_{LOX} [ton]	181.50	176.70	-2.6 %
構造重量	$W_{structure}$ [ton]	63.30	66.67	5.4 %
付加重量	$W_{auxiliary}$ [ton]	8.50	8.61	1.2 %
最小必要重量	$W_{required}$ [ton]	313.75	316.31	0.8 %
ペイロード重量	$W_{payload}$ [ton]	-13.75	-16.31	—

表6.4 SCR 不使用の場合と使用の場合の重量内訳

機体構成要素	SCR なし [kg]	SCR あり [kg]	増加量 [kg]
body structure	16163	16891	728
wing	7397	6039	-1358
vertical fin	0	0	0
horizontal stabilizer	0	0	0
landing gear	5961	5961	0
thrust structure	2322	2319	-3
ATR engine	14476	13986	-490
SCR engine	0	5000	5000
ROC engine	2946	1994	-952
liquid hydrogen tank	7229	7692	463
liquid oxygen tank	3163	3079	-84
liquid hydrogen system	822	837	15
liquid oxygen system	349	363	14
pressurization system	2467	2513	46
avionics	1900	1900	0
aerodynamic control system	1667	1699	32
electrical system	1635	1682	47
hydraulic/pneumatic system	387	419	32
OMS	220	220	0
OMS tank	22	22	0
OMS propellant	1500	1500	0
crew, crew life support	1171	1171	0
胴体/翼	29521	28891	-630
エンジン	19744	23299	3555
タンク/配管系	14030	14484	454
制御/動力系	5589	5700	111
その他	2913	2913	0
LH_2	60459	64327	3868
LOX	181498	176697	-4801
構造重量	63295	66674	3379
付加重量	8502	8613	111
推進剤重量	241957	241024	-933
ペイロード重量	-13754	-16311	-2557

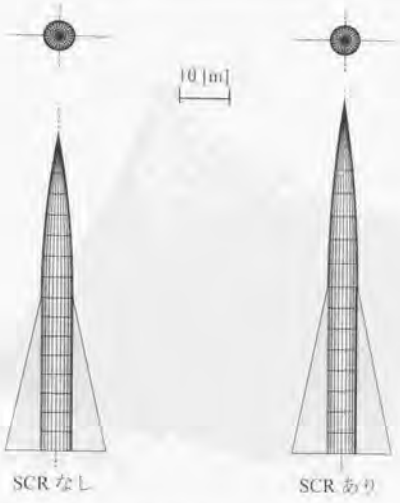


図6.20　SCR 不使用の場合と使用の場合の機体形状

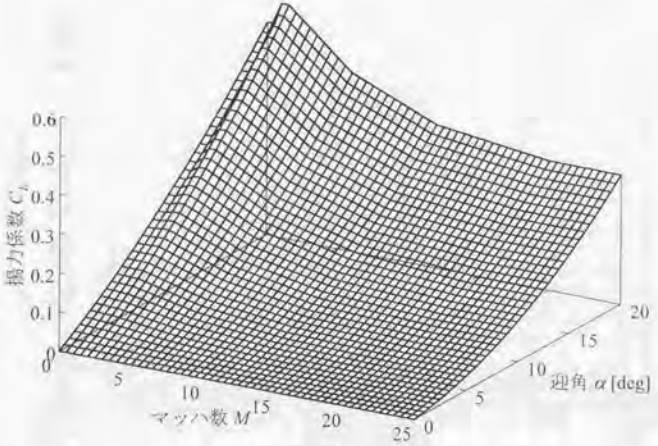


図6.21　揚力係数

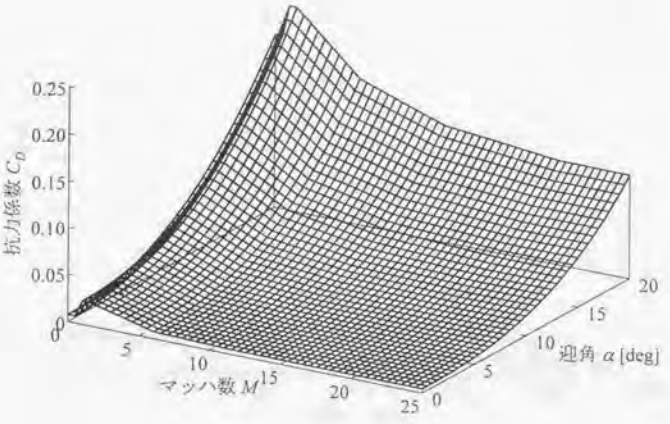


図6.22 抗力係数

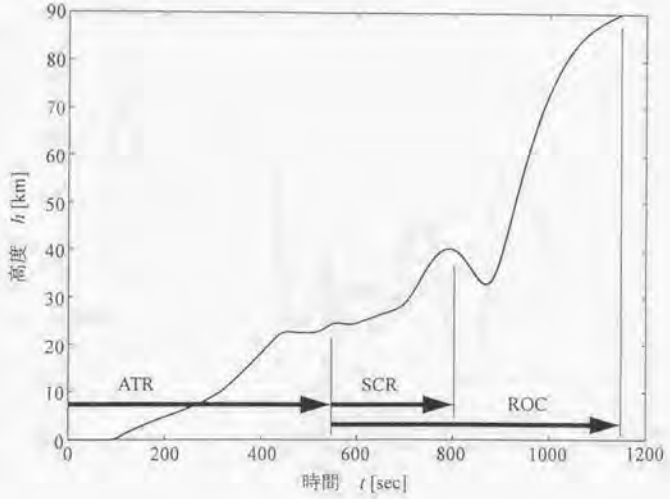


図6.23 高度の時間履歴

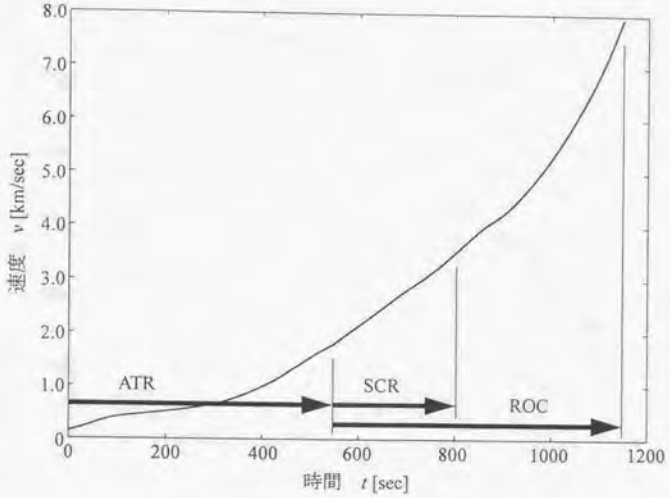


図6.24 速度の時間履歴

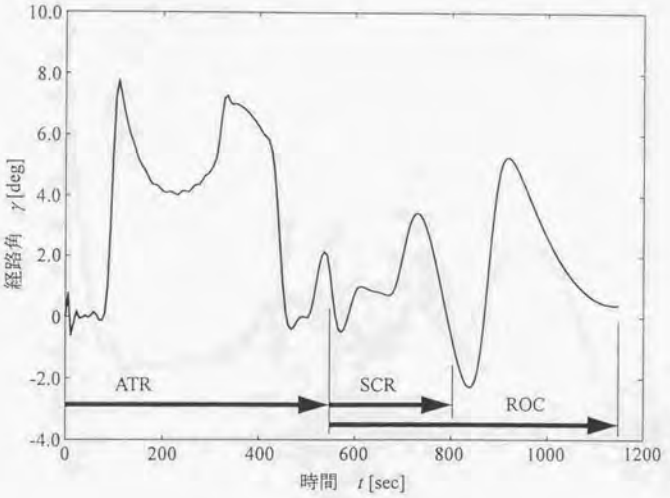


図6.25 経路角の時間履歴

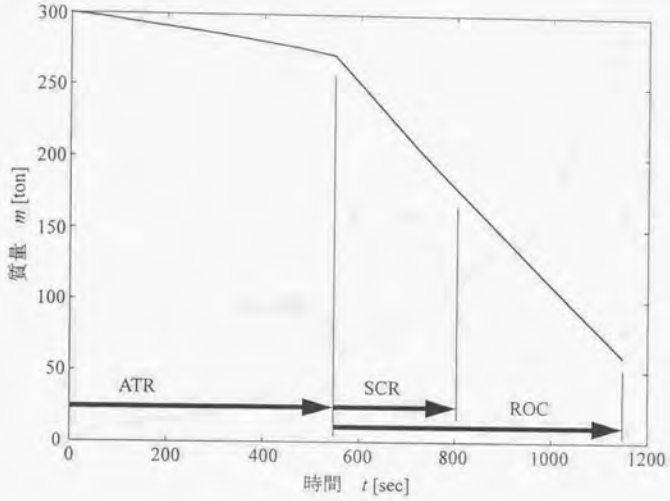


図6.26 質量の時間履歴

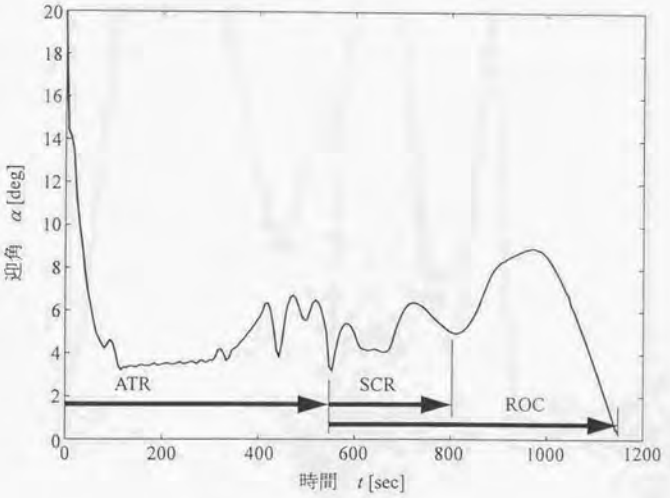


図6.27 迎角の時間履歴

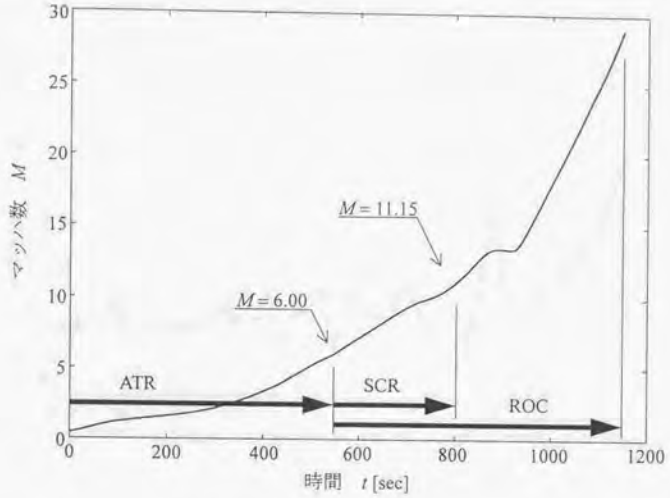


図6.28 マッハ数の時間履歴

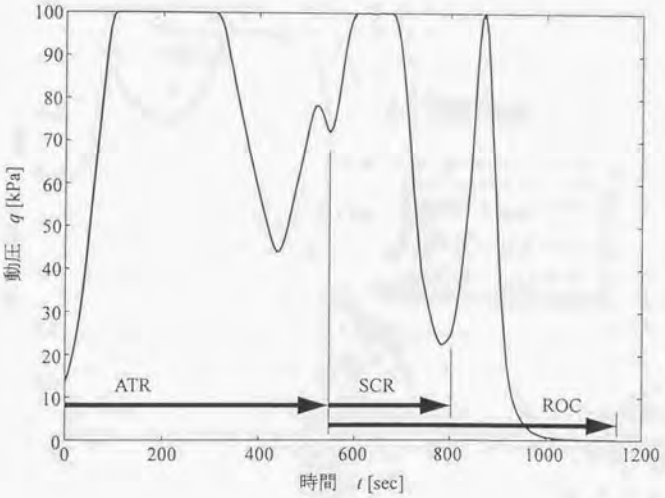


図6.29 動圧の時間履歴

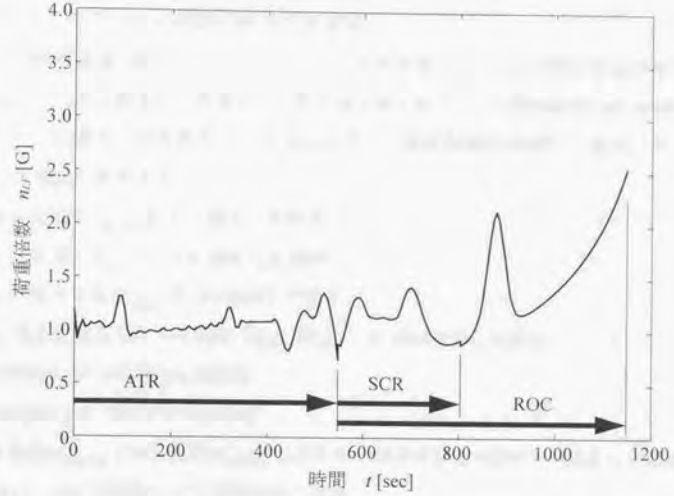


図6.30 荷重倍数の時間履歴

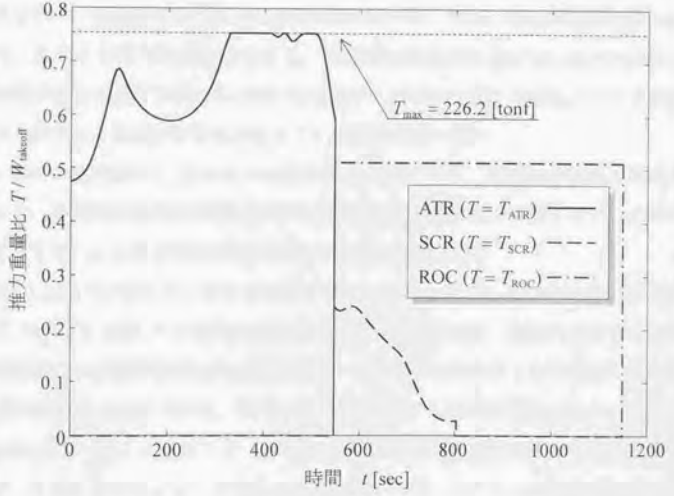


図6.31 推力重量比の時間履歴

6.3.3 スペースプレーン実現に向けた改善量

本項では離陸重量 300 [ton]のスペースプレーンを実現するためには構成要素がどの程度改善されることが必要なのか考察する．前々項で示されたように離陸重量 300 [ton]のスペースプレーンの最小必要重量は 313.75 [ton]となり，離陸重量を 5%近く上回る．そこで，次の 8 つの場合を想定する．

(1) ATR の比推力 $I_{SP_{ATR}}$ が 10%増加した場合．

(2) ROC の比推力 $I_{SP_{ROC}}$ が 10%増加した場合．

(3) ATR の推力係数 $C_{T_{ATR}}$ が 10%増加した場合．

(4) ROC 推進器重量当たりの推力（T_{ROC}/W_{ROC}）が 10%増加した場合．

(5) 揚力係数 C_L が 10%増加した場合．

(6) 抗力係数 C_D が 10%減少した場合．

(7) 構造重量 $W_{structure}$ と付加重量 $W_{auxiliary}$ に含まれる要素の重量が各々10%減少した場合．

(8) 上の(1)～(7)の改善がすべて実現できた場合．

(1)～(4)は推進技術の進展，(5)，(6)は空力技術の進展，(7)は構造と材料技術の進展を想定している．以上の 8 つの場合それぞれについて，前々項の最適解を初期解として最適化計算を再び行い，最適解がどのように変化するか調べる．なお，前項で考察を行った SCR は比推力，推力ともに 10%増加させても，最適解に変化が見られないことを確認している．最適解が変化する，すなわち SCR を使用する解が現れるためには，エンジン重量に対する推力が大きく増加する必要があることは前項で述べた．

表6.5，6.6に結果を示す．これらの表を見てわかるように，各要素は複雑に関連し合っているため，各要素を個別に改善しても影響は関連した要素だけに留まらず，すべての諸元を変化させる．この事は統合的最適化の重要性を示している．

表6.5には上の(1)～(4)のエンジンに関する技術の進展を考慮した場合の解が示されている．まず，ROC と ATR それぞれの比推力を 10%増加した場合，比推力の大きさは飛行時の単位時間当たりの消費推進剤重量に反比例するため，比推力が 1.1 倍になれば推進剤重量は約 10%減少するはずである．特に，ROC が消費する推進剤重量は最小必要重量の多くを占めるため，ROC の比推力を 10%増加できれば，最小必要重量は離陸重量 300[ton]を下回り，それだけでペイロード重量を正にして，スペースプレーンの実現性を増すことができると思われた．しかし，得られた結果によると，ROC の比推力が 10%増しても ROC が消費する推進剤重量は 3.8%，ATR の場合は 7.7%しか減少せず，思うほど実現性は改善

されない．その理由は，比推力の変化は飛行中の機体重量を変化させ，最適上昇飛行経路を変化させるためである．例えば，ROC の比推力が 10%増加すると，ROC 使用時の機体重量が増すため，ROC 推力，ROC の飛行時間が数%増加して，ROC による推進剤重量は 3.8%しか減少しない．なお，同じ 10%の改善であっても，上の想定(3)，(4)にある推力の改善は比推力の改善ほど大きく最小必要重量を減少させない．

表6.6には揚力係数と抗力係数の改善後の最適解が示されている．揚力係数を増加させられれば，翼スパンは小さくなり，翼と翼を支持する構造の重量が減少する．一方，抗力係数の減少が可能となれば，機体全長を短くし，そのかわり胴体幅を大きくして構造重量を 2 [ton]も減少させることが可能となる．加えて，エンジンを小さくして，抗力の影響が大きい ATR 区間で消費される推進剤重量を少なくすることができる．その結果，2.4 [ton]もの最小必要重量の改善が可能である．揚力と抗力の大きさを単純に比較することはできないが，結果の数値から判断すると，抗力軽減のほうが重要であることが分かる．

また，表6.6には構造重量と付加重量に含まれる構成要素の重量をそれぞれ 10%減少させることができた場合の最適解も示されている．構成要素の重量軽減にほぼ見合うだけの重量の改善が最適解に見られる．

すべての要因を改善させることができた場合，ペイロード重量は正の値 8.94 [ton]にすることができる．この場合の構造重量を除く重量の変化量は，個々の想定による変化量を単純に加えた量にほぼ等しい．一方，構造重量はこれまでに想定した要因による改善量を考慮すれば，1.5 [ton]ほどより軽減できるはずである．これは要因どうしの結びつきが構造重量に強い非線形性で影響を及ぼしていることを意味している．

本項ではスペースプレーンの構成要素を一律に 10%改善するとしたが，各専門分野の事情によってその困難さは異なるであろう．どの要因にどれだけの量の改善を要求するかを判断するためには，まず，スペースプレーンの目標とするペイロード重量を定める．そして，各要因に対してどれ程の努力量（開発コスト，時間など）でどれほどの改善量が将来的に見込めるかを定量的に示す．その上で，各努力量の釣り合いを考えつつ，各要因ごとの改善量を決定すべきであろう．

表6.5 現在の技術水準による最適解と改善後の最適解（その1）

諸元	現状	$I_{SP\,ATR}$	$I_{SP\,ROC}$	$C_{T\,ATR}$	T_{ROC}/W_{ROC}
機体全長 l_4 [m]	63.48	63.68	64.51	63.03	64.33
ノーズ長さ l_1 [m]	32.99	33.35	33.30	33.87	34.08
胴体高さ $2a$ [m]	6.00	6.00	6.00	6.00	6.00
胴体幅 $2b$ [m]	6.36	6.18	6.05	6.46	6.31
翼スパン $2s$ [m]	20.04	20.18	19.97	20.36	20.20
最大動圧 q_{max} [kPa]	100.0	100.0	100.0	100.0	100.0
最大荷重倍数 n_{LFmax} [G]	3.82	3.80	3.43	3.86	3.84
最大推力 T_{max} [tonf]	226.6	227.2	233.8	229.3	228.2
ATR インテーク面積 S_{ATR} [m^2]	12.59	12.14	12.35	11.58	12.47
SCR インテーク面積 S_{SCR} [m^2]	0.00	0.00	0.00	0.00	0.00
ROC 推力 T_{ROC} [tonf]	226.6	227.2	233.8	229.3	228.2
ATR 飛行時間 [sec]	566	584	550	553	564
SCR 飛行時間 [sec]	0	0	0	0	0
ROC 飛行時間 [sec]	418	420	428	413	415
ATR の推進剤 [ton]	30.21	27.89	29.56	30.07	30.17
ROC の推進剤 [ton]	211.75	213.58	203.62	211.78	211.69
構造重量 $W_{structure}$ [ton]	63.30	62.33	62.09	62.06	63.06
付加重量 $W_{auxiliary}$ [ton]	8.50	8.50	8.50	8.49	8.51
最小必要重量 $W_{required}$ [ton]	313.75	312.31	303.79	312.40	313.43
ペイロード重量 $W_{payload}$ [ton]	-13.75	-12.31	-3.79	-12.40	-13.43

表6.6 現在の技術水準による最適解と改善後の最適解（その2）

諸元	現状	C_L	C_D	構造/付加重量	すべて改善
機体全長 l_4 [m]	63.48	64.15	62.40	64.63	60.37
ノーズ長さ l_1 [m]	32.99	35.07	33.67	33.91	32.99
胴体高さ $2a$ [m]	6.00	6.00	6.00	6.00	6.00
胴体幅 $2b$ [m]	6.36	6.37	6.49	6.24	6.27
翼スパン $2s$ [m]	20.04	19.30	20.47	20.08	19.53
最大動圧 q_{max} [kPa]	100.0	100.0	100.0	100.0	100.0
最大荷重倍数 $n_{f/max}$ [G]	3.82	3.72	3.68	3.85	3.33
最大推力 T_{max} [tonf]	226.6	221.6	219.7	229.3	233.0
ATR インテーク面積 S_{ATR} [m^2]	12.59	12.32	11.58	12.50	10.43
SCR インテーク面積 S_{SCR} [m^2]	0.00	0.00	0.00	0.00	0.00
ROC 推力 T_{ROC} [tonf]	226.6	221.6	219.7	229.3	233.0
ATR 飛行時間 [sec]	566	570	585	557	553
SCR 飛行時間 [sec]	0	0	0	0	0
ROC 飛行時間 [sec]	418	427	432	413	434
ATR の推進剤 [ton]	30.21	29.93	29.55	29.91	25.87
ROC の推進剤 [ton]	211.75	211.68	212.06	211.85	205.70
構造重量 $W_{structure}$ [ton]	63.30	62.30	61.28	57.12	51.91
付加重量 $W_{auxiliary}$ [ton]	8.50	8.49	8.48	7.66	7.59
最小必要重量 $W_{required}$ [ton]	313.75	312.40	311.38	306.53	291.06
ペイロード重量 $W_{payload}$ [ton]	-13.75	-12.40	-11.38	-6.53	8.94

6.3.4 離陸重量とペイロード重量

本章ではこれまで 300 [ton]としていた離陸重量を増減させてみる．表6.7に離陸重量を 100 [ton]から 500 [ton]まで変化させたときの諸元値を示す．図6.32は機体サイズの変化を，図6.33～6.39は諸元値の変化をグラフで示している．

