

変数係数線型常微分方程式系に対する並列計算解法の研究

Parallel Numerical Solutions of Linear
Variable-coefficient System of Ordinary Differential
Equations

江崎 信行

目次

第1章	序論	1
1.1	背景と目的	1
1.2	本研究の概要	3
1.3	本論文の構成	4
第2章	並列計算環境と計算手法	7
2.1	並列計算アーキテクチャ	7
2.2	プロセスの流れとデータ分割	10
2.3	並列化言語	12
2.4	微分方程式の数値解法に適用する並列化技法	14
2.5	並列化効果の指標	15
2.6	並列計算量の評価	16
第3章	移流拡散方程式とその数値解	19
3.1	移流拡散方程式が表わす現象	19
3.2	モデル問題	20
3.3	従来の結果と最近の研究	22
3.4	適用する離散変数法の収束性・安定性	25
3.4.1	収束性解析	26
3.4.2	A 安定性解析	28
3.4.3	NA 安定性解析	30
3.5	変数係数線型常微分方程式系と準解析的数値解法	31
第4章	移流拡散方程式の特性曲線差分法	33
4.1	特性曲線法による空間離散化	33
4.1.1	特性曲線に沿った格子	33
4.1.2	特性曲線に沿った空間差分法	34
4.2	数値スキームの収束性と安定性	36
4.2.1	局所離散化誤差	37
4.2.2	大域誤差	37
4.2.3	収束性の評価	40
4.2.4	数値的安定性の基準	40
4.3	方法および解析の数値的検証	42

4.3.1	古典的 Runge-Kutta 法の適用	42
4.3.2	係数行列の固有値	42
4.3.3	安定性条件を満たす刻み幅	44
4.3.4	収束性の数値的確認	45
4.3.5	硬度比の動向	45
4.4	陽的解法による並列化実験と陰的解法の適用の展望	46
4.4.1	並列化実験	47
4.4.2	陰的 Runge-Kutta 法の適用	48
4.5	本章のまとめ	50
第 5 章	変数係数線型系に対する並列化向き ROW 型解法	53
5.1	陰的 Runge-Kutta 法による試み	53
5.2	ROW 型の近似公式	54
5.3	3 次公式のパラメタ決定	56
5.3.1	次数条件式	57
5.3.2	2 次および 3 次の公式	58
5.3.3	対角化可能および NA 安定性のための条件	59
5.3.4	パラメタの決定	60
5.4	ROW 型解法の並列化	61
5.5	数値スキームの収束性と安定性	62
5.5.1	収束性解析	64
5.5.2	NA 安定性解析	65
5.6	数値実験	69
5.6.1	数値的収束性の検証	70
5.6.2	並列計算による速度向上度	70
5.7	本章のまとめ	72
第 6 章	ブロック Rosenbrock 法とその並列化	75
6.1	ブロック Rosenbrock 法の一般的な記述と高次達成のための拡張	75
6.2	4 次公式の構成	77
6.2.1	次数条件式	79
6.2.2	パラメタの決定	80
6.3	大域収束性	82
6.4	NA 安定性解析	83
6.5	数値実験	86
6.5.1	数値的収束性の検証	87
6.5.2	並列計算による速度向上度	87
6.6	移流拡散方程式の特性曲線差分法への適用の検討	90
6.7	本章のまとめ	91

第7章	移流拡散問題に対するブロック Rosenbrock 法の並列化数値実験	93
7.1	モデル問題	93
7.2	特性曲線差分法の適用	94
7.3	適用する離散変数法	95
7.4	実験結果	96
7.4.1	収束性の数値的確認	96
7.4.2	安定性と逐次実行時間	98
7.4.3	並列実行時間と効果	100
7.5	本章のまとめ	101
第8章	結論	103
8.1	本論文のまとめ	103
8.2	今後の課題	104
8.2.1	空間2次元の移流拡散問題	105
8.2.2	特性曲線の再構成	106
8.2.3	近似 NA 安定性の妥当性	106
	謝辞	107
	参考文献	109

第1章 序論

1.1 背景と目的

本研究は、大規模で疎な線型常微分方程式系のうち、その行列の係数が変数に依存するものを対象とし、その数値解法、およびその解法アルゴリズムの並列計算による高速化を目的とする。

微分方程式による数理モデル化と数値計算

物理学や化学等の自然科学に限らず、経済学等の社会科学を含む幅広い分野の現象の探求において、数学的な定式化や解析を用いる手法を数理科学と呼ぶ。これらの多くの場合、現象を微分方程式として記述し、数理モデル化する。しかし、こうした数理モデルの微分方程式は、概して、解の存在性・一意性は証明されても、その様子を関数式で表わすことができない場合がほとんどである。このような場合に数値解法は有効で、計算機の性能向上の著しい今日、ハードウェア的にもソフトウェア的にも、より高速かつ正確に解を求める手法が求められている。ここで述べたように、科学の問題を微分方程式で数理モデル化し、計算機によるシミュレーションによって現象を分析する一連の手法を計算数理科学と呼ぶ。

伝統的な微分方程式の数値解法の多くは、与えられる方程式が一般的な形であることを前提として解法の構成がなされてきた。つまり、解析的手法が困難なモデルについて、汎用性に重点をおいて、計算機によるシミュレーションを主たる研究手段とするものである。しかし、幅広い問題に適用可能な解法が、個々の特定の問題に対しては、必ずしも最良の方法でないことは十分ありえる。

現在の計算数理科学においては、現象の数理モデル化には従来の常微分方程式や偏微分方程式だけでなく、ハミルトン方程式や遅延微分方程式、確率微分方程式を用いることも多い。これらの方程式系についても、解析解の表現の困難さから、数学的な手法による解析よりも計算機を用いる数値解析の有効性は容易に想像できる。また、こうした特定の現象の中には、何らかの保存則を内在し保存性の再現が重要な意味をもつ現象がしばしばみられる。数値計算においてこれらの保存量の再現がどの程度の正確さで可能であるかを知ることは重要で、ゆえに、それぞれの系に特化して適用可能な専用解法ともいべき数値計算法が種々紹介されているが、共通の問題として、その数値解の安定性の達成が重要となる。つまり、数値計算における保存量の再現が、数値的安定性と密接な関係があると考えられる。

移流拡散方程式の数値解

そこで、本研究における問題の出発点として、移流拡散方程式を対象とし、それについて新たな数値解法を提案するとともに、その解法における問題点を検討する。

計算数理学のうち流体力学に関する現象である、質量・運動量・角運動量・エネルギー等の保存則は空間および時間を独立変数とする偏微分方程式系で記述できる。そしてこれらの系を、有限差分法・有限要素法・境界要素法等の計算手法を用いて離散化し、コーディングの結果をいろいろなコンピュータに投入して、現象をシミュレーションし、流れ場の理解、新しい法則の発見や設計に必要な情報を得る。

こうした数理モデルの応用上、しばしば登場する移流拡散系の数値解析においては、空間1階微分項として表現され、不安定要因となりうる移流項の離散化が重要である。非線型の移流拡散方程式に対して、通常微分項の基本的離散化法とともに、例えば、1階微分の離散化には、Lax-Wendroff法や風上差分法を用いる等の工夫が要求されるが、その場合は数値拡散と数値粘性の大きさが問題となり、移流項優位の方程式において局所的に拡散項が勝るような場合、しばしば不自然な振動現象が数値解に表れた。

そこで我々は、移流拡散方程式において拡散係数が十分小さいとき、本来放物型であるその系をほとんど双曲型とみなせることに着目し、その特性曲線に沿う方向に離散化することが自然と考えた。そして、この原理で空間離散化し、その上で時間発展型の離散化方程式を解くことを提案する。この離散化法は、移流方程式における特性曲線法、“線の方法 (method of line, MOL)” に倣った方法で、特性曲線の方向に濃度が保存される様子から、保存系モデル化の一つとみなすことができる。また、Y. TOURIGNY ら [43] は、移流拡散方程式の初期値・境界値問題に対して、特性曲線に沿って節点 (node point) を動かす有限要素法を提案し、その収束性を調べた。この有限要素法を念頭に置きながら、我々は線の方法を特性曲線に沿って適用する移動節点差分法を導く。つまり、空間の節点を特性曲線に沿って動かしていく差分法であり、また、その差分化された常微分方程式を数値的に解くものである。