ペイロード重量は離陸重量から各構成要素の重量を引いた量であるから，もし離陸重量と構成要素の重量が比例関係にあれば，離陸重量が増加するにつれてペイロード重量は 0 [ton]よりさらに小さくなるはずである．しかし，重量算出プログラム WAATS によると，多くの構成要素の重量は離陸重量の増加と比べて滑らかに増加するため，離陸重量が増すとペイロード重量は逆に 0 [ton]に近付く．よって，離陸重量を増せばスペースプレーンの実現性は増す．しかし，離陸重量を 100 [ton]から 500 [ton]に変化させてみてもペイロード重量は約 3.6 [ton]ほどしか増加しない．500 [ton]以上の離陸重量でも計算を試みたが，ペイロード重量はほとんど変化しない．よって，WAATS の適用可能範囲内である常識的な離陸重量において，ペイロード重量を正の値，すなわちスペースプレーンがペイロードを積んで飛行できる最適解は得られなかった．離陸重量に対する最小必要重量の比によると，スペースプレーンを実現するためには数%から 15%ほどの重量軽減に努めなければならない．もっとも，今回用いている各種データは現在入手できるデータを様々な仮定の下で使用しているので，今後の技術進歩等により十分改善可能であり直ちにスペースプレーンの実現を否定するものではない．本章の目的はあくまでもスペースプレーンを実現するための指標を示すことにある．

いずれの離陸重量においても，得られた最適解の傾向は 300 [ton]において既述した傾向とまったく同じである．翼重量の増加を押さえるため，翼の大きさは離陸重量が増すにつれて徐々に変化しなくなる．そのかわり胴体幅を広げて揚力を得るようになる．また，翼面積，ATR の大きさは飛行に最小限の大きさであり，SCR を使用しないことで構造重量を小さくしている．さらに，離陸重量が増加すると離陸から軌道投入までの時間は増すが，飛行履歴は図6.11～6.19に示した離陸重量 300 [ton]とほとんど同じである．

なお，図6.37～6.39によると，離陸重量と各種機体諸元の間に成り立つ関係則がわかる．ATR の推力はインテーク面積に比例し，それは離陸重量に比例するはずである．実際，ATR インテーク面積の離陸重量に対する増加量はほぼ一定である．同様に ROC 推力も離陸重量に比例している．また，胴体と翼の平面積は揚力にほぼ比例するため，離陸重量との間に比例関係が成り立つ．これらの関係は得られた最適解の妥当性を示している．

表6.7 離陸重量 100～500[ton]での諸元値の最適解

離陸重量 $W_{takeoff}$ [ton]	100	200	300	400	500
胴体全長 l_4 [m]	30.53	49.80	63.48	69.29	71.93
ノーズ長さ l_1 [m]	25.38	32.00	32.99	38.19	43.05
胴体高さ $2a$ [m]	6.00	6.00	6.00	6.00	6.00
胴体幅 $2b$ [m]	6.00	6.00	6.36	7.90	9.81
翼スパン $2s$ [m]	15.21	18.42	20.04	23.98	28.26
平面積 [m^2]	156.4	345.6	542.5	697.5	832.1
ROC 停止までの時間 t_f^{ROC} [sec]	825	920	983	1008	1022
最大動圧 q_{max} [kPa]	99.3	100.0	100.0	100.0	100.0
最大荷重倍数 n_{LFmax} [G]	4.00	4.00	3.82	3.78	3.72
最大推力 T_{max} [tonf]	123.8	170.3	226.6	299.4	372.4
ATR インテーク面積 S_{ATR} [m^2]	8.48	10.10	12.59	15.65	19.01
SCR インテーク面積 S_{SCR} [m^2]	0.00	0.00	0.00	0.00	0.00
ROC 推力 T_{ROC} [tonf]	74.8	156.4	226.6	299.4	372.4
LH_2 重量 W_{LH2} [ton]	22.69	41.64	60.46	80.41	100.60
LOX 重量 W_{LOX} [ton]	59.02	120.15	181.50	242.13	302.67
構造重量 $W_{structure}$ [ton]	27.51	45.11	63.30	80.88	97.85
付加重量 $W_{auxiliary}$ [ton]	5.75	7.28	8.50	9.46	10.28
ペイロード重量 $W_{payload}$ [ton]	-14.96	-14.17	-13.75	-12.88	-11.39
最小必要重量 $W_{required}$ [ton]	114.96	214.17	313.75	412.88	511.39
最小必要重量／離陸重量	1.150	1.071	1.046	1.032	1.023

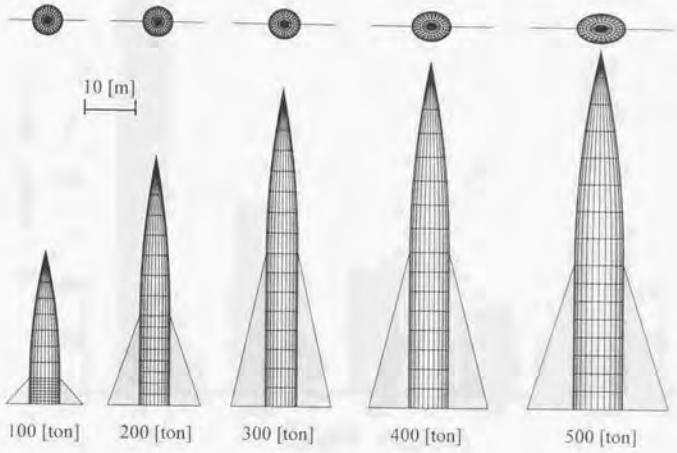


図6.32 離陸重量と機体形状

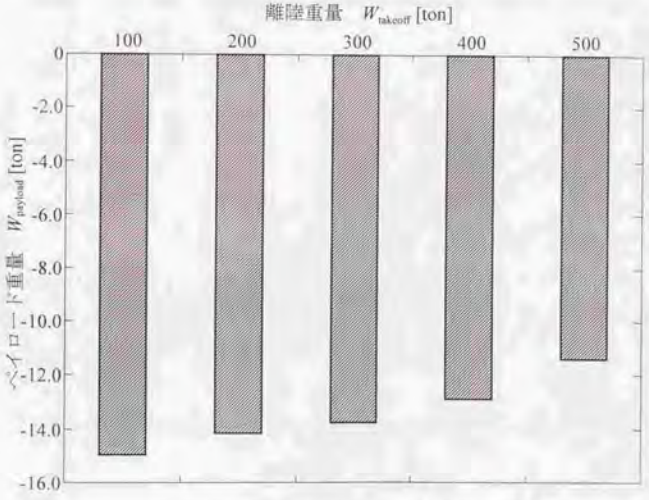


図6.33 離陸重量とペイロード重量

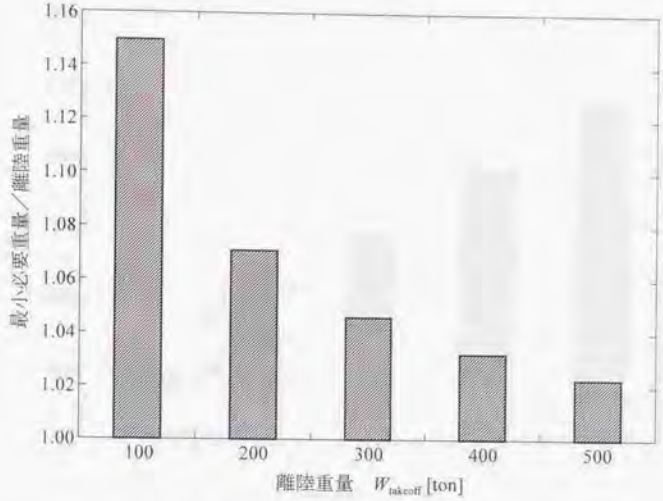


図6.34 離陸重量と最小必要重量/離陸重量

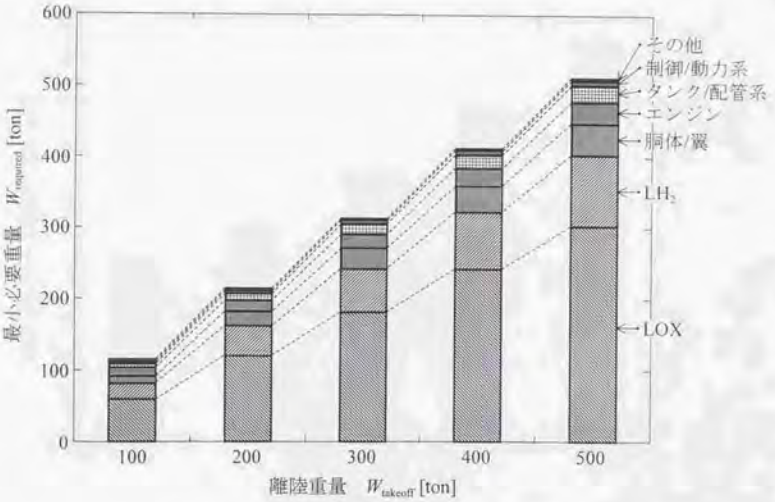


図6.35 離陸重量と最小必要重量内訳

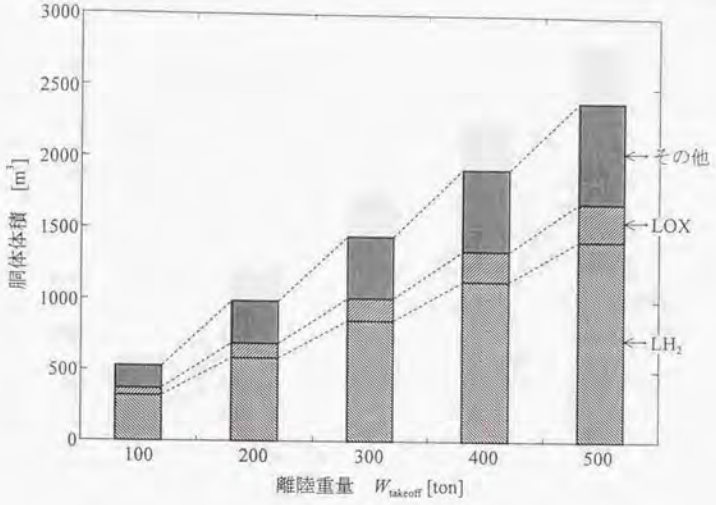


図6.36 離陸重量と胴体体積内訳

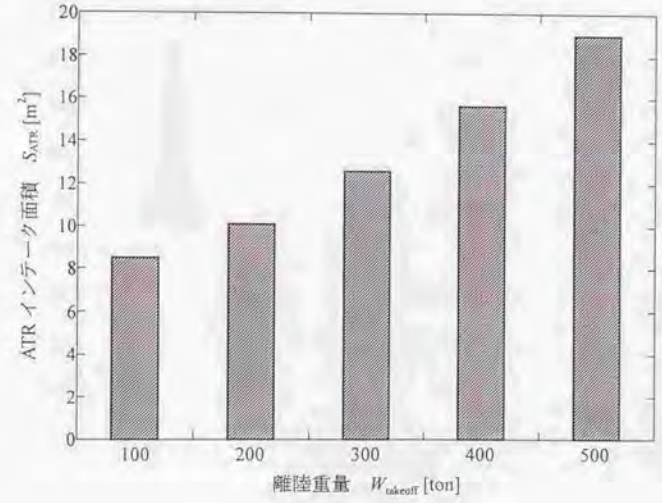


図6.37 離陸重量と ATR インテーク面積

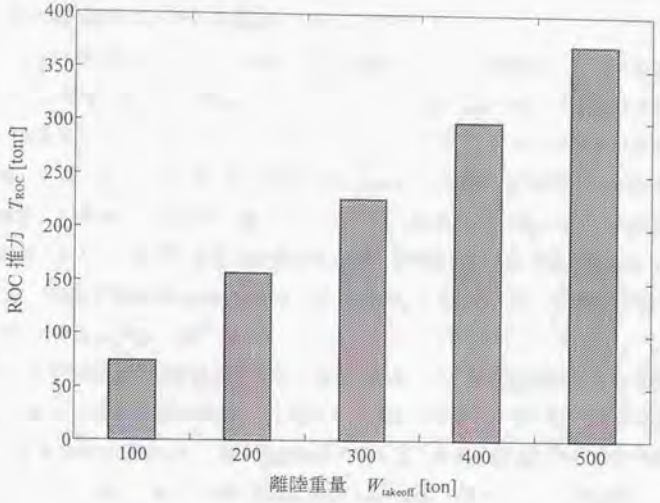


図6.38 離陸重量と ROC 推力

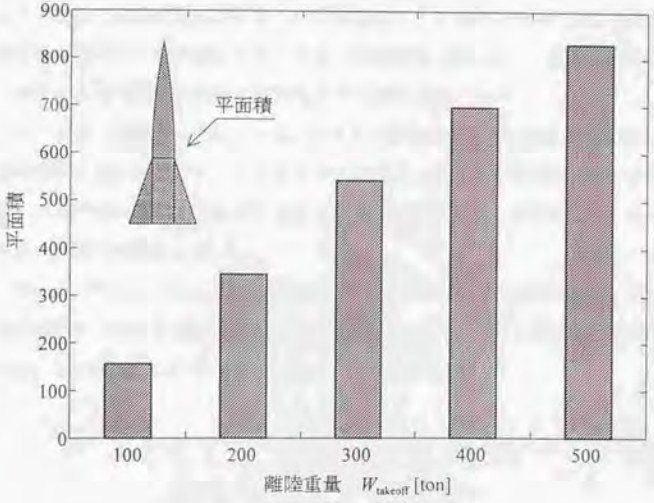


図6.39 離陸重量と平面積

6.4 並列最適化法と全体最適化法の比較

本節では並列最適化法と全ての制約条件，変数を一括して最適化する全体最適化法を比較する．本問題の変数の総数は 1000 を超え，また制約条件の数もそれとほぼ同数存在するが，全体最適化法で解くことは不可能ではない．ただし，かなりのメモリ量と計算時間を必要とする．ここでは ROC 停止高度を 90 [km]から 100 [km]に変更し，前節の最適解を初期解として最適化計算を再び実行する．それ以上計算を繰り返しても，制約条件誤差が 10^{-6} 未満であるか，全体最適化では繰り返し計算 1 回あたり，並列最適化では outer loop 1 回あたりの変数の最も大きな変化量が 10^{-8} を上回ることが無くなった時点で解は収束したと判断して計算を停止した．

表6.8に 1 計算機当たりに必要なメモリ量，図6.40に計算実行時間に対する目的関数値の推移，図6.41に計算実行時間に対する制約条件誤差の推移を示す．制約条件誤差の意味については第 3 章で説明した．全体最適化法の繰り返し計算 1 回当たりの計算時間はおよそ 202 [sec] であるのに対し，並列最適化法の outer loop 1 回当たりの計算時間が平均 127 [sec]ですむ．ここでの計算では並列最適化法のほうが早く計算が終了し最適解を得ることができた．ただし，計算時間，繰り返し回数は収束判定，最適化プログラム内のパラメータ値によってかなり左右される．一方，1 計算機当たりに必要なメモリ量は並列最適化法のほうが変数の数に応じて少なくすむ．本章の問題設定では，4 つの部分問題によって変数の数に偏りがあるため，それほど並列化による効率が良くない．

ところで，本章の問題は，図6.4～6.6によると，機体設計問題と空力解析問題の全ての変数は接続条件に含まれるため，サブシステム間のカップリングは非常に強い．このようなカップリングの強い問題に対しても安定して最適解を得ることができるという本解法の特徴を本章の問題でも確認した．

また，図6.40，6.41によると，本問題においても，並列最適化法によって，全体最適化法より目的関数値と制約条件誤差が優れた最適解が得られた．第 5 章で現れたのと同様な現象である．その理由については第 5 章で述べた通りである．

表6.8 1 プロセス当たりに必要なメモリ量

最適化手法	必要メモリ量 [kB]
全体最適化法	96184
並列最適化法	56392

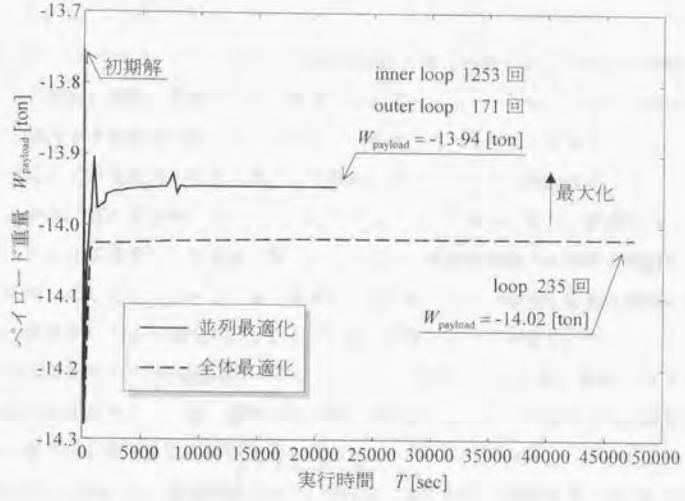


図6.40 計算実行時間に対する目的関数値（ペイロード重量$W_{payload}$）の推移

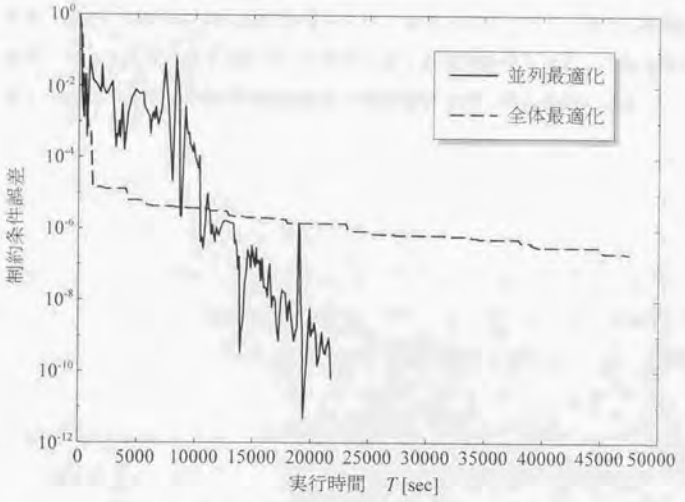


図6.41 計算実行時間に対する制約条件誤差の推移

6.5 まとめ

本章で取り上げたスペースプレーンの機体設計と飛行経路の統合的最適化問題は本論文中で最も求解が困難な問題である．問題設定の違いにより 15 題の最適化問題を解いたがどれも満足する最適解が得られた．ただし，前章までと異なり，初期解によっては繰り返し計算途中で部分最適化問題が実行不可能解に陥り，そこから解が改善しないことがあった．しかし，解が最適解に近付くにつれそのようなことはなくなり，最適解近くでは前節で示したように安定して最適解が得られる．また，複雑な問題では簡単な問題に比べて並列化効率は良くない．しかし，全く最適解の推定ができない複雑な最適化問題の場合，より良い解が得られる可能性があることが本章の問題によって確認された．

また本章で得られた最適解はスペースプレーンを実現する上で最小限満たされなければならない条件を示している．既存の航空機，宇宙機をもとにして算出された機体を構成する上で最小限必要な重量は離陸重量を上回っている．スペースプレーンを実現するためには数%から 10%以上の重量軽減とエンジン性能，空力特性の改善を図らなければならない．機体のサイズは推進剤タンクの容積を大きく上回らず，できるだけ小さいほうが良く，抗力を軽減するために細長い胴体が良い．翼は離陸条件から必要な揚力を得る最小限の大きさとする．また，エンジンの 1 つであるエアターボラムジェットエンジンも必要以上に大きくせず，スクラムジェットエンジンは使用しない最適解が得られた．これらは後にスペースプレーンについてより精巧な問題設定で概念設計を行う際の指針となる．

第7章
結論

7.1 結論

7.1.1 並列最適化法の提案

本論文では複合領域最適化問題の一種である航空機の機体設計と飛行経路の統合的最適化問題に対して並列最適化法を提案し，幾つかの問題に適用した．

統合的最適化の対象になるシステムは，システムを定義する変数と制約条件が莫大な数になる大規模システムであり，統合的最適化問題は大規模最適化問題の一種といえる．そこで大規模システムを複数のサブシステムに分割し，各サブシステムから小規模な部分最適化問題を定義する．複数の部分最適化問題はサブシステムレベルで並列的に解かれ，その後，得られた最適解がシステム全体の最適解となるようにシステムレベルで部分問題の設定を変更し，再びサブシステムレベルの計算を繰り返す．この手法は分割解法と呼ばれる．分割解法の利点として，並列的に最適化が行われることで計算負荷が軽減できる点，各専門領域ごとにサブシステムを分割すると，すでに存在する個々の領域の解析法，最適化手法が使用できる点が挙げられる．従来の分割解法では，元の大規模最適化問題を直接解いて解を得るのと比較して収束性が悪く，多くのシステム解析と最適化計算が必要であった．また，問題設定，初期解によらずに確実に解を得ることができるかどうかを意味するロバスト性が悪いという欠点があった．特に分割されたサブシステムどうしのカップリングが強いと収束性が悪化することが報告されている．

一方，大規模最適化問題を並列処理する方法として，複合領域最適化とは別の観点から並列最適化法の研究が進められている．一般に，並列計算は並列化するレベルにより，大きく「問題の並列化」と「解法の並列化」に分けられる．「問題の並列化」とは複合領

域最適化法に代表される分割解法である．一方，「解法の並列化」は問題の解法に含まれる個々の過程を並列化する方法であり，元の解法の性質を残したままで優れた並列化効率を得ることができることから並列計算法の研究の中心となっている．本論文では従来の解法に見られる「問題の並列化」ではなく「解法の並列化」に注目した．並列化する解法として，現時点で最も優れた最適化手法といわれている逐次 2 次計画法を考えた．逐次 2 次計画法で最も大きな計算時間を要するのが変数の更新量とラグランジュ乗数量を求める大規模線形方程式を解く過程である．そこでこの方程式を Jacobi 法の繰り返し計算で並列的に解くこととし，さらにその収束性を Aitken の加速法で高めることにした．一方で，この並列最適化法に対する見方を変えると，並列に進められる個々の計算はある最適化問題を解く計算に等しいことが分かった．よって，1 つの大規模最適化問題を解くために小規模な複数の最適化問題を並列的に解くこととなり，「解法の並列化」に注目して得られたこの並列最適化法は，実は「問題の並列化」による分割解法と等価になる．したがって，こうして得られた並列最適化法は，優れた特性をもつ逐次 2 次計画法を並列化したことになるため，カップリングが強い問題に対しても良好な収束性とロバスト性をもち，さらに並列化効率に優れた分割解法である．加えて，並列最適化法には興味深い特性があることが判明した．最適化計算は反復計算により収束した解を最適解と見なすのであるが，並列最適化法による収束解は，最適化問題を一括して解く方法で得られた解に比べて優れた解であることが本論文の例題では分かった．これは並列最適化法の特出すべき特徴である．

7.1.2 最適化問題への適用結果

本論文において提案された並列最適化法は 5 個の例題に適用された．そのうち 3 題は軌道最適化問題，2 題は機体設計と飛行経路の統合的最適化問題である．

一般に，複雑な軌道最適化問題の最適解を得ようとする場合，または精度の良い最適解を得ようとする場合，変数と制約条件の数が非常に多い大規模最適化問題を解くことが必要となる．そこで，制御時間を幾つかの区間に分割し，1 つの区間を 1 つのサブシステムと見なして部分最適化問題を構成することにした．適用された軌道最適化問題のうち，2 題は理論解を求めることができる簡単な問題であり，計算された数値解が理論解と一致することを確認し，提案された並列最適化法が優れた並列化効率をもつことを確認した．

また，機体設計と飛行経路の統合的最適化問題の簡単な問題例として，紙飛行機のサイズと投げ方の最適化問題を考えた．並列最適化法は有効に機能し，得られた最適解も妥

当性が認められた．

スペースプレーンの最適化問題として，まず上昇軌道最適化問題を扱った．従来よりほぼ同じ問題設定で最適解が求められていたが，本論文では最適解の精度を上げるために変数と制約条件の数を増やした．得られた最適な飛行経路は従来求められていた最適解の傾向と一致した．またここでは上述したように並列最適化法による最適解が従来の解法で求められた最適解よりも優れていることを確認した．

最後に，スペースプレーンの機体形状と上昇飛行経路の統合的最適化問題を考え，並列最適化法の有効性の確認とスペースプレーンを実現するための技術レベルの見積もりを行った．最適化問題は機体設計分野，空力解析分野，軌道計画分野の3つの専門領域からなり部分最適化問題は 4〜6 個で構成される．得られた最適解によるとスペースプレーンを実現するためにはその離陸重量に応じて数%から 10%以上の重量軽減，エンジン性能と空力性能の向上を図らなければならない．機体のサイズは推進剤のタンク容積から最小の大きさが決まるが可能な限り小さく，また抗力を低減するために細長い胴体が良い．翼の大きさは離陸時の状態から決まる最小限の大きさ，エンジンの1つであるエアターボラムジェットエンジンも飛行できる最小の大きさにするのが良い．また，エアブリージングエンジンの1つであるスクラムジェットエンジンは極超音速で作動可能な推進器として現在研究が進められているが，エンジン単体が重いこと，もし使用するならば比推力が小さいため推進剤である液体水素の容量が増加して機体が大きくなることから，必要性が認められなかった．最適なエンジンの組み合わせについては今後の検討が必要である．また，この問題でも本論文で提案する並列最適化法の有効性が確認された．本問題の最適化を通して，カップリングの強い領域を統合的最適化する重要性が認識された．

7.2 今後の課題

本論文で得られた成果から，さらに今後の課題として以下の点が挙げられる．

(1) 統合的最適化法及び複合領域最適化法に関して

個々の領域に応じた解析法，最適化法を柔軟に組み込め，少ない解析回数で最適解（あるいは最適解に近い解）を得ることができるアルゴリズムを開発し，最終的には最適化支援ツールとして完成させなければならない．それに向けて次の事項を考える必要がある．

- ラグランジュ乗数が求まらない解法を部分最適化法に適用することを考える.
- 組み合わせ最適化問題, 多目的最適化問題など, 臨機応変な問題設定を可能にする.
- 解析回数を減らすために統計学的手法を活用する.
- ロバスト性を有する最適解を得ることが可能な最適化法を考慮する.

(2) 統合的最適化法及び複合領域最適化法の応用, 特に再使用型宇宙往還機に関して.

各分野の専門家の意見を取り入れ, 最新の航空機, 宇宙機から現在と将来の技術水準を見極め, より正確に問題設定を行う必要がある. 将来的には, 概念設計に留まらず, 実機の設計に複合領域最適化法を適用しなければならない.

- より精巧なエンジンモデル, 空力解析法, 構造解析法を組み入れる.
- 目的関数を運用コスト等に変更してみる.
- スペースプレーン以外の宇宙往還機の可能性も視野に入れる.

補遺A

非線形計画法

本章では本論文中で議論されてきた非線形計画問題の定義とその数値解法について簡単にまとめる．非線形計画法自体は，歴史的にも古くから研究が進められており，現在，テキスト[41, 42, 43, 44]，数値計算パッケージ[44, 45]が多数存在する．ここでは必須事項のみをまとめる．

A.1 非線形計画法と最適性の条件

本論文で取り扱う非線形計画問題（nonlinear programming problem）は，一般に次のように表される．

変数を$x \in R^n$とする関数$f: R^n \to R$，$g_E: R^n \to R^{m_E}$，$g_I: R^n \to R^{m_I}$の定義のもとで，

$$g_E(x) = 0 \tag{A.1a}$$

$$g_I(x) \leq 0 \tag{A.1b}$$

を満たし，関数$f(x)$を最小にする変数xを求めよ．

本論文中では，この種の最適化問題は以下のように簡潔に表す．

$$\text{variable} \quad x \tag{A.2a}$$

$$\text{minimize} \quad f(x) \tag{A.2b}$$

$$\text{subject to} \quad g_E(x) = 0 \tag{A.2c}$$

$$g_I(x) \leq 0 \tag{A.2d}$$

ここで，本論文中で使用される用語を定義しておく．関数f，g_E，g_Iをそれぞれ目的関数，等式制約関数，不等式制約関数といい，式(A.2c)，(A.2d)をそれぞれ等式制約条件，

不等式制約条件という．また，等式及び不等式制約関数のことをまとめて制約関数，等式及び不等式制約条件のことを単に制約条件と呼ぶことがある．制約条件を満たす解を実行可能解（feasible solutions）といい，さらにそのような解の集合を実行可能領域という．また，ある制約条件に最適解を代入したところ，その制約条件が等号で満たされたとき，制約条件は活性（active）であるいう．上の問題では変数 x に連続的な値をとるため，非線形計画問題は連続的最適化（continuous optimization）問題とも呼ばれる．特に，目的関数と制約関数が 1 次式で与えられる問題は線形計画（linear programming）問題，非線形計画問題のうち，目的関数が 2 次式で，等式，不等式制約条件が 1 次関数で与えられる問題は 2 次計画（quadratic programming: QP）問題といわれる．また，変数が離散的な値をとる問題は組合せ最適化（combinatorial optimization）問題と呼ばれ，連続的最適化問題と合わせて，最適化問題は数理計画（mathematical programming）問題と総称される．なお，非線形計画問題の最適解の性質などに関連する理論や具体的な最適解の計算手法は，一般に非線形計画法（nonlinear programming）と呼ばれる．

式(A.2a)～(A.2d)の定義のもとで，次の関数をラグランジュ（Lagrange）関数と呼ぶ．

$$L(x,\lambda_E,\lambda_I)=f(x)+\lambda_E{}^{\mathrm{T}}g_E(x)+\lambda_I{}^{\mathrm{T}}g_I(x) \tag{A.3}$$

ここに，$\lambda_E\in R^{m_E}$，$\lambda_I\in R^{m_I}$ はそれぞれ等式制約条件，不等式制約条件に対するラグランジュ乗数ベクトルという．$x^*\in R^n$ を非線形計画問題（式(A.2a)～(A.2d)）に対する最適解とすると，次に示す Kuhn-Tucker 条件が与えられる．

関数 f，g_E，g_I がすべて C^1 級であるとき，x^* が最適解ならば，

$$\nabla_x L(x^*,\lambda_E{}^*,\lambda_I{}^*)=0 \tag{A.4a}$$

$$g_E(x^*)=0 \tag{A.4b}$$

$$g_I(x^*)\le 0 \tag{A.4c}$$

$$\lambda_{I_i}{}^* g_{I_i}(x^*)=0 \qquad (i=1,2,\ldots,m_I) \tag{A.4d}$$

$$\lambda_I{}^*\ge 0 \tag{A.4e}$$

を満たす $\lambda_E{}^*$，$\lambda_I{}^*$ が存在する．ここで，$\lambda_{I_i}{}^*$，g_{I_i} はラグランジュ乗数 $\lambda_I{}^*$ と不等式制約関数 g_I の第 i 成分である．

A.2 数値解法の分類

式(A.2a)～(A.2d)で表される制約付き最小化問題の数値解法は大きく分類すると，関数の微分を用いる方法と用いない方法に分けられ，前者は勾配法，後者は直接法と一般に呼ばれている．勾配法は，目的関数の値を下げるための情報として，現在の近似解のまわりの勾配を有効に使うことができるのがその利点である．一方，直接法では勾配情報は使えないので，多くの点での目的関数の値を総合的に用いて最適解を探索する．したがって，収束の速さは勾配法に比べて良くないのが普通である．しかし，多くの点での情報を総合するので，時間をかければ大域的最適解が得られるという利点を持つ．本論文では，より効率の良い方法を求めて，古くから研究が進められ，現在多くの数値計算パッケージが存在する勾配法に注目する．

関数の勾配を必要とする方法は，制約条件の扱いによってさらに分類される．ペナルティ関数（penalty function）法，拡張ラグランジュ乗数（augmented Lagrange multiplier）法に代表される解法は，制約付き最小化問題を適当に変換して，無制約最小化問題に置き換える方法（変換法）である．一方，可能方向（feasible direction）法，逐次線形計画（sequential linear programming）法，逐次2次計画（Sequential Quadratic Programming: SQP）法（準ニュートン法）に代表される解法は，制約条件を考慮しつつ目的関数を最小化する最適解を探す方法（非変換法）である．前者は，無制約最小化法のアルゴリズムがそのまま使え，手法の構成が簡単なことが長所である．しかし収束が遅いという欠点を持つ．他方後者において，現在，SQP法が有力視されている．

A.3 逐次2次計画法

逐次2次計画（SQP）法は1970年代後半にHan[46, 47]やPowell[48]らによって研究が進められ，現在では非線形計画問題の解法として，最も有効な手法といわれている．本論文でもこのSQP法に注目している．

A.3.1 アルゴリズム

SQP法は，近似解を繰り返し計算により最適解に収束させる解法である．以下にその手順を簡潔に示す．

Step 1.

初期解 $x^{(0)} \in R^n$ と正定値対称行列 $B^{(0)} \in R^{n\times n}$ を選び，$k=0$ とする．

Step 2.

以下の2次計画（QP）問題を解く．

$$\text{variable} \quad d^{(k)} \in R^n \tag{A.5a}$$

$$\text{minimize} \quad \nabla_x f(x^{(k)})d^{(k)} + \frac{1}{2}d^{(k)\mathrm{T}}B^{(k)}d^{(k)} \tag{A.5b}$$

$$\text{subject to} \quad g_E(x^{(k)}) + \nabla_x g_E(x^{(k)})d^{(k)} = 0 \tag{A.5c}$$

$$g_I(x^{(k)}) + \nabla_x g_I(x^{(k)})d^{(k)} \le 0 \tag{A.5d}$$

Step 3.

次に示すペナルティ関数を定義する．

$$F_r(x^{(k)}) \equiv f(x^{(k)}) + r\left\{\sum_{i=1}^{m_E}\left|g_{E_i}(x^{(k)})\right| + \sum_{i=1}^{m_I}\max\left[0, g_{I_i}(x^{(k)})\right]\right\} \tag{A.6}$$

ここで$r>0$は十分大きな値でありペナルティパラメータと呼ばれる．また，関数 g_{E_i}，g_{I_i} はベクトル関数 g_E，g_I の第 i 列のスカラー関数を指す．その上で，変数$0<\alpha^{(k)}\le 1$（ステップ幅）とする以下の1次元（直線）探索問題を解き，

$$\text{variable} \quad \alpha^{(k)} \tag{A.7a}$$

$$\text{minimize} \quad F_r(x^{(k)} + \alpha^{(k)}d^{(k)}) \tag{A.7b}$$

解を更新する．

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)}d^{(k)} \tag{A.8}$$

Step 4.

現在の解 $x^{(k+1)}$ が収束判定条件を満たせば終了．そうでなければ，行列 $B^{(k)}$ を $B^{(k+1)}$ に更新し，$k \leftarrow k+1$として，Step 2.へ戻る．

A.3.2 ヘッセ行列

前項のアルゴリズムに含まれる行列 $B^{(k)}$ について考えてみる．まず，QP問題（式(A.5a)～(A.5d)）のKuhn-Tucker条件を，ラグランジュ乗数 $\lambda_E^{(k+1)}$，$\lambda_I^{(k+1)}$ を導入して書く．

$$\left[\begin{array}{c|c|ccc} \boldsymbol{B}^{(k)} & \nabla_x g_E{}^{\mathrm{T}} & \nabla_x g_{I1}{}^{\mathrm{T}} & \cdots & \nabla_x g_{Im_I}{}^{\mathrm{T}} \\ \hline \nabla_x g_E & \mathbf{0} & \multicolumn{3}{c}{\mathbf{0}} \\ \hline \lambda_{I1}{}^{(k+1)}\nabla_x g_{I1} & & g_{I1} & \cdots & 0 \\ \vdots & \mathbf{0} & \vdots & \ddots & \vdots \\ \lambda_{Im_I}{}^{(k+1)}\nabla_x g_{Im_I} & & 0 & \cdots & g_{Im_I} \end{array}\right] \left[\begin{array}{c} \boldsymbol{d}^{(k)} \\ \lambda_E{}^{(k+1)} \\ \lambda_I{}^{(k+1)} \end{array}\right] = \left[\begin{array}{c} -\nabla_x f^{\mathrm{T}} \\ -g_E \\ \mathbf{0} \end{array}\right] \tag{A.9}$$

ところで，元の非線形計画問題の Kuhn-Tucker 条件（式(A.4a)～(A.4e)）は，未知数を x^*，$\lambda_E{}^*$，$\lambda_I{}^*$ とする非線形代数方程式と見なすことができる．それを Newton 法によって求めることを考えると，繰り返し計算 k 回目の変数値 $x^{(k)}$，$\lambda_E{}^{(k)}$，$\lambda_I{}^{(k)}$ から $k+1$ 回目の変数値 $x^{(k+1)}$，$\lambda_E{}^{(k+1)}$，$\lambda_I{}^{(k+1)}$ への更新公式は以下のようになるであろう．

$$\left[\begin{array}{c|c|ccc} \nabla_{xx}^2 L & \nabla_x g_E{}^{\mathrm{T}} & \nabla_x g_{I1}{}^{\mathrm{T}} & \cdots & \nabla_x g_{Im_I}{}^{\mathrm{T}} \\ \hline \nabla_x g_E & \mathbf{0} & \multicolumn{3}{c}{\mathbf{0}} \\ \hline \lambda_{I1}{}^{(k)}\nabla_x g_{I1} & & g_{I1} & \cdots & 0 \\ \vdots & \mathbf{0} & \vdots & \ddots & \vdots \\ \lambda_{Im_I}{}^{(k)}\nabla_x g_{Im_I} & & 0 & \cdots & g_{Im_I} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \\ \lambda_E{}^{(k+1)} \\ \lambda_I{}^{(k+1)} \end{array}\right] = \left[\begin{array}{c} -\nabla_x f^{\mathrm{T}} \\ -g_E \\ \mathbf{0} \end{array}\right] \tag{A.10}$$

式(A.5b)を一見したところでは，行列 $\boldsymbol{B}^{(k)}$ は目的関数 f の 2 階の偏微係数からなるヘッセ行列 $\nabla_{xx}^2 f$ のように見える．しかし，2 つの式(A.9)，(A.10)を見比べてみると，ラグランジュ関数 L のヘッセ行列 $\nabla_{xx}^2 L$ とすると，SQP 法に Newton 法の持つ収束性が期待できることが分かる．しかし，このヘッセ行列を実際に算出するのは困難が伴い，また常に本当のヘッセ行列が数値計算上望ましいとはいえない．例えば，行列 $\boldsymbol{B}^{(k)}$ が正定値であれば式(A.5a)～(A.5d)の QP 問題は凸 2 次計画（凸 QP）問題と呼ばれ，次項で説明する QP 問題の有力な解法，Goldfarb-Indnani 法を適用することができる．そこで，行列 $\boldsymbol{B}^{(k)}$ を $\nabla_{xx}^2 L$ の近似行列で代用させることにする．

Step 4.で $\boldsymbol{B}^{(k)}$ から $\boldsymbol{B}^{(k+1)}$ を逐次求めることとし，次の BFGS（Broydon-Fletcher-Goldfarb-Shanno）公式を用いる．まず，

$$\boldsymbol{p}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \tag{A.11a}$$

$$\boldsymbol{q}^{(k)} = \nabla^{\mathrm{T}} L(\boldsymbol{x}^{(k+1)}, \lambda_E{}^{(k+1)}, \lambda_I{}^{(k+1)}) - \nabla^{\mathrm{T}} L(\boldsymbol{x}^{(k)}, \lambda_E{}^{(k+1)}, \lambda_I{}^{(k+1)}) \tag{A.11b}$$

として，パラメータ

$$\theta = 1 \qquad ((\boldsymbol{p}^{(k)})^{\mathrm{T}} \boldsymbol{q}^{(k)} \geq 0.2 (\boldsymbol{p}^{(k)})^{\mathrm{T}} \boldsymbol{B}^{(k)} \boldsymbol{p}^{(k)} \text{のとき}) \tag{A.12a}$$

$$\theta=\frac{0.8(p^{(k)})^{\mathrm{T}}B^{(k)}p^{(k)}}{(p^{(k)})^{\mathrm{T}}B^{(k)}p^{(k)}-(p^{(k)})^{\mathrm{T}}q^{(k)}} \quad (\langle (p^{(k)})^{\mathrm{T}}q^{(k)}<0.2(p^{(k)})^{\mathrm{T}}B^{(k)}p^{(k)} \text{のとき}) \tag{A.12b}$$

を求め，このθを用いて$\tilde{q}^{(k)}$を

$$\tilde{q}^{(k)}=\theta q^{(k)}+(1-\theta)B^{(k)}p^{(k)} \tag{A.13}$$

のように定義する．そして，次の更新公式，

$$B^{(k+1)}=B^{(k)}+\frac{\tilde{q}^{(k)}(\tilde{q}^{(k)})^{\mathrm{T}}}{(\tilde{q}^{(k)})^{\mathrm{T}}p^{(k)}}-\frac{B^{(k)}p^{(k)}(p^{(k)})^{\mathrm{T}}B^{(k)}}{(p^{(k)})^{\mathrm{T}}B^{(k)}p^{(k)}} \tag{A.14a}$$

$$B^{(0)}=I \tag{A.14b}$$

を用いて行列$B^{(k+1)}$へ更新する．このようにして更新された$B^{(k+1)}$は$B^{(k)}$が正定値ならば常に正定値になることが容易に確認できる．その他にも様々な更新公式が提案されているが，いずれの公式も以下の準ニュートン条件（正割条件，セカント（secant）条件）と呼ばれる条件が必ず成り立つ．

$$B^{(k+1)}p^{(k)}=\tilde{q}^{(k)} \tag{A.15}$$

BFGS 公式において，$k=0$のとき$B^{(0)}$は単位行列が用いられる．このことは，SQP 法は，繰り返し計算開始時に初期解に対し，目的関数を線形近似する逐次線形計画法に等しい収束性を持つことを意味する．そして計算が進むうちに$B^{(k)}$が更新されていき，$B^{(k)}$自身がヘッセ行列$\nabla^2_{xx}L$に近付かなくても，それがある適当な部分空間の上で$\nabla^2_{xx}L$の良い近似になってゆくならば，最適解近傍で超1次収束性が期待できる．

A.3.3 凸2次計画問題の解法（Goldfarb-Indnani 法）

前項で述べた SQP 法は，一見すると単純な数値解法である．しかし，現実には Step 2. の凸 QP 問題を如何に解くのかという問題点に突き当たる．SQP 法は複雑な数値計算をすべて凸 QP 問題の解法に任せてしまっているのである．本論文では凸 QP 問題の解法に Goldfarb-Indnani（GI）法[49]を採用する．文献49では不等式制約条件のみをもつ凸 QP 問題が取り上げられたが，文献45では等式制約条件が存在する問題に拡張されている．以降は文献45に沿って GI 法を簡単にまとめる．

次のような凸 QP 問題を考える．

$$\text{variable} \quad x \in R^n \tag{A.16a}$$

$$\text{minimize} \quad c^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Gx \tag{A.16b}$$

$$\text{subject to} \quad a_i{}^{\mathrm{T}}x = b_i \qquad (i = 1,2,\ldots,m_e) \tag{A.16c}$$

$$a_i{}^{\mathrm{T}}x \le b_i \qquad (i = m_e+1,\ldots,m) \tag{A.16d}$$

ここで，$c \in R^n$，$a_i \in R^n$ であり，$G \in R^{n\times n}$ は正定値対称行列である．以下では，等式制約条件と不等式制約条件の添え字集合をそれぞれ

$$E = \{1,2,\ldots,m_e\}, \quad K = \{m_e+1,\ldots,m\} \tag{A.17}$$

と表すことにする．等式制約条件しか持たない凸 QP 問題を解くのは容易であるが，不等式制約条件を持つ凸 QP 問題では，どの不等式制約条件が活性であるか分からないため解くことが難しい．そこで，不等式制約条件の幾つかが活性であると仮定して解を求め，その解が制約条件を満たしていなければ，活性と仮定した制約条件を変更して再び解を求め直す過程を繰り返し，最終的に全ての制約条件を満たす解を求める．

Step 1.

まず，GI 法では幾つかの制約条件が活性であると仮定して解を求める．それらの制約条件の添え字集合を J とし，得られた解を J-最適解と呼ぶ．計算を始めるときには，集合 J を等式制約条件の添え字集合 E とする（$J = E$）のが良いであろう．すなわち，次のような凸 QP 問題を解くことになる．

$$\text{variable} \quad x \tag{A.18a}$$

$$\text{minimize} \quad c^{\mathrm{T}}x + \frac{1}{2}x^{\mathrm{T}}Gx \tag{A.18b}$$

$$\text{subject to} \quad A_J{}^{\mathrm{T}}x = b_J \tag{A.18c}$$

ここで，行列 A_J はベクトル a_i（$i \in J$）を列とするような $n \times |J|$ 行列である．この問題の解の変数を x_J，等式制約条件のラグランジュ乗数を λ_J とすると，Kuhn-Tucker 条件は

$$c + Gx_J + A_J\lambda_J = 0 \tag{A.19a}$$

$$A_J{}^{\mathrm{T}}x_J - b_J = 0 \tag{A.19b}$$

と表せ，後に数式を展開するときの簡便化のため，

$$A_J^{+} = (A_J^{\top} G^{-1} A_J)^{-1} A_J^{\top} G^{-1} \tag{A.20a}$$

$$H_J = G^{-1}(I - A_J A_J^{+}) \tag{A.20b}$$

と新たな行列を定義しておくと，変数 x_J とラグランジュ乗数 λ_J は

$$x_J = (A_J^{+})^{\top} b_J - H_J c \tag{A.21a}$$

$$\lambda_J = -(A_J^{\top} G^{-1} A_J) b_J - A_J^{+} c \tag{A.21b}$$

のように求められる．

Step 2.

次に集合 J に含まれていない不等式制約条件の変数に式(A.21a)の値を代入してみる．もしすべての不等式制約条件が満たされるのであれば，式(A.21a)，(A.21b)の変数 x_J，λ_J が最適解であるため，計算を停止する．そうでなければ，不等式制約条件を満たしていない条件を1つ選び，集合 J に加えることにする．ここではその条件の添え字を s とする．すなわち，

$$c_s = a_s^{\top} x_J - b_s > 0 \tag{A.22}$$

である．満たされない不等式制約条件が複数ある場合，アルゴリズムの計算効率を考慮すれば最も満たされていない条件を選ぶべきであろう．

Step 3.