変数係数線型常微分方程式の数値解

一般に、時間発展型の偏微分方程式の数値解を得るために、まず空間変数について差分法や有限要素法を用いて離散化すると、方程式は時間を独立変数とする大次元連立常微分方程式に帰着する。本論文の解法のように、移流拡散方程式を特性曲線に沿って差分化した場合も同様で、

$$\mathbf{y}'(t) = \mathbf{L}(t)\mathbf{y}(t) + \mathbf{F}(t)$$

の形の連立常微分方程式系 (ODEs) が得られる。

この時間発展型の方程式系を解くため、時間方向に適切な数値積分法を用いると、最終的には大規模線型方程式系を繰り返し解くことに帰結する。一般に、不安定要因を含む方程式系に適用すべき数値的に安定な解法では、1ステップあたり上の意味で計算量が増大する。モデルのシミュレーションのためには、これらの問題点を克服して、上記の常微分方程式系を高速に、かつ、正確に解くことが重要となる。

変数係数常微分方程式系の数値解法についての先行研究としては、準解析的な解法として、Magnus 展開法が W. MAGNUS[29] によって提案された。この研究の系列として、数値的な手法が A. ISERLES ら [25] によって示されている他に、I.B. BERNSTEIN ら [1] による数値解が急速に振動する場合のスキームの導出が挙げられる。しかし、いずれの解法もアルゴリズムが複雑であったり、要求される大規模系への適用が困難で、実用的とはいえない。

ゆえに本研究では、この変数係数行列をもつ線型の常微分方程式系に特化して数値的に解く、並列計算機向きの離散変数法を新たに構成する。これまでに提案された定数係数の線型常微分方程式系に適用可能な陰的 Runge-Kutta 法 [18, 27] に倣って、陰的 ROW 型の解法を、さらにそれを発展して、ブロック Rosenbrock 法を提案し、具体的なスキームを構成する。この新たな近似解法に対して収束性や安定性を解析し、その結果を数値実験により確認する。さらに、実際に並列計算機上で実行し、並列化の効果を検証する。

1.2 本研究の概要

移流拡散方程式の特性曲線差分法とは、まず、空間中の節点を等間隔の初期値から移流項に対応する特性曲線方程式に従って移動するように定める。そして拡散項については、時間刻みごとに得られる節点上の数値解を用いて、不等間隔の中心差分によって差分化することになる。以上の空間差分によって得られる時間方向の常微分方程式系は、特性曲線に沿う方向の数値解を求めるものとなる。つまり、与えられた偏微分方程式を特性曲線を解く部分と差分化方程式を解く部分の 2 つの常微分方程式系に分け、それぞれの部分で並列化を試みるものである。この方法について有効な並列化は、共有メモリ型アーキテクチャ上で動作し、離散変数法として陽的な解法を用いる手法である。しかし、安定性解析の結果、差分化方程式の安定条件が

$$\Delta t < \frac{\mathcal{M}h^2}{4\varepsilon}$$

であることがわかり、時間刻み幅が制限される陽的解法の限界から、安定性のため陰的で、かつ、並列化によって高速化が期待できる解法の構成が要求される。ただし、ここで解くことが要求されているのは、変数係数の線型常微分方程式系で、この系に特化した専用解法を開発する。

線型常微分方程式系に陰的 Runge-Kutta 法を適用するとき、段値 (stage value) についての非線型方程式系は、大連立線型系に置き換えられる。陰的 ROW 型解法は、この大連立線型系に対して線型変換による対角化を適用することで、解くべき方程式系を分割し、次元数の軽減と並列化によって高速化を図るものである。A 安定でかつ条件付き NA 安定である 2 段 3 次の公式を導出し、2CPU の分散メモリ型環境で実行可能な並列化コードを生成した。この数値実験によってその収束性、安定性および並列化の効果を確認できるが、方法の最高到達次数は 3 であることがわかり、より高精度が要求される場合に適当でない。ただし、並列化の観点では非常に有効な手法であるため、これを応用して、つまり、ROW 型の並列性をなるべく損なわずに、より高精度のスキームが得られる解法を考案する。