次に，この添え字 s の不等式制約条件が活性となり，満足されるように変数とラグランジュ乗数を x_J，λ_J から変更する．その変更量を Δx_J，$\Delta\lambda_J$，添え字 s の制約条件に対するラグランジュ乗数を λ_s とすると，添え字 s の制約条件を含めた Kuhn-Tucker 条件，

$$c + G(x_J + \Delta x_J) + A_J(\lambda_J + \Delta\lambda_J) + a_s\lambda_s = 0 \tag{A.23a}$$

$$A_J^{\top}(x_J + \Delta x_J) - b_J = 0 \tag{A.23b}$$

$$a_s^{\top}(x_J + \Delta x_J) - b_s = 0 \tag{A.23c}$$

が新たに満たされなければならない．変更量 Δx_J，$\Delta\lambda_J$ はこれら式(A.23a)～(A.23c)に式(A.19a)～(A.19b)，(A.22)を代入することで，

$$\lambda_s = \frac{c_s}{a_s^{\mathrm{T}} H_J a_s} \tag{A.24a}$$

$$\Delta x_J = -H_J a_s \lambda_s \tag{A.24b}$$

$$\Delta \lambda_J = -A_J^{+} a_s \lambda_s \tag{A.24c}$$

と求められる．なお，$H_J a_s \neq 0$ とし，そうでない場合はStep 4.に進む．

式(A.24a)～(A.24c)が求まれば，集合 J に新たに添え字 s を加え，変数 x_J，λ_J を $x_J + \Delta x_J$，$\lambda_J + \Delta \lambda_J$ に変更すれば良いと思われる．しかしながら，これでは集合 J に不等式制約条件が含まれていた場合（$J \cap K \neq \phi$），これらの条件は変更後も活性のままであると仮定していることになりその保証はない．$\lambda_{J \cap K} \geq 0$ であるが，集合 J の変更後に，ある添え字 i（$i \in J \cap K$）の不等式制約が活性でなくなると $\lambda_i + \Delta \lambda_i < 0$ になる．そこで，集合 J に含まれる不等式制約条件の各々について，ラグランジュ乗数 $\lambda_i + \Delta \lambda_i$ が 0 になるラグランジュ乗数 λ_s を改めて λ_{si} として

$$\lambda_i + \Delta \lambda_i = \lambda_i - (A_J^{+})_i a_s \lambda_{si} = 0 \qquad (i \in J \cap K) \tag{A.25}$$

より求め，その中で最も小さい値 λ_s'

$$\lambda_s' = \min\left\{ \lambda_{si} = \frac{\lambda_i}{(A_J^{+})_i a_s} \;\middle|\; (A_J^{+})_i a_s > 0 \right\} \tag{A.26}$$

が式(A.24a)の λ_s より小さいか調べる．なお $(A_J^{+})_i$ は行列 A_J^{+} の i 行目からなる行ベクトルである．また，同時に λ_s' をとる制約条件の添え字を k とする．ここで，2つの場合に分けられる．

(1) 式(A.26)の λ_s' が存在しないか，$\lambda_s < \lambda_s'$ の場合．

この場合，変数 x_J，λ_J を $x_J + \Delta x_J$，$\lambda_J + \Delta \lambda_J$ に変更しても，集合 J に含まれる不等式制約はすべて活性のままである．よって，

$$x_J \leftarrow x_J + \Delta x_J \tag{A.27a}$$

$$\lambda_J \leftarrow \lambda_J + \Delta \lambda_J \tag{A.27b}$$

とし，さらに $J \leftarrow J \cap \{s\}$ として λ_J に λ_s を加える．この後，Step 2.に戻り，再び集合 J に含まれていない不等式制約条件を評価し直す．

(2) $\lambda_s > \lambda_s'$の場合.

この場合，変数$\boldsymbol{x}_J$，$\boldsymbol{\lambda}_J$を$\boldsymbol{x}_J + \Delta\boldsymbol{x}_J$，$\boldsymbol{\lambda}_J + \Delta\boldsymbol{\lambda}_J$に変更すると，集合$J$に含まれる添え字$k$の不等式制約を含めた少なくとも1つの制約が活性でなくなる．よって，

$$\boldsymbol{x}_J \leftarrow \boldsymbol{x}_J - \boldsymbol{H}_J \boldsymbol{a}_s \lambda_s' \tag{A.28a}$$

$$\boldsymbol{\lambda}_J \leftarrow \boldsymbol{\lambda}_J - \boldsymbol{A}_J^{+} \boldsymbol{a}_s \lambda_s' \tag{A.28b}$$

とし，さらに$J \leftarrow J - \{k\}$として$\boldsymbol{\lambda}_J$からλ_kを除く．この後，Step 3.に戻り，再び添え字sの不等式制約条件を活性にするような変数$\boldsymbol{x}_J$，ラグランジュ乗数$\boldsymbol{\lambda}_J$の変更量を$\Delta\boldsymbol{x}_J$，$\Delta\boldsymbol{\lambda}_J$を求め直す．

Step 4.

Step 3.では$\boldsymbol{H}_J \boldsymbol{a}_s \neq \boldsymbol{0}$とした．$\boldsymbol{H}_J \boldsymbol{a}_s = \boldsymbol{0}$であるならば，行列$\boldsymbol{H}_J$の定義（式(A.20b)）より，

$$\boldsymbol{a}_s = \boldsymbol{A}_J \boldsymbol{r}_J \qquad (\boldsymbol{r}_J \equiv \boldsymbol{A}_J^{+} \boldsymbol{a}_s) \tag{A.29}$$

となる．つまり，ベクトル$\boldsymbol{a}_s$はベクトル$\boldsymbol{a}_i$（$i \in J$）の1次結合で表され，これは添え字sの不等式制約条件が集合Jに含まれる制約条件と独立でないことを意味している．もし等式制約条件の1つが他の等式制約条件と独立でないならば実行可能解が存在しない．しかし，不等式制約条件の1つが他の制約条件と独立でなくても実行可能解が存在する場合がある．よって$\boldsymbol{H}_J \boldsymbol{a}_s = \boldsymbol{0}$のとき，実行可能解がない場合とある場合に分けられる．

(1) 実行可能でない場合の条件.

J-最適解$\boldsymbol{x}_J$から解を$\Delta\boldsymbol{x}_J$だけ変化させて，添え字sの制約条件を満たすようにする．

$$\boldsymbol{A}_{J\cap E}^{\mathrm{T}} (\boldsymbol{x}_J + \Delta\boldsymbol{x}_J) - \boldsymbol{b}_{J\cap E} = \boldsymbol{0} \tag{A.30a}$$

$$\boldsymbol{A}_{J\cap K}^{\mathrm{T}} (\boldsymbol{x}_J + \Delta\boldsymbol{x}_J) - \boldsymbol{b}_{J\cap K} \leq \boldsymbol{0} \tag{A.30b}$$

$$\boldsymbol{a}_s^{\mathrm{T}} (\boldsymbol{x}_J + \Delta\boldsymbol{x}_J) - b_s \leq 0 \tag{A.30c}$$

式(A.19b)を代入すれば，

$$\boldsymbol{A}_{J\cap E}^{\mathrm{T}} \Delta\boldsymbol{x}_J = \boldsymbol{0} \tag{A.31a}$$

$$\boldsymbol{A}_{J\cap K}^{\mathrm{T}} \Delta\boldsymbol{x}_J \leq \boldsymbol{0} \tag{A.31b}$$

が成り立つ．一方，式(A.30c)は式(A.22)を考慮すると，

$$a_s^{\mathrm{T}}\Delta x_J \le -\left(a_s^{\mathrm{T}}x_J - b_J\right) < 0 \tag{A.32}$$

が成り立つ．また，式(A.29)，(A.31a)，(A.31b)から$a_s^{\mathrm{T}}\Delta x_J$は

$$a_s^{\mathrm{T}}\Delta x_J = r_J^{\mathrm{T}}A_J^{\mathrm{T}}\Delta x_J = r_{J\cap K}^{\mathrm{T}}A_{J\cap K}^{\mathrm{T}}\Delta x_J \tag{A.33}$$

と変形される．ここで式(A.31b)を考慮すると，$r_{J\cap K} \le 0$のとき式(A.32)が成り立たなくなる．すなわち，$r_{J\cap K} \le 0$が成り立つとき，添え字集合$J\cup\{s\}$の制約条件のもとで問題は解を持たない．これは元の問題が実行可能でないことを意味する．

(2) 実行可能である場合．

添え字集合Jから1つの添え字$k\in J$を取り除いて添え字集合$\bar{J}=J-\{k\}$とする．このとき，どのような添え字kを選ぶべきか考える．

式(A.29)は

$$a_k = \frac{1}{r_k}\left(a_s - \sum_{i\in\bar{J}} r_i a_i\right) \tag{A.34}$$

と変形できることから，

$$\begin{aligned} A_J\lambda_J &= \lambda_k a_k + \sum_{i\in\bar{J}}\lambda_i a_i \\ &= \frac{\lambda_k}{r_k}a_s + \sum_{i\in\bar{J}}\left(\lambda_i - \frac{\lambda_k}{r_k}r_i\right)a_i \end{aligned} \tag{A.35}$$

となるため，

$$\bar{\lambda}_i = \lambda_i - \frac{\lambda_k}{r_k}r_i \qquad (i\in\bar{J}) \tag{A.36a}$$

$$\bar{\lambda}_s = \frac{\lambda_k}{r_k} \tag{A.36b}$$

とおき，$\bar{J}=\bar{J}\cup\{s\}$とすれば，式(A.19a)，(A.35)，(A.36a)，(A.36b)より

$$c + Gx_J + A_{\bar{J}}\bar{\lambda}_{\bar{J}} = 0 \tag{A.37}$$

が成り立つ．そこで，添え字kに

$$\frac{\lambda_k}{r_k} = \min\left\{\frac{\lambda_i}{r_i} \;\middle|\; r_i > 0,\ i \in J \cap K\right\} \tag{A.38}$$

であるような k を選べば，式(A.36a)，(A.36b)より，

$$\bar{\lambda}_{\bar{J}\cap K} \geq \mathbf{0} \tag{A.39}$$

が満たされる．ここで，

$$\bar{\lambda}_k = 0 \tag{A.40}$$

であることから，添え字 k の不等式制約条件は活性でなくなり，添え字集合 J の中で添え字 k 以外の不等式制約条件は活性のままで，添え字 k を取り除くことができる．しかし，添え字 s の不等式制約条件は満たされていないため，$J \leftarrow \bar{J}$，$\lambda_i \leftarrow \bar{\lambda}_i$ $(i \in \bar{J})$ として，再び Step 3.に戻る．

以上が GI 法の要約である．このアルゴリズムを効率的に実行するためには，行列に対する数値計算法に配慮が必要である．これらに関しては文献45を参照されたい．

A.3.4 1次元探索

A.3.1項のアルゴリズム内の Step 3.にある 1 次元探索で用いるペナルティ関数は正確なペナルティ関数（exact penalty function）と呼ばれる．ペナルティパラメータ r が，最適解におけるラグランジュ乗数 λ_E，λ_I の成分である λ_{Ei}，λ_{Ii} と比べて，

$$r \geq \max\left(\left|\lambda_{Ei}\right| \,\middle|\, i=1,2,\ldots,m_E\right),\quad r \geq \max\left(\left|\lambda_{Ii}\right| \,\middle|\, i=1,2,\ldots,m_I\right) \tag{A.41}$$

を満たせば，元の非線形計画問題は関数 F_r を目的関数とする制約なしの最適化問題と等価であると考えることができる．ただし，SQP 法における 1 次元探索では，現在の近似解よりペナルティ関数が減少するような解が得られれば充分であるから，必ずしも厳密な最小化を行う必要はなく，適切なステップ幅 $\alpha^{(k)}$ をできるだけ少ない計算量で見つけることが重要になってくる．ここでは，SQP 法の大域的収束性を保証することも考え，変数 α を $\alpha=1$ より徐々に小さくしていき，

$$F_r(x^{(k)} + \alpha d^{(k)}) < F_r(x^{(k)}) + \beta\alpha\left\{\overline{F}_r(x^{(k)}, d^{(k)}) - F_r(x^{(k)})\right\} \tag{A.42a}$$

$$\overline{F}_r(x,d) \equiv f(x) + \nabla_x f(x)d + \frac{1}{2}dB^{(k)}d + r\left\{\sum_{i=1}^{m_E}\left|g_{E_i}(x) + \nabla_x g_{E_i}(x)^{\mathrm{T}}d\right| + \sum_{i=1}^{m_I}\max\left[0, g_{I_i}(x) + \nabla_x g_{I_i}(x)^{\mathrm{T}}d\right]\right\} \tag{A.42b}$$

を満たすようなできるだけ大きなαをステップ幅$\alpha^{(k)}$とする．ただし，βは$0<\beta<1$であるパラメータである．本論文では，変数αは1より0.5倍の比率で小さくしていくことにし，$\beta=0.01$とした．

ところで，式(A.41)により，ペナルティ関数のペナルティパラメータはラグランジュ乗数の絶対値より大きくならなければならないことを述べた．その理由を説明しておく．

次の問題を考える．

$$\text{variable} \quad x \tag{A.43a}$$

$$\text{minimize} \quad f(x) \tag{A.43b}$$

$$\text{subject to} \quad c(x)=0 \tag{A.43c}$$

なお，最適化を行う変数xはベクトル，目的関数f，制約関数cはスカラーとする．次に，ペナルティパラメータrを用いてペナルティ関数を

$$F_r(x) \equiv f(x) + r|c(x)| \tag{A.44}$$

と定義する．制約条件を満たしつつ，ペナルティ関数F_rの最小値をとる変数値をx^*とする．つまり，

$$c(x^*)=0 \tag{A.45}$$

が成り立つ．式(A.45)により変数x^*で関数F_rは微分不可能であることに注意する．

いま，変数xを最適値x^*から制約関数を増加させるような微小変動Δxを考えて$x^*+\Delta x$とする．すなわち，

$$\begin{aligned} c(x^*+\Delta x) &\approx c(x^*) + \nabla_x c(x^*)\Delta x \\ &= \nabla_x c(x^*)\Delta x \geq 0 \end{aligned} \tag{A.46}$$

とする微小変動Δxを考える．このとき，ペナルティ関数値F_rは増加するため，

$$\begin{aligned} F_r(x^* + \Delta x) - F_r(x^*) &= \left\{ f(x^* + \Delta x) + r\left|c(x^* + \Delta x)\right| \right\} - \left\{ f(x^*) + rc(x^*) \right\} \\ &\approx \nabla_x f(x^*)\Delta x + r\nabla_x c(x^*)\Delta x \geq 0 \end{aligned} \tag{A.47}$$

が成り立つ．これから

$$r \geq -\frac{\nabla_x f(x^*)\Delta x}{\nabla_x c(x^*)\Delta x} \tag{A.48}$$

が求められる．

一方，変数 x を最適値 x^* から $x^* - \Delta x$ と変動させると，今度は

$$\begin{aligned} c(x^* - \Delta x) &\approx c(x^*) + \nabla_x c(x^*)(-\Delta x) \\ &= -\nabla_x c(x^*)\Delta x \leq 0 \end{aligned} \tag{A.49}$$

から

$$\begin{aligned} F_r(x^* - \Delta x) - F_r(x^*) &= \left\{ f(x^* - \Delta x) + r\left|c(x^* - \Delta x)\right| \right\} - \left\{ f(x^*) + rc(x^*) \right\} \\ &\approx -\nabla_x f(x^*)\Delta x - r\nabla_x c(x^*)\Delta x \geq 0 \end{aligned} \tag{A.50}$$

が成り立ち，

$$r \geq \frac{\nabla_x f(x^*)\Delta x}{\nabla_x c(x^*)\Delta x} \tag{A.51}$$

が求められる．よって，変数 x^* が，制約条件を満たし，ペナルティ関数 F_r を最小とするならば，式(A.48)かつ式(A.51)より

$$r \geq \left| \frac{\nabla_x f(x^*)\Delta x}{\nabla_x c(x^*)\Delta x} \right| \tag{A.52}$$

が成立する．

さて，ラグランジュ乗数を λ とすると，ラグランジュ関数は

$$L(x, \lambda) = f(x) + \lambda c(x) \tag{A.53}$$

と定義される．式(A.43a)，(A.43b)からなる問題の最適解を x^*，ラグランジュ乗数の最適

解をλ^*とすると，

$$\nabla_x L(x^*,\lambda^*)=\nabla_x f(x^*)+\lambda^*\nabla_x c(x^*)=0 \tag{A.54}$$

が成り立たなければならない．ここで，上で導入した変数の微小変動Δxを考えると，

$$\lambda=-\frac{\nabla_x f(x^*)\Delta x}{\nabla_x c(x^*)\Delta x} \tag{A.55}$$

が成り立つ．したがって式(A.52)，(A.55)より，以下が成り立つとき，ペナルティ関数を最小とする変数値は，元の最適化問題の最適解と一致する．

$$r\geq|\lambda| \tag{A.56}$$

複数の制約条件が存在する場合，不等式制約条件が存在する場合も同様に説明できる．

A.3.5 収束性と収束速度

アルゴリズムが大域的収束性をもつとは，任意の点x_0を初期解として生成される点列$\{x_k\}$において，最適解，もしくはKuhn-Tucker条件を満たす解x^*が有限回の繰り返しで得られるか，あるいは無限点列$\{x_k\}$の極限点として得られることである．また，初期解が解x^*の十分近傍に限定されるとき，そのアルゴリズムは局所的収束性をもつという．

収束の速さの尺度として，点列$\{x_k\}$がx^*にqステップのp次収束（$p\geq1$）するとは，$\beta\in(0,\infty)$，$k_0\geq1$，$q\geq1$に対して，

$$\left\|x_{k+q}-x^*\right\|\leq\beta\left\|x_k-x^*\right\|^p \tag{A.57}$$

が任意の$k\geq k_0$に対して成り立つことである．またqステップの超1次収束性とは，

$$\lim_{k\to\infty}\frac{\left\|x_{k+q}-x^*\right\|}{\left\|x_k-x^*\right\|}=0 \tag{A.58}$$

が成り立つことである．特に断らなければ$q=1$である．

SQP法は1次元探索を行うことで大域的収束性が保証されており，局所的超1次収束性をもつとされている．

補遺B

代表的な複合領域最適化法

ここでは第 1 章で概要のみを紹介した代表的な複合領域最適化法をより詳細にまとめておく．

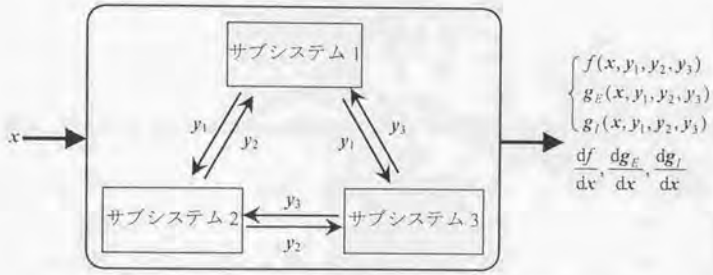
B.1 All-at-Once（All-in-One）法

All-at-Once 法[7]はその名前の通りシステム全体の変数を全て同時に最適化する．各サブシステムで行われるのは目的関数値，制約関数値の計算とその感度計算だけである．よって，各サブシステムで最適化計算は行われない．

図B.1にあるような 3 つのサブシステムからなるシステムを考える．最適化を行う変数を $\boldsymbol{x}$，またサブシステム i から他のサブシステムに向かうデータを変数 $\boldsymbol{y}_i$ とする．よって次のように定義できる．

$$\boldsymbol{y}_1 = \boldsymbol{y}_1(\boldsymbol{x}, \boldsymbol{y}_2, \boldsymbol{y}_3) \tag{B.1a}$$

$$\boldsymbol{y}_2 = \boldsymbol{y}_2(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_3) \tag{B.1b}$$



図B.1 3 つのカップリングしたサブシステムからなるシステム

$$y_3 = y_3(x, y_1, y_2) \tag{B.1c}$$

ある与えられた変数値 x に対して目的関数 f，等式制約関数 g_E，不等式制約関数 g_I の値を求めるためには，サブシステム間で繰り返し計算を行い，変数 y

$$y = \left(y_1^{\mathrm{T}}, y_2^{\mathrm{T}}, y_3^{\mathrm{T}}\right)^{\mathrm{T}} \tag{B.2}$$

を求めてから関数値を得ることができる．一方，システムの感度解析とは各関数の変数 x に対する微分であるが，

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x,y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} \tag{B.3a}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} g_E(x,y) = \frac{\partial g_E}{\partial x} + \frac{\partial g_E}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} \tag{B.3b}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} g_I(x,y) = \frac{\partial g_I}{\partial x} + \frac{\partial g_I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} \tag{B.3c}$$

より，微分値 $\mathrm{d}y/\mathrm{d}x$ が必要となる．ここで，サブシステムの感度情報を

$$J_{ij} = \left[\frac{\partial y_i}{\partial y_j}\right] \tag{B.4a}$$

$$J_{i0} = \left[\frac{\partial y_i}{\partial x}\right] \qquad (i, j = 1,2,3,\ \ i \neq j) \tag{B.4b}$$

とすると，次の式が成り立つ．

$$\frac{\mathrm{d}y_i}{\mathrm{d}x} = J_{i0} + \sum_{\substack{j=1 \\ (j \neq i)}}^{3} J_{ij}\frac{\mathrm{d}y_j}{\mathrm{d}x} \tag{B.5}$$

ここから，以下の global sensitivity equations（GSE）[7]と呼ばれる線形方程式が得られる．

$$\begin{bmatrix} I & -J_{12} & -J_{13} \\ -J_{21} & I & -J_{23} \\ -J_{31} & -J_{32} & I \end{bmatrix} \frac{\mathrm{d}y}{\mathrm{d}x} = \begin{bmatrix} J_{10} \\ J_{20} \\ J_{30} \end{bmatrix} \tag{B.6}$$

この方程式を解きさえすれば，式(B.3a)～(B.3c)よりシステムの感度情報が得られる．ここで，感度解析のためにシステム解析を繰り返す必要はないことが特徴である．このようにして得られた感度情報を使って，システムレベルで最適化計算を繰り返す．

この All-at-Once 法は他の複合領域最適化法に比べると少ない解析回数で最適解を得ることができ，またロバスト性もあることが報告されている[13]．しかし，複合領域にわたるシステムの設計過程の中に組み込まれた各領域の専門家達は単なる解析者として扱われることよりも，自分の専門領域の中で最適化を行い変数を決定したいと思うものである．この意識こそが実在する組織に基づいた複合領域最適化法の分割解法の発達を促すことになったといえる．

B.2 モデル調整法

モデル調整法[4]は最適化する変数 $\boldsymbol{x}$ を次のように分ける．

$$\boldsymbol{x}=\left(\boldsymbol{y}^{\mathrm{T}},\boldsymbol{x}_1{}^{\mathrm{T}},\boldsymbol{x}_2{}^{\mathrm{T}},\ldots,\boldsymbol{x}_N{}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{B.7}$$

ここで，変数 $\boldsymbol{y}$ はサブシステム間の相互変数，調整変数あるいはグローバル変数などと呼ばれ，N は部分システムの数，また変数 $\boldsymbol{x}_i$ はサブシステム i に関するローカルな変数である．また，分割により制約関数 g_E，g_I と目的関数 f は次の形に書かれる．

$$\boldsymbol{g}_E(\boldsymbol{x})\equiv\begin{bmatrix} g_{E1}(\boldsymbol{y},\boldsymbol{x}_1)\\ g_{E2}(\boldsymbol{y},\boldsymbol{x}_2)\\ \vdots\\ g_{EN}(\boldsymbol{y},\boldsymbol{x}_N)\end{bmatrix},\quad \boldsymbol{g}_I(\boldsymbol{x})\equiv\begin{bmatrix} g_{I1}(\boldsymbol{y},\boldsymbol{x}_1)\\ g_{I2}(\boldsymbol{y},\boldsymbol{x}_2)\\ \vdots\\ g_{IN}(\boldsymbol{y},\boldsymbol{x}_N)\end{bmatrix} \tag{B.8a}$$

$$f(\boldsymbol{x})\equiv\sum_{i=1}^{N} f_i(\boldsymbol{y},\boldsymbol{x}_i) \tag{B.8b}$$

すなわち，変数 $\boldsymbol{y}$ はすべての関数に現れるが，変数 $\boldsymbol{x}_i$ は関数 g_{Ei}，g_{Ii} と目的関数の f_i の項のみに現れる．

以上から，元の問題は次のように書き下すことができる．

$$\text{variable}\qquad \boldsymbol{x}=\left(\boldsymbol{y}^{\mathrm{T}},\boldsymbol{x}_1{}^{\mathrm{T}},\boldsymbol{x}_2{}^{\mathrm{T}},\ldots,\boldsymbol{x}_N{}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{B.9a}$$

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(\boldsymbol{y}, \boldsymbol{x}_i) \tag{B.9b}$$

$$\text{subject to} \quad \begin{bmatrix} g_{E1}(\boldsymbol{y}, \boldsymbol{x}_1) \\ g_{E2}(\boldsymbol{y}, \boldsymbol{x}_2) \\ \vdots \\ g_{EN}(\boldsymbol{y}, \boldsymbol{x}_N) \end{bmatrix} = \boldsymbol{0} \tag{B.9c}$$

$$\begin{bmatrix} g_{I1}(\boldsymbol{y}, \boldsymbol{x}_1) \\ g_{I2}(\boldsymbol{y}, \boldsymbol{x}_2) \\ \vdots \\ g_{IN}(\boldsymbol{y}, \boldsymbol{x}_N) \end{bmatrix} \leq \boldsymbol{0} \tag{B.9d}$$

これらに基づき，モデル調整法は次の 2 段階最適化問題として定式化される．

(1) サブシステムレベル問題

変数 $\boldsymbol{y}$ の値を固定すると，式(B.9a)～(B.9d)で表される問題は N 個の独立なサブシステムレベルの問題に分割することができる．それぞれのサブシステムの最適化問題は以下のように定義できる．

$$\text{variable} \quad \boldsymbol{x}_i \tag{B.10a}$$

$$\text{minimize} \quad f_i(\boldsymbol{y}, \boldsymbol{x}_i) \tag{B.10b}$$

$$\text{subject to} \quad g_{Ei}(\boldsymbol{y}, \boldsymbol{x}_i) = \boldsymbol{0} \tag{B.10c}$$

$$g_{Ii}(\boldsymbol{y}, \boldsymbol{x}_i) \leq \boldsymbol{0} \tag{B.10d}$$

(2) システムレベル問題

システムレベルの最適化問題では，サブシステムレベル問題の結果を受けて，最適化する変数を $\boldsymbol{y}$ とする次のような問題を定義する．

$$\text{variable} \quad \boldsymbol{y} \tag{B.11a}$$

$$\text{minimize} \quad \sum_{i=1}^{N} f_i(\boldsymbol{y}, \boldsymbol{x}_i) \tag{B.11b}$$

$$\text{subject to} \quad \hat{\boldsymbol{g}}_E(\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = 0 \tag{B.11c}$$

$$\hat{\boldsymbol{g}}_I(\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \leq 0 \tag{B.11d}$$

ここでの制約条件はサブシステムレベル問題が実行可能解を持つための条件である．よって，これらにはサブシステムレベル問題のすべての制約条件を含めることが本来求められる．しかしながらその場合，最適化計算において制約条件の処理に大きな負荷を要し分割化の利点が失われるといった理由から，これら2つの制約条件を簡略化する必要がある．例えば，文献50によれば，サブシステムレベル問題の制約条件の中から活性または活性に近い状態にある（critical）制約条件を集めて構成するとある．

以上より，モデル調整法は次のような繰り返し計算によって解かれる．

Step 1.

グローバル変数 y の初期解を選ぶ．

Step 2.

与えられた y に対して，N 個の独立なサブシステムレベル問題を解く．

Step 3.

サブシステムレベル問題の解 x_i を受けたシステムレベル問題によって変数 y を変化させる．変数 y が変化しなくなり，システムレベル問題の最小値となるまで Step 2.へ戻る．

ここで，Step 3.では変数 y を最適解に近付ける更新をしているだけであって，システムレベル問題を完全に解いているわけではないことに注意する．システムレベル問題を解くときに，目的関数と制約関数のグローバル変数に対する感度が必要となる．例えば，グローバル変数に対するサブシステムレベル問題の解の感度 $\mathrm{d}x_i/\mathrm{d}y$ が得ら[51]れれば，システムレベル問題の各関数の勾配は以下より求められる[52]．

$$\frac{\mathrm{d}}{\mathrm{d}y}\sum_{i=1}^{N} f_i(y,x_i)=\sum_{i=1}^{N}\left(\frac{\partial f_i}{\partial y}+\frac{\partial f_i}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}y}\right) \tag{B.12a}$$

$$\frac{\mathrm{d}}{\mathrm{d}y}\hat{g}_E(y,x_1,\ldots,x_N)=\frac{\partial \hat{g}_E}{\partial y}+\sum_{i=1}^{N}\frac{\partial \hat{g}_E}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}y} \tag{B.12b}$$

$$\frac{\mathrm{d}}{\mathrm{d}y}\hat{g}_I(y,x_1,\ldots,x_N)=\frac{\partial \hat{g}_I}{\partial y}+\sum_{i=1}^{N}\frac{\partial \hat{g}_I}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}y} \tag{B.12c}$$

モデル調整法の最大の問題点は，グローバル変数の値によってはサブシステムレベル問題で制約条件を満足する実行可能解が存在しない場合がありうる点である．また，この手法ではグローバル変数に対するサブシステムレベル問題の感度が連続でない場合がある

ことが指摘されている[53]．さらに，目的関数が式(B.8b)のように全ての部分システムの変数 x_i ごとに分割できない場合，サブシステムレベル問題が定義することができない．これらの問題点に対しては1次レベル問題で幾つかの制約条件をペナルティ関数とすれば良いことが報告されている[6, 53, 54]．一方で，モデル調整法では繰り返し計算途中の変数は必ず実行可能解であるので，たとえそれが最適解でなくても計算を打ち切ることができ，工学的な観点から有利な解法であるともいえる．

B.3 ゴール調整法

ゴール調整法[41]ではサブシステム間の繋がりは全て切断される．グローバル変数 y は切断されたサブシステム間で値が異なっていても良いとし，変数 y_i をサブシステム i のグローバル変数とする．本項では図B.2に示すような直列的にサブシステムが結合されているシステムのみを考える．最適解で満足すべきグローバル変数の条件は次式で表される．

$$y_{i,i+1} = y_{i+1,i} \quad (i = 1,2,\ldots,N-1) \tag{B.13}$$

この条件を満たすために新たな変数 $\lambda^{\mathrm{T}} = \left(\lambda_{12}{}^{\mathrm{T}}, \lambda_{23}{}^{\mathrm{T}}, \ldots, \lambda_{N-1N}{}^{\mathrm{T}}\right)$ を導入し，新しい目的関数を次式のように定義する．

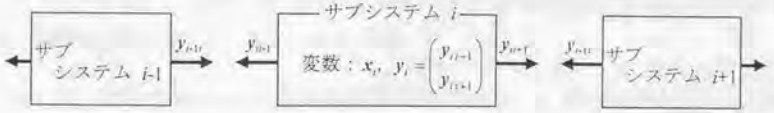
$$\phi(y_1,\ldots,y_N,x_1,\ldots,x_N,\lambda) = \sum_{i=1}^{N} f_i(y_i,x_i) + \sum_{i=1}^{N} \lambda_{i+1}{}^{\mathrm{T}}(y_{i+1} - y_{i+1,i}) \tag{B.14}$$

ここで，

$$\lambda_1 = \lambda_{12}, \quad \lambda_N = -\lambda_{N-1N} \tag{B.15a}$$

$$\lambda_i = \left(-\lambda_{i-1,i}{}^{\mathrm{T}}, \lambda_{i,i+1}{}^{\mathrm{T}}\right)^{\mathrm{T}} \quad (i = 2,3,\ldots,N-1) \tag{B.15b}$$

$$\lambda = \left(\lambda_1{}^{\mathrm{T}}, \lambda_2{}^{\mathrm{T}}, \ldots, \lambda_N{}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{B.15c}$$



図B.2 分割されたシステム

とすると，

$$\phi(y_1,\ldots,y_N,x_1,\ldots,x_N,\lambda)=\sum_{i=1}^{N}\left\{f_i(y_i,x_i)+\lambda_i^{\mathrm{T}}y_i\right\} \tag{B.16}$$

のように変形される．したがって，2段階最適化問題は以下のようになる．

(1) サブシステムレベル問題

λを固定するとサブシステムレベルの最適化問題は次のN個の部分問題から成る．

$$\begin{aligned}
&\text{variable} && x_i,\ y_i && \text{(B.17a)}\\
&\text{minimize} && f_i(y_i,x_i)+\lambda_i^{\mathrm{T}}y_i && \text{(B.17b)}\\
&\text{subject to} && g_{Ei}(y_i,x_i)=0 && \text{(B.17c)}\\
& && g_{Ii}(y_i,x_i)\le 0 && \text{(B.17d)}
\end{aligned}$$

(2) システムレベル問題

システムレベルの最適化問題の目的は，グローバル変数の一致条件（式(B.13)）が満足されるように，サブシステム問題を調整することである．すなわち，適当なλを選ぶことにより，異なった値を取る$y_{i,i+1}$と$y_{i+1,i}$を等しくすることである．

$$\text{maximize}\quad \sum_{i=1}^{N}\lambda_i^{\mathrm{T}}y_i \tag{B.18}$$

よって，ゴール調整法は次のような繰り返し計算によって解かれる．

Step 1.

変数λの初期解を選ぶ．

Step 2.

与えられたλに対して，N個の独立なサブシステムレベル問題を解く．

Step 3.

サブシステムレベル問題の解y_iを受け，システムレベル問題によって変数λを変化させる．変数λが変化しなくなるまでStep 2.に戻る．

ここで，サブシステムレベル問題は目的関数ϕを変数x_i，y_iに関して最小化する最適化問題であり，一方，システムレベル問題は目的関数ϕを変数λに関して最大化する最適化

問題である．各変数の最適解を x_i^0，y_i^0，λ^0 とすると，

$$\phi(y_i^0, x_i^0, \lambda) \le \phi(y_i^0, x_i^0, \lambda^0) \le \phi(y_i, x_i, \lambda^0) \tag{B.19}$$

が成り立つ．すなわち，この解法は変数λをラグランジュ乗数として，関数φで定義されるラグランジュ関数の最適性の条件に注目した解法である．元の問題を主問題とし，システムレベル問題では双対問題が定義されているといえる．ゴール調整法では式(B.19)に沿った鞍点が存在する場合，すなわち凸計画問題に対してのみ収束性が保証されている．逆に，非凸計画問題に対しては局所解すら見つけ出せる保証がない．また，モデル調整法では変数の総数が元の問題と変わらないが，ゴール調整法では2段階の定式化における変数の数が元の問題より多いという欠点を持つ．さらに，繰り返し計算途中の解が実行可能解でないために，途中の解が近似解として使えないので，有用な解を求めるためには最適解が得られるまで反復計算を続けなければならない．

B.4 2つの調整法のハイブリッド解法

複数のサブシステムからなる問題を分割解法により定式化する場合，各サブシステムに対して異なった調整法を用いることも可能である．すなわち，その有効性によって，あるサブシステムにはゴール調整法を，他のサブシステムに対してはモデル調整法を用いるように分割することも可能である．文献6では，2 つのサブシステムの変数 x_i と x_j に依存する次のような形の制約条件

$$h_{k1}(x_i, y) - h_{k2}(x_j, y) = 0 \tag{B.20}$$

が存在する場合を考慮している．この場合，モデル調整法による定式化は不可能であるとし，その制約条件はゴール調整法的な定式化により扱った．すなわち，ラグランジュ乗数 λ_k を導入し，目的関数を

$$\phi(y_1, \ldots, y_N, x_1, \ldots, x_N, \lambda_k) = \sum_{i=1}^{N} f_i(y_i, x_i) + \sum_k \lambda_k^{\mathrm{T}} \left\{ h_{k1}(x_i, y) - h_{k2}(x_j, y) \right\} \tag{B.21}$$

とした．その上で，まず，サブシステムレベル問題では各部分問題での変数 x_i による関数φの最小化を行う．システムレベル問題は 2 層に分けられる．下層レベルではラグランジュ乗数 λ_k による最大化，上層レベルの問題として変数 y の最小化を行う．以上，サブ

システムレベル，システムレベルの下層レベル，システムレベルの上層レベルの 3 段階からなる最適化を順に繰り返す．

このようなモデル調整法とゴール調整法のハイブリッド的な解法でも，2 つの調整法が持ち合わせている欠点は改善されていない．

B.5 Concurrent Subspace Optimization（CSSO）法

NASA ラングレー研究所の Sobieski によって文献8において Concurrent Subspace Optimization（CSSO）法と呼ばれる方法が提案されている．最適化を行う変数を領域（サブシステム）に対応した変数に分ける．この方法では，個々のサブシステムは他のサブシステムの目的関数と制約条件を含めた全ての目的関数と制約条件を持つ．個々のサブシステム内の最適化における解析作業では，そのサブシステムに関連した目的関数と制約関数はそれ自身の関数を評価するが，他のサブシステムの関数は近似化技術を使って求められる近似関数を評価することにする．例えば，All-at-Once 法で述べた GSE を使いサブシステムの感度を求め，1 次関数式に近似することが最も単純な近似法である．特に近似化技術として最近注目されているのは応答曲面（response surface）[55]による近似である．

以上のサブシステムの最適化のみを繰り返しても，変数がシステムの最適解に収束するとは限らないことは明らかである．なぜならば，近似化されたシステムの制約条件を各サブシステムが満たすことは不可能であるからである．実際，領域間のカップリングが存在しない場合のみ，システムの最適解に収束することができる．そこで，「responsibility」係数と呼ばれる定数を導入して，近似化された制約条件を完全に満たさなければならないという要求を緩めることにする．サブシステムレベルの最適化に続き，この係数に対するサブシステムの最適解の感度を求める．この感度情報を用いてシステムレベルの最適化問題では「responsibility」係数値を変更し，領域間のカップリングが満たされるようにする．よって，サブシステムレベル問題の最適化，システムレベル問題の最適化，そして近似化が繰り返し行われ変数を最適解に収束させる．

CSSO 法の計算コストはカップリングの量に依る．極端な例として，もし全ての変数が全てのサブシステムに関連するならば，各サブシステムは全ての制約条件を持つため，サブシステムの計算量は急増する．また，解法のロバスト性もカップリングの強さに依存する．All-at-Once 法，Collaborative Optimization 法に比べればロバスト性はかなり劣る[17]．

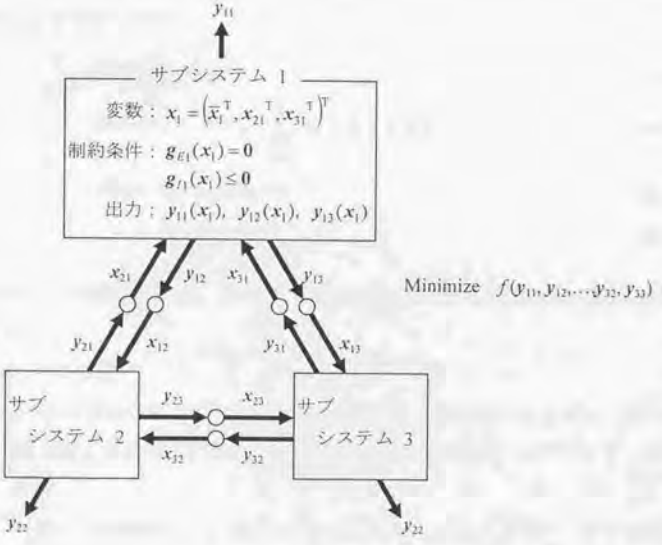
B.6 Collaborative Optimization（CO）法

Kroo，Braun らによって Collaborative Optimization（CO）法と呼ばれる方法が文献9，10，11で提案され，同文献や文献16，17などで，宇宙機，航空機の概念設計に対する適用例が見られる．この方法は現在最も有効な複合領域最適化法であるとの評価が高い．

最適化するシステムが N 個のサブシステムからなる問題を考えてみる．図B.3は $N=3$ の簡単な場合である．あるサブシステム i で最適化される変数を $\boldsymbol{x}_i$ とすると次のように定義される．

$$\boldsymbol{x}_i = \left(\bar{\boldsymbol{x}}_i^{\mathrm{T}}, \boldsymbol{x}_{1i}^{\mathrm{T}}, \ldots, \boldsymbol{x}_{ji}^{\mathrm{T}}, \ldots, \boldsymbol{x}_{Ni}^{\mathrm{T}}\right)^{\mathrm{T}} \quad (j \neq i) \tag{B.22}$$

ここで，$\boldsymbol{x}_{ji}$ はサブシステム i への他のサブシステム j からの入力である．$\bar{\boldsymbol{x}}_i$ はサブシステム i に内包された変数であり，そのサブシステムの外からは参照することができない変数である．サブシステム i の制約条件を，



図B.3　3つのサブシステムと変数の入出力

$$g_{Ei}(\boldsymbol{x}_i)=\mathbf{0} \tag{B.23a}$$

$$g_{Ii}(\boldsymbol{x}_i)\le\mathbf{0} \tag{B.23b}$$

と定義する．また他のサブシステム j（$j\neq i$）への出力を y_{ji} とする．この出力は変数 x_i の関数であることに注意する．図B.3からも明らかな通り，

$$y_{ji}=x_{ji} \tag{B.24}$$

が成り立たなければならない．さらに，目的関数のためだけに出力される変数を y_{ii} とすると，最適化する目的関数は各サブシステムからの出力の関数として定義される．

$$\text{maximize}\quad f(y_{11},y_{12},\ldots,y_{ij},\ldots,y_{NN-1},y_{NN}) \tag{B.25}$$

これらに基づき，CO 法での 2 段階アプローチは以下のように定式化される．

(1) サブシステムレベル問題

各サブシステムには次に述べるシステムレベル問題から，入力 x_{ji} と出力 y_{ij} の目標値が z_{ij} として与えられ，制約条件を満たしつつ，可能な限り入力と出力がそれに近い変数値 x_i を求めることを目標とする．

$$\text{variable}\quad \boldsymbol{x}_i \tag{B.26a}$$

$$\text{minimize}\quad J_i(\boldsymbol{x}_i)\equiv\sum_{\substack{j=1\\(j\neq i)}}^{N}\left(x_{ji}-z_{ji}\right)^2+\sum_{j=1}^{N}\left\{y_{ij}(\boldsymbol{x}_i)-z_{ij}\right\}^2 \tag{B.26b}$$

$$\text{subject to}\quad g_{Ei}(\boldsymbol{x}_i)=\mathbf{0} \tag{B.26c}$$

$$g_{Ii}(\boldsymbol{x}_i)\le\mathbf{0} \tag{B.26d}$$

サブシステムレベル問題において，目的関数は目標値と変数値の 2 乗誤差となっていることに注目する．

(2) システムレベル問題

システムレベルの最適化問題では元の問題の目的関数を最適化するように，サブシステムに与える目標値 z_{ij} を決めるための次の最適化問題が定義できる．

$$\text{variable}\quad \boldsymbol{z}=\left(\boldsymbol{z}_{11}^{\mathrm{T}},\boldsymbol{z}_{12}^{\mathrm{T}},\ldots,\boldsymbol{z}_{ij}^{\mathrm{T}},\ldots,\boldsymbol{z}_{NN-1}^{\mathrm{T}},\boldsymbol{z}_{NN}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{B.27a}$$

$$\text{minimize}\quad f(\boldsymbol{z}) \tag{B.27b}$$

$$\text{subject to} \quad \begin{bmatrix} J_1(z) \\ J_2(z) \\ \vdots \\ J_N(z) \end{bmatrix} = \mathbf{0} \tag{B.27c}$$

サブシステムレベル問題において目的関数として定義された目標値と変数値の2乗誤差が0になるように，ここでは制約条件となっていることに注意する．

よって，CO 法は次のような繰り返し計算によって解かれる．

Step 1.

目標値 z を選ぶ．

Step 2.

与えられた z に対して，N 個の独立なサブシステムレベル問題を解く．

Step 3.

システムレベル問題の結果を受け，システムレベル問題によって目標値 z を変化させる．目標値 z が変化しなくなるまで Step 2.に戻る．

ここで，Step 2.のサブシステムレベル問題から Step 3.のシステムレベル問題に移るとき，関数 J_i の値とともに勾配 $\partial J_i/\partial z$ が与えられなければならない．ここで，目標値が変化するとサブシステムレベル問題の最適解も変化するため，この勾配を正確に求めることは難しい．文献10，11によると，この目標値による変数値の変化を無視し，

$$\frac{\partial J_i}{\partial z_{jl}} = -2\left(x_{jl} - z_{jl}\right) \quad (j \neq i) \tag{B.28a}$$

$$\frac{\partial J_i}{\partial z_{il}} = -2\left(y_{il} - z_{il}\right) \tag{B.28b}$$

としても良いとある．

この CO 法では，Kirsch のモデル調整法と違い，サブシステムレベル問題の目的関数を必ず定義することができる．また，サブシステムレベル問題が実行可能解を持たなくなる場合はない．文献13によると，CO 法はカップリングが強くなると最適化計算のロバスト性は低下するものの，他の解法と比較するとロバスト性に優れた解法である．しかし，CO

法は最適解を得るために非常に多くの繰り返し計算を必要とする．また理論面にも問題点がある．最適解ではシステムレベル問題の制約関数J_iの勾配$\partial J_i/\partial z$が**0**になる．一方で目的関数の勾配は**0**にならないため，最適性の条件であるKuhn-Tucker条件（補遺A）が最適解では満たされない．よって，CO法は現在ある変数値を改善させるには良いが，解の収束性は保証されていない．特に繰り返し計算を停止させるための条件を決める理論的研究が必要である．

補遺C

最適制御問題に対するBDH法

本章では本文中に最適制御問題（軌道最適化問題）の解法として述べられたBDH法を簡単に説明する．より詳細な説明及びその性質は関連文献を参照されたい．

時間の関数である動的変数を最適化する最適制御問題には，変分法，最大原理，動的計画法等に基づく数値解法[56, 57]が多数存在し，静的変数を最適化する補遺Aの非線形計画問題とは異なる発展を遂げてきた．しかし最近の研究では，最適制御問題の解法にも非線形計画法を用いた解法が使用されつつある[58]．その理由として，この解法には扱う変数の数が多くなるという欠点が存在するが，計算機の能力の向上により多くの変数を扱えるようになってきたことが挙げられる．また，微分方程式，不等式拘束，そして最適制御問題に対する変分法に基づく解法において苦手であった状態量拘束をはじめとした様々な拘束条件を扱える有効な計算アルゴリズムが提案されつつあるためである．ブロック対角ヘシアン（Block Diagonal Hessian: BDH）法も，時間の関数である制御量と状態量，状態方程式を離散化して非線形計画問題へ変換する解法（collocation法）の1つである．

C.1 最適制御問題の定義

ある動的システムにおいて，時間$t \in [t_0, t_f]$の関数である状態変数$w(t) \in R^n$と制御変数$u(t) \in R^m$，また時間に依存しない静的変数$\pi \in R^l$を定義する．その上で，システムの状態方程式が次の微分方程式で与えられているとする．

$$\dot{w}(t) = f[w(t), u(t), \pi] \tag{C.1}$$

初期時間$t = t_0$における初期条件と，終端時間$t = t_f$における終端条件がそれぞれ以下のように与えられている．

$$\varphi_0[w(t_0),u(t_0),\pi]=0 \tag{C.2a}$$

$$\varphi_f[w(t_f),u(t_f),\pi]=0 \tag{C.2b}$$

加えて，制御時間 $t\in[t_0,t_f]$ に満たされなければならない拘束条件が存在する場合には

$$C[w(t),u(t),\pi]=0 \tag{C.3a}$$

$$S[w(t),u(t),\pi]\le 0 \tag{C.3b}$$

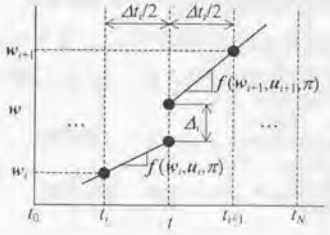
を定める．そして，次の評価関数を最小化する状態変数，制御変数，初期時間，終端時間，静的変数を求める問題を最適制御問題と呼ぶ．

$$J=\phi[w(t_0),w(t_f),u(t_0),u(t_f),t_0,t_f,\pi]+\int_{t_0}^{t_f}\psi[w(t),u(t),\pi]dt \tag{C.4}$$

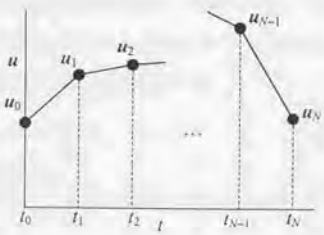
C.2 非線形計画問題への変換

BDH 法は前節で定義された最適制御問題を非線形計画問題に変換し，得られた問題を解いて最適解を求める数値解法である．非線形計画問題へ変換するために，状態量 $w(t)$ と制御量 $u(t)$ を離散化し区分関数によって表すことにする．はじめに，初期時間 t_0 から終端時間 t_f までの時間 t を N 個の要素に分割する．要素と要素をつなぐ点を節点と呼び，その時間を t_i （$i=0,\ldots,N$）と表す．

$$t_0<\ldots<t_i<\ldots<t_N=t_f \tag{C.5}$$



図C.1 状態量の離散化



図C.2 制御量の離散化

各節点上の状態量，制御量を w_i， u_i とする．

$$w_i \equiv w(t_i) \tag{C.6a}$$

$$u_i \equiv u(t_i) \tag{C.6b}$$

とする．また時間要素幅を

$$\Delta t_i \equiv t_{i+1} - t_i = K_i(t_f - t_0) \tag{C.7}$$

で与える．ここで，K_i は定数であり制御時間に占める時間要素幅の割合を表す．最適化により初期時間 t_0，終端時間 t_f は変化するが，定数 K_i は一定であることに注意する．その上で，各要素上で状態量を，図C.1に示す通り 2 本の 1 次関数で表すと，要素中央で状態量が連続する必要性から以下の関係式が要求される．

$$\begin{aligned} \Delta_i \equiv & \left\{ w_i + f(w_i, u_i, \pi)\frac{\Delta t_i}{2} \right\} \\ & - \left\{ w_{i+1} - f(w_{i+1}, u_{i+1}, \pi)\frac{\Delta t_i}{2} \right\} = 0 \end{aligned} \tag{C.8}$$

式(C.8)の定式化は状態量を求めるために最適制御問題の状態方程式を修正オイラー法で数値積分することに等しい．

次に，離散化された変数を集め変数 x_i （$i = 0, \ldots, N+1$）を次のように定める

$$x_i \equiv \left(w_i^{\mathrm{T}}, u_i^{\mathrm{T}} \right)^{\mathrm{T}} \in R^{n+m} \qquad (i = 0, \ldots, N) \tag{C.9a}$$

$$x_{N+1} \equiv \left(\pi^{\mathrm{T}}, t_0, t_f \right)^{\mathrm{T}} \in R^{l+2} \tag{C.9b}$$

なお，初期時間あるいは終端時間が特定され最適化する必要のない問題に対しては，式(C.9b)の変数 x_{N+1} の中に初期時間 t_0 または終端時間 t_f を含める必要はない．変数 π についても同様である．これら変数を合わせ，次の非線形計画問題を定義する．

$$\text{variable} \quad x \equiv \left(x_0^{\mathrm{T}}, \ldots, x_N^{\mathrm{T}}, x_{N+1}^{\mathrm{T}} \right)^{\mathrm{T}} \in R^{(n+m)(N+1)+l+2} \tag{C.10a}$$

$$\text{minimize} \quad f(x) \equiv \phi(w_0, w_N, u_0, u_N, t_0, t_f, \pi) + \sum_{i=0}^{N-1} \sigma_i(w_i, u_i, w_{i+1}, u_{i+1}, \pi, t_0, t_f) \tag{C.10b}$$

$$\text{subject to} \quad g_E(x) \equiv \begin{bmatrix} \varphi_0(w_0, u_0, \pi) \\ \begin{bmatrix} C(w_i, u_i, \pi) \\ \Delta_i \end{bmatrix} \\ \varphi_f(w_N, u_N, \pi) \end{bmatrix} = 0 \tag{C.10c}$$

$$g_I(x) \equiv [S(w_i, u_i, \pi)] \le 0 \qquad (i = 0, \ldots, N) \tag{C.10d}$$

ここで，式(19)の非線形計画問題の目的関数fには最適制御問題の評価関数ϕが代入され，関数σ_iは以下に定める．

$$\sigma_i(w_i, u_i, w_{i+1}, u_{i+1}, \pi, t_0, t_f) \equiv \{\psi(w_i, u_i, \pi) + \psi(w_{i+1}, u_{i+1}, \pi)\} \frac{\Delta t_i}{2} \tag{C.11}$$

これは評価関数の積分項の被積分関数ψを台形則で数値積分することに等しい．等式制約関数g_Eには最適制御問題の初期条件，終端条件，等式拘束条件の関数φ_0，φ_f，Cと共に式(C.8)が加えられる．不等式制約関数g_Iには不等式拘束条件の関数Sが代入される．

C.3 Block Diagonal Hessian（BDH）法

式(C.10a)～(C.10d)で定義される非線形計画問題の解法に逐次2次計画（SQP）法（補遺A）を用いる．この解法では関数f，g_E，g_Iの変数xに対する勾配からなるヤコビ行列，及びラグランジュ乗数をλ_E，λ_Iとするラグランジュ関数

$$L(x, \lambda_E, \lambda_I) \equiv f + \lambda_E{}^{\mathrm{T}} g_E + \lambda_I{}^{\mathrm{T}} g_I \tag{C.12}$$

の2階の偏微係数からなるヘッセ行列$\nabla^2_{xx} L$の近似行列Bを必要とする．式(C.10a)～(C.10d)で構成される問題のヤコビ行列は容易に求められる．一方，ヘッセ行列は以下の構造をしている．

$$\nabla^2_{xx} L = \left[\begin{array}{ccc:c} \nabla^2_{x_0 x_0} L & & 0 & \nabla^2_{x_0 x_{N+1}} L \\ & \ddots & & \vdots \\ 0 & & \nabla^2_{x_N x_N} L & \nabla^2_{x_N x_{N+1}} L \\ \hdashline \nabla^2_{x_{N+1} x_0} L & \cdots & \nabla^2_{x_{N+1} x_N} L & \nabla^2_{x_{N+1} x_{N+1}} L \end{array}\right] \in R^{\{(n+m)(N+1)+(l+2)\} \times \{(n+m)(N+1)+(l+2)\}} \tag{C.13}$$

このヘッセ行列$\nabla^2_{xx} L$は近似公式によって求められる近似行列で代用することを補遺Aで説明した．ここで，行列$\nabla^2_{xx} L$を式(C.13)のように大きく4つに分割すると左上部分がブ

ロック対角化されていることに注目する．ヘッセ行列$\nabla^2_{xx}L$の近似行列$\boldsymbol{B}$を，$\nabla^2_{x_i x_i}L$（$i=0,\ldots,N$）の近似行列$\boldsymbol{B}_i \in \boldsymbol{R}^{(n+m)\times(n+m)}$，$\nabla^2_{x_{N+1}x_{N+1}}L$の近似行列$\boldsymbol{B}_{N+1}\in\boldsymbol{R}^{(l+2)\times(l+2)}$として，

$$\boldsymbol{B}=\left[\begin{array}{ccc:c} \boldsymbol{B}_0 & & 0 & \\ & \ddots & & 0 \\ 0 & & \boldsymbol{B}_N & \\ \hdashline & 0 & & \boldsymbol{B}_{N+1} \end{array}\right] \tag{C.14}$$

と置く．これにより，行列$\boldsymbol{B}$を密行列と考えて行列全体に近似公式を適用して近似していくのではなく，各ブロック行列$\boldsymbol{B}_i$ごとに公式を適用して，それぞれを近似，保存できるようになり，また SQP 法の数値計算をある程度分割して行うことが可能になる．これにより収束性を高め，計算の効率化をはかることができる．

ところで，式(C.14)において対角化のため式(C.13)の非対角ブロック行列$\nabla^2_{x_i x_{i+1}}L$（$=\nabla^2_{x_{i+1}x_i}L^{\mathrm{T}}$）を零行列としている．この場合でも，変数$\boldsymbol{x}_i$の最適化を行うとき，あるいは変数$\boldsymbol{x}_{N+1}$の最適化を行うとき，他方の変数を一定として相互に及ぼし合う影響を考慮せず繰り返し最適化していく手法に相当する収束性はもつことが保証されている．

以上の非線形計画問題の定式化（式(C.10a)～(C.10d)）とその解法をブロック対角ヘシアン（Block Diagonal Hessian: BDH）法と呼ぶ．

補遺D

空力特性の推算法“CRSFLW”

本論文では第 6 章の空力解析に文献59，60にある空力特性推算法を用いた．この方法によるプログラム CRSFLW が別途文献61にある．本研究ではすでに空力特性の解析が行われている既存の機体モデルを変更するだけでなく，新しい形状の機体を生み出すことを考え，任意形状に対する空力解析法を取り入れる必要があった．CRSFLW によれば任意の迎角とマッハ数で飛行する物体の空気力を簡易的に推定することができる．この方法では，空気力を法線力と軸力に分け，スレンダーボディ理論によって与えられるポテンシャル項とニュートニアン理論によって修正された粘性横断流項から機体の法線力を求める．軸力は摩擦力，造波抗力，底面からの抗力の総和として求められるが，文献62，63を参考に修正を加えている．ただし，胴体のノーズ，翼前縁から流れる渦による非線形効果は計算に含まれておらず，また有限要素法的な CFD と比べると非常に簡単な推定法である．実際，この方法と風洞試験データを比較してみると，揚力を亜音速で小さく見積もっていることが判明したため，独自のパラメータを加えた．

本章ではまず機体形状と本章で使用するパラメータについてまとめた後，文献59，60に従ってプログラム CRSFLW の方法を説明する．各式の詳しい拠出法とその意味は文献59，60，62，63を参照されたい．次に，本章では風洞試験データのある機体モデルに本推定法を適用して試験データと比較し，パラメータ値を決定する．

D.1 機体形状

本手法で扱われる機体形状とその説明に使用される記号を図D.1にまとめる．機体は主翼と胴体からなる．翼はデルタ翼などの任意の平面形である中翼である．ただし厚みを持たない平板の翼のみしか CRSFLW では扱うことができない．胴体先端からの位置 x での

断面に現れる翼幅の半分の長さを $s(x)$ とするが，翼がない位置での $s(x)$ は $b(x)$ に等しい．

また，CRSFLW を適用するためには胴体の断面形状は楕円でなければならない．胴体は任意の平面形状をもつノーズ部分と楕円柱部分からなる．本論文ではノーズとして tangent ogive ノーズ（図D.2）を採用した．このノーズは後ろに続く楕円柱の胴体と滑らかに接続する．tangent ogive ノーズの幾何学的パラメータを以下にまとめておく．

ノーズの平面形状 $a(x)$，$b(x)$，体積 V は以下のようになる．

$$a(x)=\frac{a_N^{\,2}-l_N^{\,2}+\sqrt{(a_N^{\,2}-l_N^{\,2})^2-4a_N^{\,2}x(x-2l_N)}}{2a_N} \tag{D.1a}$$

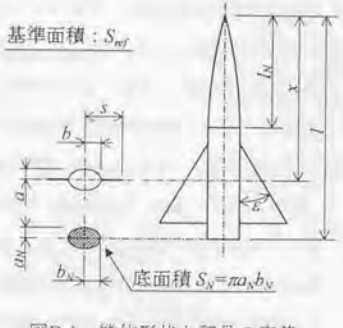
$$b(x)=\frac{b_N^{\,2}-l_N^{\,2}+\sqrt{(b_N^{\,2}-l_N^{\,2})^2-4b_N^{\,2}x(x-2l_N)}}{2b_N} \tag{D.1b}$$

$$\begin{aligned}V=&\frac{\pi(a_N^{\,4}+b_N^{\,4})}{2a_N^{\,3}b_N^{\,3}}l_N^{\,5}-\frac{2\pi(a_N^{\,2}+b_N^{\,2})}{3a_Nb_N}l_N^{\,3}+\pi a_Nb_Nl_N\\&+\frac{\pi a_N(b_N^{\,4}-l_N^{\,4})}{4b_N^{\,2}}\sin^{-1}\left(\frac{2b_Nl_N}{a_N^{\,2}+l_N^{\,2}}\right)+\frac{\pi b_N(a_N^{\,4}-l_N^{\,4})}{4a_N^{\,2}}\sin^{-1}\left(\frac{2a_Nl_N}{b_N^{\,2}+l_N^{\,2}}\right)\end{aligned} \tag{D.1c}$$

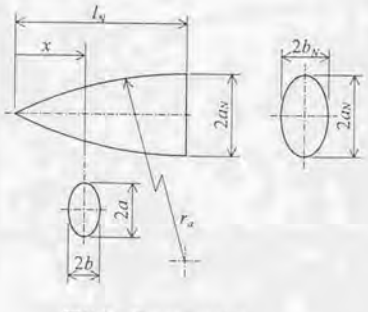
表面積 A_S は断面が楕円であるために解析的に求めるのが難しく，次のように近似する．

$$A_S=2\pi(l_N^{\,2}+a_Nb_N)\left\{\frac{l_N^{\,2}-a_Nb_N}{(a_N+b_N)^2}\sin^{-1}\left(\frac{-l_N(a_N+b_N)}{l_N^{\,2}+a_Nb_N}\right)+\frac{l_N}{a_N+b_N}\right\}\frac{64-3\sigma^4}{64-16\sigma^2} \tag{D.2a}$$

$$\sigma=\frac{a_N-b_N}{a_N+b_N} \tag{D.2b}$$



図D.1　機体形状と記号の定義



図D.2　Tangent ogive

D.2 揚力係数と抗力係数

プログラム CRSFLW では空気力を法線力と軸力に分けて考える．揚力係数C_L，抗力係数C_Dは，迎角αにより，法線力係数C_Nと軸力係数C_Aから次のように表される．

$$C_L = C_N \cos\alpha - C_A \sin\alpha \tag{D.3a}$$

$$C_D = C_N \sin\alpha + C_A \cos\alpha \tag{D.3b}$$

次節から法線力係数C_Nと軸力係数C_Aの計算法を示す．

D.3 法線力係数

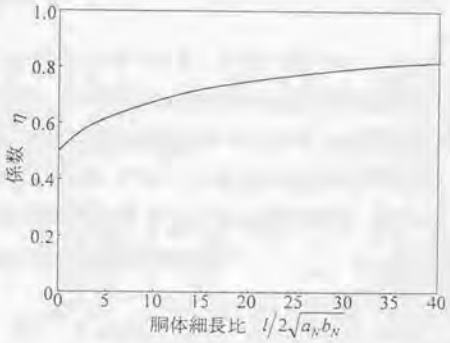
胴体のみ，あるいは翼に相当する揚力面を持つ物体に対する法線力係数C_Nは，

$$C_N = C_{N1} \sin 2\alpha \cos\frac{\alpha}{2} + C_{N2} \sin^2\alpha \tag{D.4a}$$

$$C_{N1} = \frac{S_B}{S_{ref} l} \int_0^l \left(\frac{C_n}{C_{n0}}\right)_{SB} \lambda dx \tag{D.4b}$$

$$C_{N2} = \frac{2\eta C_{dn}}{S_{ref}} \int_0^l \left(\frac{C_n}{C_{n0}}\right)_{Newt} \sqrt{ab}\, dx \tag{D.4c}$$

と表される．式(D.4a)の第 1 項目はスレンダボディポテンシャル理論から算出され，第 2 項目は剥離を伴う粘性横断流による法線力を表す．ここで，(C_n/C_{n0})は円柱断面の抵抗係数C_{n0}と任意形状を持つ胴体断面の抗力係数C_nの比である．比$(C_n/C_{n0})_{SB}$はスレンダボディ理論から決まり，比$(C_n/C_{n0})_{Newt}$はニュートニアン理論から決定される．また，式(D.4c)のC_{dn}は 2 次元円柱の抵抗係数であり，横断流マッハ数（$M_\infty \sin\alpha$），横断流レイノルズ数（$Re\sin\alpha$）によって変化する．本式では 2 次元円柱の抵抗係数C_{dn}を 3 次元円柱の抵抗係数に変換するために係数ηを用いている．係数ηは胴体細



図D.3 C_{dn}に対する補正値η

長比（$l/2\sqrt{a_N b_N}$）によって，図D.3のように求められる．

式(D.4b)には，デルタ翼を持つ胴体に対するスレンダボディ理論の修正係数λが含まれている．この係数λは以下のように与えられる．

$$\lambda = \frac{1}{E\left(\sqrt{1-\beta^2\tan^2\varepsilon}\right)} \qquad (\text{subsonic leading edge，}\ \beta\tan\varepsilon \le 1 \text{のとき}) \qquad \text{(D.5a)}$$

$$\lambda = \frac{2}{\pi\beta\tan\varepsilon} \qquad (\text{supersonic leading edge，}\ \beta\tan\varepsilon \ge 1 \text{のとき}) \qquad \text{(D.5b)}$$

ここで，式(D.5a)の関数Eは第2種完全楕円積分，εはデルタ翼の半頂角，また

$$\beta = \sqrt{M_\infty{}^2 - 1} \qquad \text{(D.6)}$$

である．

さて，翼を持つ楕円断面に対する$\left(C_n/C_{n0}\right)_{\mathrm{SB}}$は次のように求められる．

$$\left(\frac{C_n}{C_{n0}}\right)_{\mathrm{SB}} = \frac{1}{ab}\left(\kappa^2 + b^2\right) \qquad \text{(D.7a)}$$

$$\kappa = \sigma - \frac{(a+b)^2}{4\sigma} \qquad \text{(D.7b)}$$

$$\sigma = \frac{1}{2}\left(\bar{s} + \sqrt{\bar{s}^2 + a^2 - b^2}\right) \qquad \text{(D.7c)}$$

$$\bar{s} = b + k_{\mathrm{SB}}(s-b) \qquad \text{(D.7d)}$$

ここで，パラメータk_{SB}は本論文で導入したパラメータである．本論文で参考にしている文献60では，楕円断面形状を持つ胴体に対し，横に突き出た厚みを持たない平板の翼を想定している．後述するように，風洞試験によって得られた航空機モデルにこの式を適用すると，亜音速で法線力を小さく見積もる傾向がある．そこで，翼による法線力の寄与分を補正するために，翼の半スパンに相当するsをパラメータk_{SB}によって修正する．$k_{\mathrm{SB}}=1$のとき文献60の定式化と一致するが，具体的な値は後に決定される．

一方，$\left(C_n/C_{n0}\right)_{\mathrm{Newt}}$は，$a>b$のとき

$$\left(\frac{C_n}{C_{n0}}\right)_{Newt}=\frac{3}{2}\sqrt{\frac{b}{a}}\left[\frac{a^2/b^2}{\left(a^2/b^2-1\right)^{3/2}}\tan^{-1}\left(\sqrt{\frac{a^2}{b^2}-1}\right)-\frac{1}{a^2/b^2-1}+\frac{s}{b}-1\right] \tag{D.8a}$$

$a<b$ のとき，

$$\left(\frac{C_n}{C_{n0}}\right)_{Newt}=\frac{3}{2}\sqrt{\frac{b}{a}}\left\{\frac{-a^2/b^2}{\left(1-a^2/b^2\right)^{3/2}}\log\left[\frac{b}{a}\left(1+\sqrt{1-\frac{a^2}{b^2}}\right)\right]-\frac{1}{1-a^2/b^2}+\frac{s}{b}-1\right\} \tag{D.8b}$$

と求められる．

D.4 軸力係数

-90 から 90 [deg]までの任意の迎角αにおける軸力係数 C_A は迎角 0 [deg]の軸力係数を C_{A0} とすると，

$$C_A=C_{A0}\cos^2\alpha \tag{D.9}$$

と表される．ここで軸力係数C_{A0}は，摩擦抗力係数C_{ASF}，ノーズの体積に基づく造波抗力係数C_{AW}，底面の圧力による軸力の係数C_{AB}で代表される 3 つの有害抗力の和として表される．

$$C_{A0}=\frac{C_{ASF}+C_{AW}+C_{AB}}{S_{ref}} \tag{D.10}$$

ここで，マッハ数M_∞が 1 未満であるときC_{AW}とC_{AB}はともに 0 であり，この 2 つはマッハ数M_∞が 1 以上のときのみ軸力として作用する．

摩擦抗力C_{ASF}は

$$C_{ASF}=C_f(Re)f(M_\infty)S_{WET} \tag{D.11a}$$

$$C_f(Re)=\frac{0.455}{\left(\log_{10}Re\right)^{2.58}} \quad \text{(Prandtl の式)} \tag{D.11b}$$

$$f(M_\infty)=\left(1+0.15M_\infty{}^2\right)^{-0.58} \quad \text{(Hoerner の式)} \tag{D.11c}$$

と計算される．上式は全面に乱流境界層，非圧縮性の摩擦係数C_fに平板境界層の関係を適用している．ここでS_{WET}はぬれ面積である．

マッハ数M_∞が1以上のとき，断面が楕円であるノーズの造波抗力係数C_{AW}は2次のスレンダボディ理論から次のように求められる．まず，

$$\dot{a}=\frac{\mathrm{d}a}{\mathrm{d}x},\quad \dot{b}=\frac{\mathrm{d}b}{\mathrm{d}x} \tag{D.12}$$

とすると，ノーズ断面積 S の増加量に対する造波抗力係数C_{AW}の増加量の割合$\mathrm{d}C_{AW}/\mathrm{d}S$が

$$\begin{aligned}\frac{\mathrm{d}C_{AW}}{\mathrm{d}S}=&\dot{a}\dot{b}(2\lambda+1)+\beta^2\dot{a}\dot{b}\left[3\dot{a}\dot{b}\lambda^2+\frac{3}{2}(\dot{a}^2+\dot{b}^2)\lambda-\frac{1}{2}(\dot{a}-\dot{b})^2+\frac{1}{2}\dot{a}\dot{b}\right]\\&+\dot{a}^2\dot{b}^2\left[(\gamma+1)\frac{M_\infty{}^4}{\beta^2}-(2+M_\infty{}^2)\lambda+\left(\frac{1}{4}M_\infty{}^2-1\right)\frac{\dot{a}^2+\dot{b}^2}{2\dot{a}\dot{b}}\right]\\&+M_\infty{}^2\dot{a}^2\dot{b}^2\left[\frac{3}{8}\frac{\dot{a}^2+\dot{b}^2}{\dot{a}\dot{b}}-(\lambda+1)+\frac{3}{2}\frac{\dot{a}\dot{b}}{\dot{a}^2-\dot{b}^2}\log\frac{\dot{a}}{\dot{b}}\right]\end{aligned} \tag{D.13}$$

より求められる．なお，

$$\lambda=\log\frac{4}{\beta(a+b)}-1,\quad \beta=\sqrt{M_\infty{}^2-1} \tag{D.14}$$

とする．よって，造波抗力係数C_{AW}は

$$C_{AW}=\int_0^{x_N}\frac{\mathrm{d}C_{AW}}{\mathrm{d}S}\frac{\mathrm{d}S}{\mathrm{d}x}\mathrm{d}x \tag{D.15}$$

と求められる．

また，マッハ数M_∞が1以上のとき，底面から発生する軸力係数C_{AB}は

$$C_{AB}=-\frac{2}{\gamma M_\infty{}^2}\left\{\left(\frac{2}{\gamma+1}\right)^{1.4}\left(\frac{1}{M_\infty}\right)^{2.8}\left[\frac{2\gamma M_\infty{}^2-(\gamma-1)}{\gamma+1}\right]-1\right\}S_N \tag{D.16}$$

より計算する．ここでγは比熱比であり1.40とする．

D.5 離散近似

前節までの推算式を用いれば任意のマッハ数と迎角の揚力係数，抗力係数を得ることができるが，しかし，本論文ではスペースプレーンの機体形状と上昇飛行経路の最適化問題において空力解析分野と軌道計画分野を分けるために，マッハ数とレイノルズ数，迎角に依存する揚力と抗力を幾つかの代表点で離散化する必要があった．まず，機体はあるマッハ数において大迎角を取らないという前提の下で，揚力係数を 2 次まで，抗力係数を 3 次項までテーラ展開すると，揚力係数と抗力係数はマッハ数とレイノルズ数に依存する 3 個のパラメータ C_{N1}，C_{N2}，C_{A0} を使って次のように表すことができる．

$$C_L = (2C_{N1} - C_{A0})\alpha + C_{N2}\alpha^2 \tag{D.17a}$$

$$C_D = C_{A0} + (2C_{N1} - \frac{3}{2}C_{A0})\alpha^2 + C_{N2}\alpha^3 \tag{D.17b}$$

ここで，C_{N2} に含まれるパラメータ C_{dn} は横断流マッハ数，横断流レイノルズ数の関数であるため迎角に依存する．そこで C_{dn} も迎角についてテーラ展開したところ，$\alpha = 0$[deg] の値である 1.2 とすればよいことが分かる．

パラメータ C_{N1}，C_{N2}，C_{A0} を求める代表マッハ数 M とレイノルズ数 Re の組み合わせはそれぞれ表D.1に示す 5 つとする．サンプリング点 i でのパラメータを沿え字によって C'_{N1}，C'_{N2}，C'_{A0} と表すことにする．マッハ数 M はエンジン切り替えマッハ数を参考に，レイノルズ数 Re は最適解として得られるであろう機体の大きさ，飛行速度と高度の関係を参考に決めた．

実際に，あるマッハ数 M，迎角 α の揚力係数と抗力係数が必要になった時，そのマッハ数 M を挟む 2 つのサンプリングマッハ数を 5 つの中から選び，それらのマッハ数のパラメータ C'_{N1}，C'_{N2}，C'_{A0} を使った式(D.17a)，(D.17b)に迎角 α を代入して揚力係数と抗力係

表D.1 サンプリングマッハ数とレイノルズ数

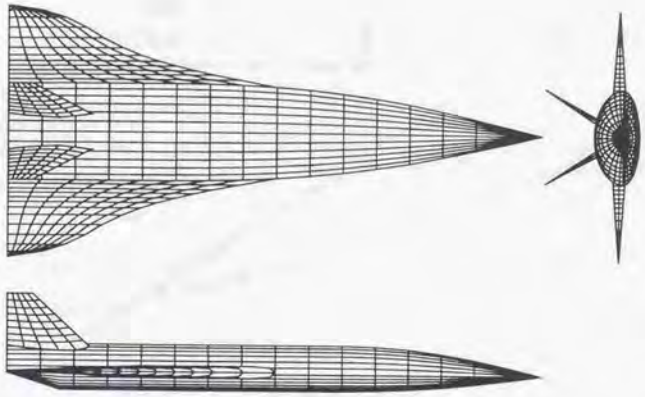
サンプリング点 i	マッハ数 M	レイノルズ数 Re
1	0.5	1×10^8
2	1.5	1×10^8
3	6.0	1×10^7
4	12.0	1×10^6
5	20.0	1×10^5

数を求め，マッハ数 M の揚力係数と抗力係数の値を直線補間により求める．ただし，マッハ数 0.8 以下の亜音速域では代表マッハ数 0.5 の揚力係数と抗力係数値，マッハ数 1.0 から 1.5 の超音速域では代表マッハ数 1.5 の係数値，またマッハ数 20 以上では代表マッハ数 20 の係数値で近似する．

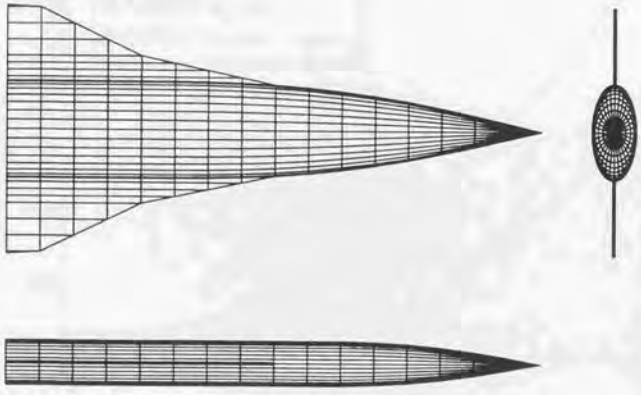
D.6 風洞試験データとの比較

本節では，CRSFLW による空力係数を風洞試験データと比較し，パラメータの決定と解析法の妥当性を調べる．ここでは機体モデルとして NAL0 次形状を適用する．文献37の中でこの機体モデルは亜音速域から極超音速域まで広く風洞試験と空力解析が行われている．NAL0 次形状を図D.4に示す．CRSFLW では平板の中翼を有する楕円断面の胴体を持つ機体の空気力を求めることができる．よって，図D.5のように NAL0 次形状を簡略化した機体モデルに空力解析法を適用した．なお，揚力係数，抗力係数のための基準面積 S_{ref} は $0.13l^2$ である．

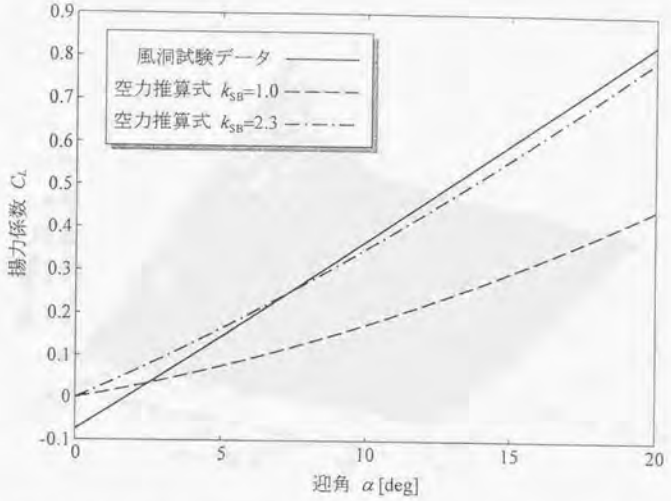
図D.6と図D.7で，揚力係数と抗力係数について亜音速域での風洞試験と CRSFLW の値を比較した．k_{SB} =1.0 である文献60に基づく推定では法線力を小さく見積もりすぎているため，風洞試験結果と比べると CRSFLW の揚力係数と抗力係数の値はともに小さい．亜音速時の揚力係数の値は離陸時に必要な翼の大きさに影響を与えるため特に重要である．そこで，風洞試験データに近づけるためにパラメータ k_{SB} を 2.3 と決定した．図D.8から図D.11において，このパラメータ値での CRSFLW による揚力係数と抗力係数を風洞試験の結果と比較している．遷音速域から超音速域にかけて多少の違いは見られるものの，スペースプレーンの飛行の大部分を占める極超音速域において両者は良く一致している．



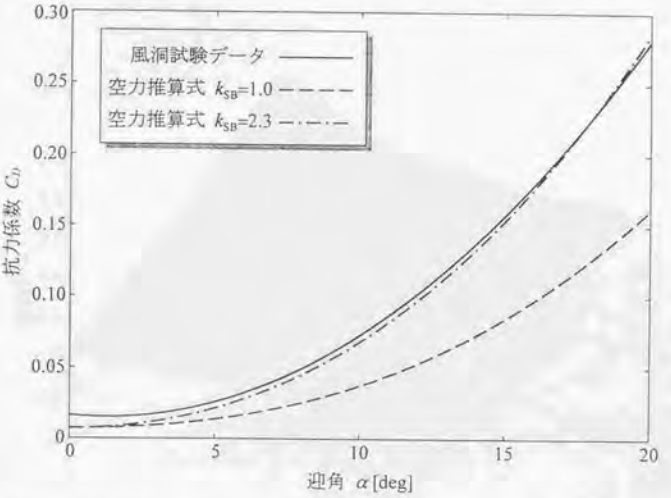
図D.4　NAL0 次形状



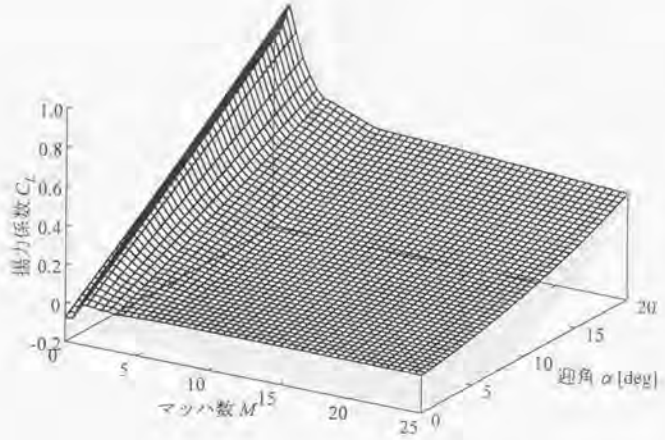
図D.5　本空力解析法を適用した機体モデル



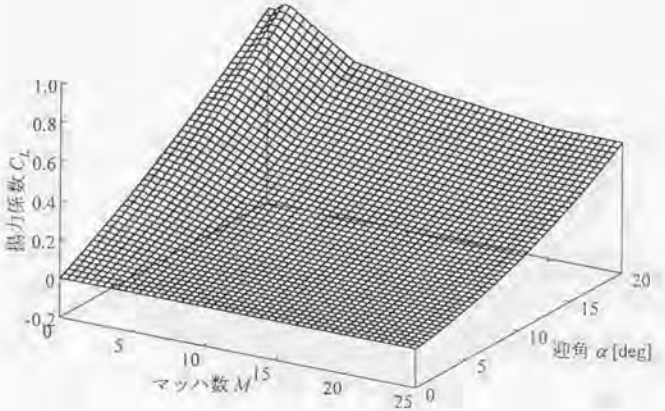
図D.6　亜音速域での風洞試験データと本空力解析法の比較（揚力係数）



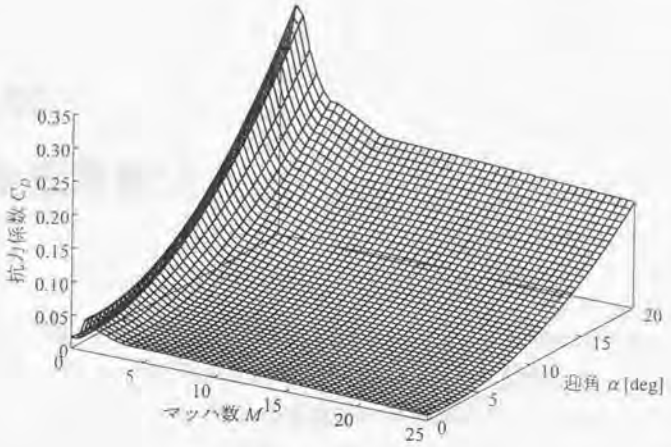
図D.7　亜音速域での風洞試験データと本空力解析法の比較（抗力係数）



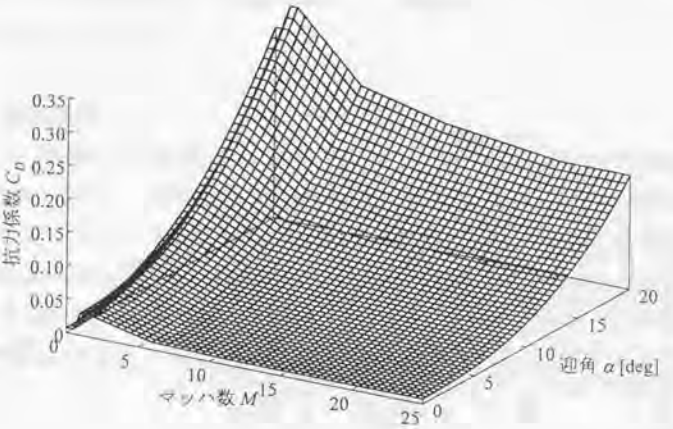
図D.8　風洞試験による揚力係数



図D.9　CRSFLW による揚力係数



図D.10　風洞試験による抗力係数



図D.11　CRSFLW による抗力係数

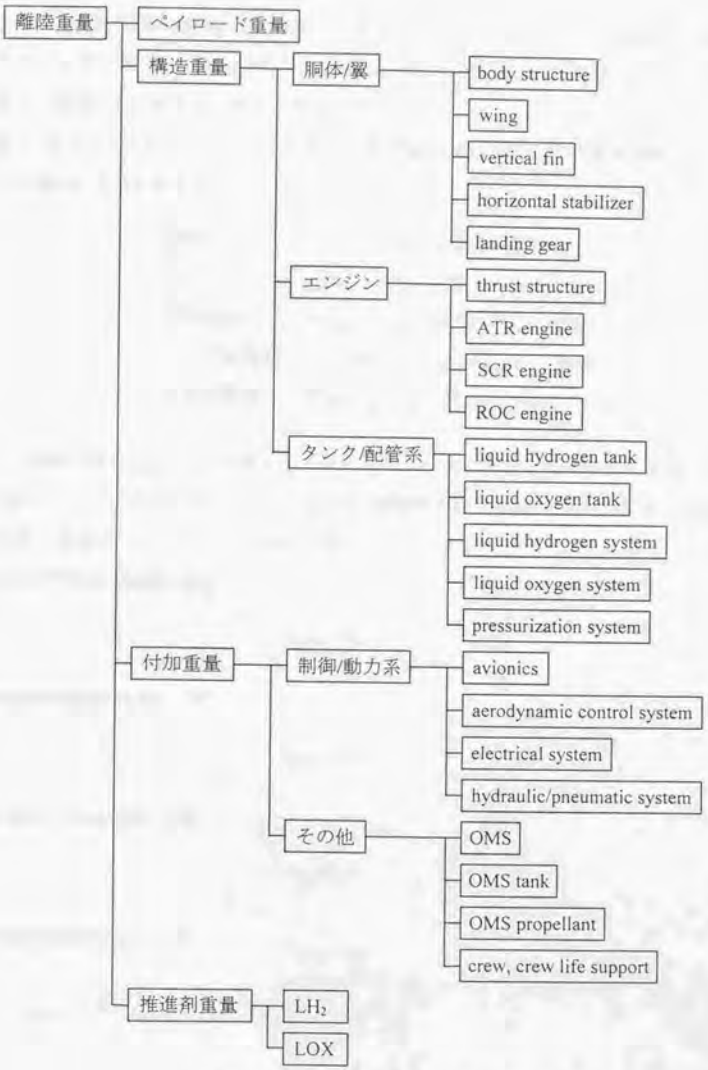
補遺E

重量推算法“WAATS”

本章では，第6章で使用した重量算出式 WAATS（A Computer Program for Weight Analysis of Advanced Transportation System）[64]による機体構造重量と搭載可能ペイロード重量の推算法について説明する．WAATS は，有翼高速機の重量評価のためのプログラムであるが，基本的には経験則によるプログラムであるため，これまでに実在した航空機等の重量データに基づく．だが，その出版時期から分かる通り，主として 1970 年代半ばまでの航空機により成立しており，スペースシャトルすらそのデータに含まれておらず，WAATS の適用には限界があるといえる．しかし，重量推算には非常に便利であるので，構造材料等，現在までの技術の進歩を考え，文献36を参考にして係数を更新することにより，このプログラムを用いることにする．

E.1　構成要素

WAATS は機体を構成要素に分割し，それぞれの重量を積算することで離陸重量を推算する方法をとっている．図E.1に WAATS をもとに本論文で考慮されている機体重量の構成要素を示す．ここで，離陸時の機体重量である離陸重量（takeoff weight）はペイロード重量（payload weight），構造重量（structure weight），付加重量（auxiliary weight），推進剤重量（propellant weight）に分けている．スペースプレーンの重量推算を行うという問題の性質上，オリジナルの WAATS とは幾つかの点で変更が加えられている．



図E.1 本論文で考慮した構成要素

E.2 設計変数とパラメータ

以下の重量推算式の適用に際し単位に注意を払わなくてはならない．WAATS の原文の単位は ft-lb 系であるが，WAATS にない新しい項目の単位は SI 単位系である．以後，計算項目，変数と共にそれらの単位も同時に示す．

第 6 章のスペースプレーンの機体形状設計問題において次の設計変数が扱われた．記号の定義は6.1.1項を参照されたい．

胴体サイズ	l_1 [ft]，l_4 [ft]，a [ft]，b [ft]
翼サイズ	l_2 [ft]，l_3 [ft]，s [ft]
飛行性能	$n_{LF\max}$，$q_{\max}$ [lbf/ft^2]，$T_{\max}$ [lbf]
エンジン性能	S_{ATR} [m^2]，S_{SCR} [m^2]，T_{ROC} [kgf]
推進剤重量	$W_{ATR+SCR}$ [kg]，$W_{propellant}$ [kg]

また，離陸重量$W_{takeoff}$ [lb]が定数として与えられていた．まず，WAATS を適用するために必要なパラメータを計算しておく．ここで，推進剤である LOX と LH_2 の混合比 LOX/LH_2 は ATR と SCR では 0，ROC では 6 とする．

body reference length [ft]

$$l_{BODY} = l_4 \tag{E.1}$$

maximum body height [ft]

$$h_{BODY} = 2a \tag{E.2}$$

geometric wing span [ft]

$$s_G = 2s \tag{E.3}$$

wing structural span [ft]

$$s_{ST} = s_G \sqrt{1 + \left\{ \frac{l_3 - l_2}{2(s-b)} \right\}^2} \tag{E.4}$$

wing thickness at theoretical root [ft] （仮定）

$$t_{ROOT} = 0.05 \times (l_3 - l_2) \tag{E.5}$$

theoretical wing area per airplane [ft^2]

$$S_W = (l_3 - l_2)(s - b) \tag{E.6}$$

total body wetted area [ft^2]

$$S_{BODY} = A_S + \pi\sigma(a+b)(l_4 - l_1) + \pi ab \tag{E.7}$$

ここで，胴体の濡れ面積S_{BODY}を求めるために必要なノーズの表面積A_S[ft^2]，パラメータσの計算については補遺Dを参考にされたい．

LH_2重量 [kg]

$$W_{LH2} = W_{ATR+SCR} + 0.1429 \times \left(W_{propellant} - W_{ATR+SCR}\right) \tag{E.8}$$

LOX 重量 [kg]

$$W_{LOX} = 0.8571 \times \left(W_{propellant} - W_{ATR+SCR}\right) \tag{E.9}$$

total volume of liquid hydrogen tank [ft^3]

$$V_{LHTK} = 40577 \times W_{LH2} \tag{E.10}$$

total volume of liquid oxygen tank [ft^3]

$$V_{LOTK} = 2507 \times W_{LOX} \tag{E.11}$$

thrust of OMS [kgf] （仮定）

$$T_{OMS} = 5.0 \tag{E.12}$$

number of crew （仮定）

$$n_{CREW} = 5 \tag{E.13}$$

E.3 重量推算式

機体の構成要素の重量推算式を示す.

basic body structure [lb]

$$W_{\mathrm{BASIC}} = 0.256 \times ((l_{\mathrm{BODY}} \times n_{LF\max} / h_{\mathrm{BODY}})^{0.15} \times q_{\max}{}^{0.16} \times S_{\mathrm{BODY}}{}^{1.05}) + 0.735 \times S_{\mathrm{BODY}} \tag{E.14}$$

wing [lb]

$$W_{\mathrm{WING}} = 2324.0 \times (W_{\mathrm{takeoff}} \times n_{LF\max} \times s_{\mathrm{ST}} \times S_W / t_{\mathrm{ROOT}})^{0.608} \times 10^{-6} \tag{E.15}$$

vertical fin [lb] （仮定）

$$W_{\mathrm{VERT}} = 0 \tag{E.16}$$

horizontal stabilizer [lb] （仮定）

$$W_{\mathrm{HORZ}} = 0 \tag{E.17}$$

landing gear [lb]

$$W_{\mathrm{LG}} = 0.31 \times W_{\mathrm{takeoff}}{}^{0.795} \tag{E.18}$$

thrust structure [lb]

$$W_{\mathrm{THRST}} = 10.25 \times 10^{-3} \times T_{\max} \tag{E.19}$$

ATR engine [kg]

$$W_{\mathrm{ATR}} = 1150 \times S_{\mathrm{ATR}} \tag{E.20}$$

SCR engine [kg]

$$W_{\mathrm{SCR}} = 1000 \times S_{\mathrm{SCR}} \tag{E.21}$$

ROC engine [kg]

$$W_{\mathrm{ROC}} = 1.3 \times 10^{-2} \times T_{\mathrm{ROC}} \tag{E.22}$$

liquid hydrogen tank [lb]

$$W_{\mathrm{LHTK}} = 0.53 \times V_{\mathrm{LHTK}} \tag{E.23}$$

liquid oxygen tank [lb]

$$W_{\mathrm{LOTK}} = 1.25 \times V_{\mathrm{LOTK}} \tag{E.24}$$

liquid hydrogen system [lb]

$$W_{\mathrm{LHSYS}} = 0.003 \times T_{\max} + 1.5 \times l_{\mathrm{BODY}} \tag{E.25}$$

liquid oxygen system [lb]

$$W_{\mathrm{LOSYS}} = 0.001 \times T_{\max} + 1.3 \times l_{\mathrm{BODY}} \tag{E.26}$$

pressurization system [lb]

$$W_{\mathrm{PRSYS}} = 0.09 \times V_{\mathrm{LHTK}} + 0.49 \times V_{\mathrm{LOTK}} \tag{E.27}$$

avionics [lb]

$$W_{\mathrm{AVONC}} = 33.185 \times W_{\mathrm{takeoff}}{}^{0.361} \tag{E.28}$$

aerodynamic control system [lb]

$$W_{\mathrm{AERO}} = 0.323 \times (W_{\mathrm{takeoff}}{}^{0.667} \times (l_{\mathrm{BODY}} + s_{\mathrm{G}})^{0.25})^{0.903} \tag{E.29}$$

electrical system [lb]

$$W_{\mathrm{ELECT}} = 1.167 \times W_{\mathrm{takeoff}}{}^{0.5} \times l_{\mathrm{BODY}}{}^{0.25} \tag{E.30}$$

hydraulic/pneumatic system [lb]

$$W_{\mathrm{HYPNU}} = 2.64 \times (0.001 \times S_W \times q_{\max})^{0.334} \times (l_{\mathrm{BODY}} + s_{\mathrm{ST}})^{0.5} \tag{E.31}$$

OMS [kg]

$$W_{\mathrm{OMS}} = 44 \times T_{\mathrm{OMS}} \tag{E.32}$$

OMS tank [kg]

$$W_{\mathrm{OMSTK}} = 0.1 \times W_{\mathrm{OMS}} \tag{E.33}$$

OMS propellant [lb] （仮定）

$$W_{\mathrm{OMSP}} = 3307 \tag{E.34}$$

crew, crew life support [lb]

$$W_{\mathrm{CREW}} = 5 \times 10^{-4} \times W_{\mathrm{takeoff}} + 220 \times n_{\mathrm{CREW}} + 250 \tag{E.35}$$

構造重量

$$\begin{aligned} W_{\mathrm{structure}} = {} & W_{\mathrm{BASIC}} + W_{\mathrm{WING}} + W_{\mathrm{VERT}} + W_{\mathrm{HORZ}} + W_{\mathrm{LG}} \\ & + W_{\mathrm{THRST}} + W_{\mathrm{ATR}} + W_{\mathrm{SCR}} + W_{\mathrm{ROC}} \\ & + W_{\mathrm{LHTNK}} + W_{\mathrm{LOTNK}} + W_{\mathrm{LHSYS}} + W_{\mathrm{LOSYS}} + W_{\mathrm{PRSYS}} \end{aligned} \tag{E.36}$$

ここで，W_{ATR}，W_{SCR}，W_{ROC} の 3 つは単位が[kg]であることに注意する．

付加重量 [lb]

$$\begin{aligned} W_{\mathrm{auxiliary}} = {} & W_{\mathrm{AVONC}} + W_{\mathrm{AERO}} + W_{\mathrm{ELECT}} + W_{\mathrm{HYPNU}} \\ & + W_{\mathrm{OMS}} + W_{\mathrm{OMSTK}} + W_{\mathrm{OMSP}} + W_{\mathrm{CREW}} \end{aligned} \tag{E.37}$$

E.4 ペイロード重量

第 6 章で必要なペイロード重量 W_{payload} は離陸重量 W_{takeoff}，構造重量 $W_{\mathrm{structure}}$，推進剤消費重量 $W_{\mathrm{propellant}}$，付加重量 $W_{\mathrm{auxiliary}}$ から次のように計算される．

$$W_{\mathrm{payload}} = W_{\mathrm{takeoff}} - \left(W_{\mathrm{structure}} + W_{\mathrm{propellant}} + W_{\mathrm{auxiliary}} \right) \tag{E.38}$$

補遺F

PCによる並列計算の実現法

本研究では必要な並列計算を行うために並列計算機を使用することなく，ネットワーク接続された複数台のパーソナルコンピュータ（PC）によって数値計算を行った．また，実行時間を計測するときには通信時間を短くし，個々の計算が同じ性能の計算機で行われなければならないことから，1台のPCのみで並列計算を模擬した．本章ではPC 1台で並列計算を模擬する方法と，ネットワーク結合されたPCによって並列計算を行う簡単な方法を示す．ここで問題点となるのは複数の計算プロセスの実行を開始する方法と，これらプロセス間でデータを交換し，実行を同期させるための通信方法である．

F.1　1台のPCによる並列計算の模擬方法

現在最も普及しているMicrosoft Windowsをオペレーティングシステム（OS）としたPCが持つ機能の特徴の1つとして，マルチタスクが可能であることが挙げられる．複数のアプリケーションを同時に起動させることができ，それらはあたかも1台の計算機上で複数のアプリケーションが並列で動作するように見える．個々のアプリケーションの実行過程はプロセスと呼ばれている．一方，1つのアプリケーション内でも様々な処理を並列して行うことができる．このようなプロセス内のプロセスはスレッド（thread）と呼ばれる．よって，1台のコンピュータで並列計算を模擬するとき，個々の計算をプロセスレベルで実行するか，スレッドレベルで行うかのいずれかになる．

2つのうちプロセスレベルの実現は容易である．並列で進められる1つの計算を1つの実行可能ファイル（executable file）にしておき，計算を開始する時，1つ1つのファイルを（ダブルクリックして）手動で起動させれば良い．この方法は起動させるプロセスの数が多くなると煩わしくなるが，Windowsのアプリケーション開発者のために用意されて

いる API（Application Programming Interface）関数の中のプロセス制御に関する関数を使用すれば，複数の実行可能ファイルを1つのアプリケーションから自動的に起動させることができる．本論文の数値計算はこの方法で行った．並列最適化法は第2章で述べた通り1つのシステムレベルのプロセスと複数個の部分最適化問題を解くサブシステムレベルのプロセスからなる．システムプロセスのプログラムを起動させると，サブシステムプロセスのプログラムが自動的に起動するようにした．

次にプロセス間の通信方法についてはファイルマッピング（File Mapping）や，共有メモリ（Shared Memory）を使用する方法があるが，最も単純な方法はディスク（ハードディスク）上のファイルを媒介する方法である．2つのプロセスによるファイルの読み書きが競合したときの処理が必要なこと，ディスクを使用するためにメモリを使用する方法に比べて時間を要することが欠点である．

2番目のスレッドレベルで並列計算を実行する場合，個々の並列計算過程は1つのプログラム内で関数（サブルーチン）として組み込まれる．スレッドの開始には API 関数が使われる．またスレッド間の通信は同じプログラム中で行われるので変数，あるいは関数の引数によって簡単に行える．

以上の2つの方法によって行われる並列計算は，実際は通常1個しか存在しない CPU の処理動作をプロセスあるいはスレッドごとに定期的に割り当て，あたかも複数の処理が並列して動いているように見えるだけである．よって，プロセス間通信が存在しない完全に独立した n 個の全く同じプロセスを同時に実行させても，個々のプロセスの処理速度は $1/n$ になるため，速度向上率（第3章）は n にならず，1のままである．したがって，並列計算機を使用したときの実行時間を計算機1台のみの使用で求めるためには，並列で行われるはずの計算を逐次的に行い，最も実行時間が長かった計算の時間を，並列処理が行われた場合の実行時間とする．このとき，あるプロセスで計算が行われている間，他のプロセスは CPU の処理能力を使用してはならない．通常，ループ処理により自分のプロセスに計算の順番が回ってくるのを待っていると，それだけで CPU を使用していることになり，個々のプロセスの正確な実行時間が計測できない．そこで他のプロセスに CPU の処理能力が割り当てられるように，一定時間プロセスを停止するための API 関数を使用する．

F.2 複数台の PC による並列計算の方法

現在，パーソナルコンピュータの普及と共に，かつては 1 台 1 台独立していたコンピュータがネットワークに結合され，極めて自然に他のコンピュータとデータの共有が行えるようになった．これを利用して高価な並列計算機を使用しなくても，複数の PC で計算処理を分割して行うことが可能である．一般に，種々の計算機をネットワークで接続して構築された並列計算機はクラスタ型並列計算機と呼ばれ，現在急速に普及しつつある並列計算機の形態の 1 つである．ネットワーク結合された PC を並列計算機の一部として利用するためには専用の通信関数ライブラリが必要になる．特に有名なのが PVM（Parallel Virtual Machine），MPI（Message Passing Interface），AFAPI（Aggregate Function API）である．これらをインストールすればどのような PC でもクラスタ型並列計算機を構成することができ，非常に安価で，柔軟性が高い並列計算機を手に入れることができる．

一方，本研究で使用した計算機はすべて Microsoft Windows を OS とした PC である．この OS に組み込まれた既存の機能によって，それほど多くの並列度を必要としない計算であれば，LAN（Local Area Network）接続されたコンピュータにプロセスを振り分けて，並列計算を行うことが簡単に可能である．ただし，個々のプロセスは実行可能ファイルとして存在しており，計算を開始するためには，PC ごとに手動でプロセスを起動させなければならない．よって，並列度が増すと計算を開始することが非常に煩わしくなり，上述したライブラリが必要になってくる．

また，プロセス間通信の方法は幾つか考えられる．RPC（Remote Procedure Call）と呼ばれる機能を使用すると，あたかも自分自身のプログラムの一部である関数（サブルーチン）を呼ぶように，ネットワーク接続先の PC のルーチンを呼ぶことができる．RPC より低レベルでありユーザが通信プロトコルなどを決めなければならない方法として，ソケット（Socket），パイプ（Pipe）と呼ばれるネットワーク通信方法も存在する．最も単純な PC 間のデータ交換方法はフォルダ（ディレクトリ）の共有機能を使うことである．1 台の PC でプロセス間通信にディスク上のファイルを使うことを述べたが，Microsoft Windows には，他の PC のディスク上にあるフォルダが，あたかも自分の PC 上にあるかのようにその中にあるファイルを読み書きできる機能をもつ．よって，媒介となるファイルを決めておき，そのファイルを通してデータを交換すれば良い．並列度が増すと煩わしくなること，ディスクへの読み書きに時間がかかることが欠点として挙げられる．しかし本研究では最も単純なこの方法を採用した．

謝辞

本論文に関する研究を遂行するにあたり，多くの方々のご指導とご支援を賜りました．末筆ながらお世話になった方々に感謝の気持ちを述べたいと思います．

指導教官である鈴木真二教授には，修士課程入学時から5年間にわたり終始熱心且つ丁寧に，研究の方向性から，論文の書き方，講演会での発表法まで多くのことをご指導頂きました．深く感謝します．

助手の柄沢研治氏には研究室の重鎮として最良な研究環境を与えて頂き，また学生生活に喝を入れて頂きました．

技官の松永大一郎氏には計算機の使用法について有益なアドバイスを承りました．

また，著者は当研究室での5年間に多くの先輩，同輩，後輩の方々と議論を交わし，助言，励まし，刺激を受けてまいりました．

研究室外においても，講演会，シンポジウム，研究会等において，航空宇宙分野に関わる方々，最適化に関する研究をされている方々から貴重なご意見をたくさん賜りました．

今後も皆様方とのお付き合いは続くと思いますが，本論文の完成を一区切りとして深く感謝の念を述べたいと思います．

参考文献

1) Sobieszczanski-Sobieski, J. and Haftka, R. T.: Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments, AIAA 96-0711, 1996.

2) 加藤寛一郎: スペースプレーン, 東京大学出版会, 東京, 1989.

3) Lasdon, L. S.: Optimization Theory for Large Systems, The Macmillan Company, London, 1970.

4) Kirsch, U.: Optimum Structural Design, McGraw-Hill, New York, 1981.

5) Sobieszczanski, J. and Loendorf, D.: A Mixed Optimization Method for Automated Design of Fuselage Structure, J. of Aircraft, 9 (1972), pp. 805-811.

6) 鈴木克幸, 大坪英臣: 分散構造最適設計におけるシステム間の制約の導入, 計算工学講演会講演集 Vol. 2, 1997, pp. 613-616.

7) Sobieszczanski-Sobieski, J.: Recent Mathematical Methods for engineering System Design, Part 2: System Approach in Engineering Optimization, Applied Mathematics in Aerospace Science and Engineering, edited by Miele, A. and Salvetti, A., Plenum Press, New York, 1994, pp. 445-470.

8) Sobieszczanski-Sobieski, J.: Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems, NASA CP-3031 Part 1 (1989).

9) Kroo, I., Altus, S., Braun, R., Gage, P. and Sobieski, J.: Multidisciplinary Optimization Methods for Aircraft Preliminary Design, AIAA 94-4325 1994.

10) Braun, R. D.: Collaborative Optimization: An Architecture for Large-Scale Distributed Design, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Stanford University, Stanford, 1996.

11) Braun, R. D. and Kroo, I. M.: Development and Application of the Collaborative Optimization Architecture in Multidisciplinary Design Environment, Multidisciplinary Design Optimization:

State of the Art, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 98-116.

12) 船橋博人: 設計統合化・最適化支援ソフト「iSIGHT」「Computer-Aided Optimization: CAO」実現へ向けて, 計算工学, 4 (1999), pp. 102-105.

13) Balling, R. J. and Wilkinson, C. A.: Execution of Multidisciplinary Design Optimization Approaches on Common Test Problems, AIAA J., 35 (1997), pp. 178-186.

14) 岩壷卓三, 河村庄造, 安達和彦,: 機械構造物の構造系と制御系の同時最適設計に関する研究動向と今後の課題, 日本機械学会論文集（C編）, 59 (1993), pp. 631-637.

15) Livne, E., Schmit, L. A. and Friedmann, P. P.: Integrated Structure/Control/Aerodynamic Synthesis of Actively Controlled Composite Wings, J. of Aircraft, 30 (1993), pp. 387-394.

16) Rowell, L. F., Braun, R. D., Olds, J. R. and Unal: Multidisciplinary Conceptual Design Optimization of Space Transportation Systems, J. of Aircraft, 36 (1999), pp. 218-226.

17) Braun, R. D., Moore, A. A. and Kroo, I. M.: Collaborative Approach to Launch Vehicle Design, J. of Spacecraft and Rockets, 34 (1997), pp. 478-486.

18) Xavier, C. and Iyengar, S. S.: Introduction to Parallel Algorithms, A Wiley-Interscience Publication, New York, 1998.

19) 渡辺勝正: 並列処理概説, コロナ社, 東京, 1991.

20) Schnabel, R. B.: A View of the Limitations, Opportunities, and Challenges in Parallel Nonlinear Optimization, Parallel Computing, 21 (1995), pp. 875-905.

21) Cramer, E. J., Dennis, J. E., Jr., Frank, P. D., Lewis, R. M. and Shubin, G. R.: Problem Formulation for Multidisciplinary, SIAM J. on Optimization, 4 (1994), pp. 754-776.

22) Michelena, N. F. and Papalambros, P. Y.: Optimal Model-Based Decomposition of Powertrain System Design, J. of Mechanical Design, 117 (1995), pp. 499-505.

23) 森正武: 数値解析法, 朝倉書店, 東京, 1984.

24) Tapia, R. A.: Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization, J. of Optimization Theory and Applications, 22 (1997), pp. 135-194.

25) 伊理正夫, 藤野和建: 数値計算の常識, 共立出版株式会社, 東京, 1985.

26) Greenstadt, J.: Variations on Variable-Metric Methods, Mathematics of Computation, 24 (1970), pp. 1-22.

27) Dennis, J. E., Jr. and More, J. J.: Quasi-Newton methods: Motivation and theory, SIAM Review, 19 (1977), pp. 46-89.

28) Amdahl, G. M.: Validity of the single processor approach to achieving large scale computing capabilities, AFIPS conference proceedings, 1967 Spring Joint Computer Conference, 1967, pp.483-485.

29) Bryson, A. E., Jr. and Ho, Y. C.: Applied Optimal Control, Blaisdell Publishing Company, MA, 1969.

30) 加藤寛一郎, 大屋昭男, 柄沢研治: 航空機力学入門, 東京大学出版会, 東京, 1982.

31) 牧野光雄: 航空力学の基礎（第2版）, 産業図書, 東京, 1989.

32) Freeman, D. C., Talay T. A., Stanley, D. O., Lepsch, R. A., and Wilhite, A. W.: Design Options for Advanced Manned Launch Systems, J. of Spacecraft and Rockets, 32 (1995), pp. 241-249.

33) Berry, W. and Grallert, H.: Performance and Technical Feasibility Comparison of Reusable Launch Systems: A Synthesis of the ESA winged Launcher Studies, Acta Astronautica, 38 (1996), pp. 333-347.

34) Koelle, D. E.: Economics of Small Fully Reusable Launch Systems (SSTO vs. TSTO), Acta Astronautica, 40 (1997), pp. 535-544.

35) Freeman, D. C., Talay T. A., Stanley, D. O.: Single-Stage-To-Orbit - Meeting the Challenge, Acta Astronautica, 38 (1996), pp. 323-331.

36) 白水正男: 宇宙往復機の重量評価と感度解析（その1）SSTO 第1報, 航空宇宙技術研究所資料 TM-598, 1989.

37) Nomura, S., Hozumi, K., Kawamoto, I. and Miyamoto, Y.: Experimental Studies on Aerodynamic Characteristics of SSTO Vehicle at Subsonic to Hypersonic Speeds, The 16th International Symposium on Space Technology and Science, 1988, pp. 1547-1554.

38) Sakata, K., Minoda, M., Yanagi, R. and Nouse, H.: Hypersonic Turbomachinery-Based Air-Breathing Engines for the Earth-to-Orbit Vehicle, J. of Propulsion and Power, 7 (1991), pp. 108-114.

39) 升谷五郎, 若松義男: スクラムジェットの性能計算, 航空宇宙技術研究所資料 TR-987 (1988).

40) 国立天文台編: 理科年表（平成10年版）, 丸善, 東京, 1998.

41) Vanderplaats, G. N.: Numerical Optimization Techniques for Engineering Design, McGraw-Hill, New York, 1984.

42) 坂和正敏: 非線形システムの最適化, 森北出版株式会社, 東京, 1986.

43) 伊理正夫, 今野浩, 刀根薫: 最適化ハンドブック, 朝倉書店, 東京, 1995.

44) ASNOP 研究会編: パソコン FORTRAN 版 非線形最適化プログラミング, 日刊工業新聞社, 東京, 1991.

45) 茨木俊秀, 福島雅夫: FORTRAN77 最適化プログラミング, 岩波書店, 東京, 1991.

46) Han, S. P.: Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, Mathematical Programming, 11 (1976), pp. 263-282.

47) Han, S. P.: A Globally Convergent Method for Nonlinear Programming, J. of Optimization Theory and Applications, 22 (1977), pp. 297-309.

48) Powell, M. J. D.: Variable Metric methods for constrained optimization, Mathematical Programming, The State of the Art, edited by Bachem, A., Grotshel, M. and Korte, B., Springer, Berlin, pp. 288-311, 1983.

49) Goldfarb, D. and Idnani, A.: A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs, Mathematical Programming, 27 (1983), pp. 1-33.

50) Schmit Jr., L. A. and Ramanathan, R. K.: Multilevel Approach to Minimum Weight Design including Buckling Constraints, AIAA J., 16 (1978), pp. 97-104.

51) Sobieszczanski-Sobieski, J., Barthelemy, J. F. and Riley, K. M.: Sensitivity of Optimum Solutions of Problem Parameters, AIAA J., 20 (1982), pp. 1291-1299.

52) Kirsch, U.: An Improved Multilevel Structural Synthesis Method, J. Structural Mechanics, 13 (1985), pp. 123-144.

53) Haftka, R. T.: An Improved Computational Approach for Multilevel Optimum Design, J. Structural Mechanics, 12 (1984), pp. 245-261.

54) 鈴木克幸, 大坪英臣: 多段階最適化手法による船体構造最適設計, 日本造船学会論文集, 178 (1995), pp. 405-411.

55) Batill, S. M., Stelmack, M. A. and Sellar, R. S.: Framework for Multidisciplinary Design Based on Response-Surface Approximation, J. of Aircraft, 36 (1999), pp. 287-297.

56) Bryson, A. E., Jr. and Ho, Y. C.: Applied Optimal Control, Blaisdell Publishing Company, MA, 1969.

57) Betts, J. T.: Survey of Numerical Methods for Trajectory Optimization, J. of Guidance, Control, and Dynamics, 21 (1998), pp. 193-207.

58) Hull, D. G.: Conversion of Optimal Control Problems into Parameter Optimization Problems, J. of Guidance, Control, and Dynamics, 20 (1997), pp. 57-60.

59) Jorgensen, L. H.: Prediction of Static Aerodynamic Characteristics for Space-Shuttle-Like and Other Bodies at Angle of Attack from 0° to 90° , NASA TN D-6996 (1973).

60) Jorgensen, L. H.: A Method for Estimating Static Aerodynamic Characteristics for Slender Bodies of Circular and Noncircular Cross Section Alone and with Lifting Surfaces at Angles of Attack from 0° to 90° , NASA TN D-7228 (1973).

61) Mendenhall, M. R., Goodwin, F. K., Dillenius, M. F. E. and Kline, D. M.: Computer Program for Calculating the Static Longitudinal Aerodynamic Characteristics of Wing-Body-Tail Configurations, NASA CR-2474 (1975).

62) 山名正夫，中口博: 飛行機設計論，養賢堂，東京, 1978.

63) 吉田憲司: 超音速旅客機の空力形状に関する要素研究について，日本航空宇宙学会誌，42 (1994)，pp. 403-415.

64) Glatt, G. R.: WAATS - A Computer Program for Weights Analysis of Advanced Transportation Systems, NASA CR-2420 (1974).

関連文献

学会誌論文

- 土屋武司, 鈴木真二: 数理計画法を用いた最適制御問題解法に関する研究（その１）感度微分方程式の導入,日本航空宇宙学会誌, 45 (1997), pp. 231-237.
- 土屋武司, 鈴木真二: 数理計画法を用いた最適制御問題解法に関する研究（その２）ブロック対角ヘシアン法の提案, 日本航空宇宙学会誌, 46 (1998), pp. 497-503.
- 土屋武司, 鈴木真二: スペースプレーンの機体設計と飛行経路の同時最適化に関する数値解法, 日本航空宇宙学会誌, 46 (1998), pp. 346-353.

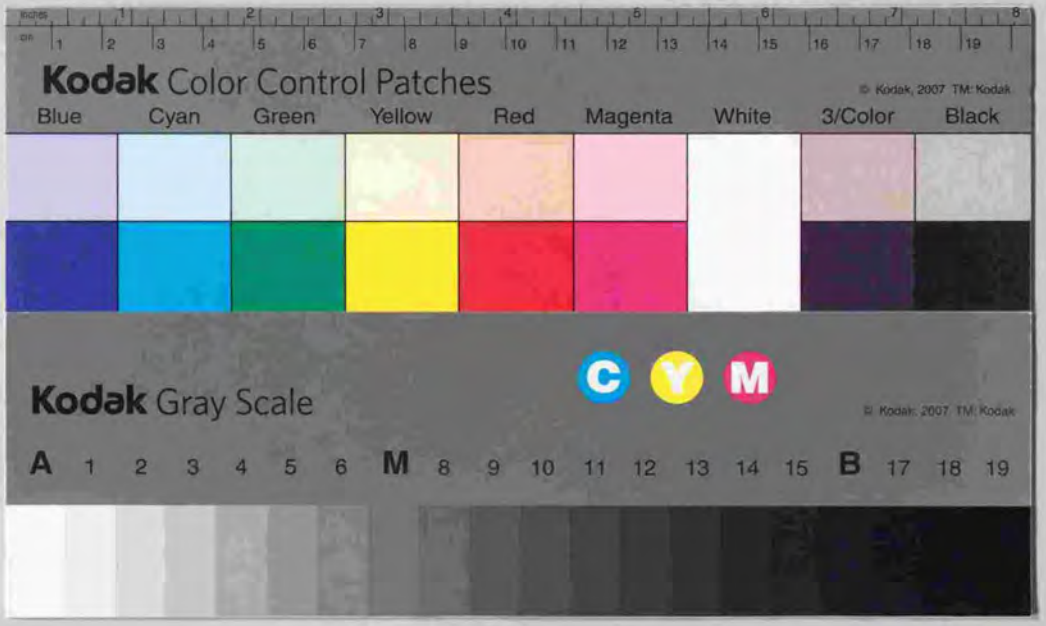
国際会議発表論文

- Tsuchiya, T. and Suzuki, S.: A Study on Simultaneous Design/Trajectory Optimization of Spaceplane, International Symposium on Optimization and Innovative Design, 1997, Paper#158.
- Tsuchiya, T. and Suzuki, S.: Spaceplane Trajectory Optimization with Vehicle Size Analysis, 14th IFAC Symposium on Automatic Control in Aerospace, 1998, pp. 444-449.
- Tsuchiya, T. and Suzuki, S.: Parallel Optimization for Large-scale Design and Control Problems, Proceeding of The 36th Aircraft Symposium, 1998, pp. 709-712.
- Tsuchiya, T. and Suzuki, S.: An Integrated Method for Shape and Flight Trajectory Optimization of Spaceplane, Proceeding of The 37th Aircraft Symposium, 1999, pp. 733-736.

国内口頭発表論文

- 土屋武司, 鈴木真二: 数理計画法を用いた最適制御問題解法に関する一考察, 日本航空宇宙学会第 27 期年会講演会講演集, 1996, pp. 158-159.

- 土屋武司, 鈴木真二: スペースプレーンの機体設計／軌道の同時最適化問題解法について, 第 34 回飛行機シンポジウム講演集, 1996, pp. 367-370.
- 土屋武司, 鈴木真二: 非線形計画法による最適制御問題の一解法, 日本航空宇宙学会第 28 期年会講演会講演集, 1997, pp. 72-73.
- 土屋武司, 鈴木真二: 軌道最適化を含む動的システムの最適設計, 計算工学講演会論文集, 1997, pp. 869-872.
- 土屋武司, 鈴木真二: 最適制御問題に対する分散処理の適用, 日本航空宇宙学会第 29 期年会講演会講演集, 1998, pp. 44-47.
- 土屋武司, 鈴木真二: 大規模最適制御問題の分散処理に向けて, 計算工学講演会論文集, 1998, pp. 683-686.
- 土屋武司, 鈴木真二: スペースプレーンの上昇軌道に対する並列最適化計算法, 第 42 回宇宙科学技術連合講演会講演集, 1998, pp. 1227-1232.
- 土屋武司, 鈴木真二: 宇宙往還機の概念設計と軌道の統合的最適化法, 第 48 回理論応用力学講演会講演論文集, 1999, pp. 227-228.
- 土屋武司, 鈴木真二: 再使用型 SSTO の概念設計に対する統合的最適化, 日本航空宇宙学会第 30 期年会講演会講演集, 1999, pp. 205-208.
- 土屋武司, 鈴木真二: スペースプレーンの機体形状と飛行軌道の統合的最適化法, 航空宇宙数値シミュレーション技術シンポジウム '99, p. 39.


Kodak Color Control Patches
© Kodak, 2007 TM: Kodak
Blue
Cyan
Green
Yellow
Red
Magenta
White
3/Color
Black
Kodak Gray Scale
C
Y
M
A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19