ブロック Rosenbrock 法は、陰的 ROW 型解法を半陰的に拡張したものである。大連立線型系は、 b 個の ROW 型に対応するブロックに分割され、ブロック行列としてみたとき上三角となるようにスキームを構成する。各ブロック内の計算は、ROW 型と同等で並列性が非常に高い。ただし、ブロックごとの計算は、下段から逐次実行しなければならない。これによって、並列性は損なわれるが、4 次以上の解法の構成が可能となる。 $(b, s_1, s_2, \dots, s_b)$ 型とは、ブロック行列の段数が b で、個々のブロック中の段数がそれぞれ順に s_i であることを意味する。そして、数値実験によって構成した $(2, 2, 2)$ 型公式の収束の次数が 4 であることと、ある程度の並列化の効果を確認できる。これによって、従来有効と考えられた 2 段 4 次の陰的 Runge-Kutta 法である、Butcher-Kuntzmann 公式と同等の精度と安定性をもち、並列化による高速化が期待できる解法を構成できた。

また、陰的 ROW 型法およびブロック Rosenbrock 法は、解くべき線型常微分方程式系が 3 重対角であるとき特に有効である。これは、解くべき系が 3 重対角であるとき、ROW 型（ブロック）において解くことになる線型系も 3 重対角となり、線型次数で計算可能な Thomas アルゴリズムが適用できるからである。しかし、線型方程式系の解法の計算量が軽減できることにより、相対的に分散メモリ環境でのプロセッサ間通信が無視できなくなる。この問題についての検討は今後の課題である。

本研究の主要な成果は、変数係数線型常微分方程式に特化適用可能な離散変数法であるブロック Rosenbrock 法の考察である。この型の離散変数法について、並列計算適用可能で、かつ、条件付き NA 安定性を満たす 4 次公式を導出し、数値実験によってそれら进行评估した。数値実験結果は、この解法による数値解が従来有効とされた 2 段 4 次 Butcher-Kuntzmann 公式と同等の収束性と安定性を示し、かつ、並列化による計算量の軽減も示した。さらに、この解法を移流拡散方程式の特性曲線差分法に適用する実験においても有効な収束性・安定性を示し、並列化の効果についても計算量の見積もりの範囲内で十分な結果であった。

1.3 本論文の構成

本論文の内容は以下の全 8 章からなる。まず、第 2 章では、利用する並列計算環境のアーキテクチャやプログラミング言語を紹介し、並列化手法を述べる。また、並列化の効果を評価する基準として速度向上度 (speedup ratio) を導入する。次に、第 3 章において、我々が対象とする移流拡散方程式が表わす現象および取り上げるモデル問題を挙げ、従来の方法による結果や最近の研究の成果を述べる。さらに、第 4 章から第 6 章までの収束性および安定性の解析の理解を助けるために、一般的な Runge-Kutta 法に対する解析手法を導入する。

第 4 章は本論文の問題提起となる部分で、移流項に対応する特性曲線を定義し、その曲線上の節点 (node point) における離散化を考える移流拡散方程式の特性曲線差分法を提案する。この方法の数値的収束性、安定性を解析し、さらに、古典的な離散変数法を用いて解法の有効性と問題点を考察する。この古典的な解法を用いる並列化は効果的な結果を示すが、構成

される節点の不等間隔性に依存する不安定性から，解くべき常微分方程式系に対して，十分な安定性をもつ離散変数法の適用が必要となり，ここで新たな並列化という課題が生じる．

したがって，第5章および第6章は，変数係数線型常微分方程式系を安定にかつ高速に解くための新たな数値積分法の構成に捧げられる．並列化向き陰的 ROW 型解法およびその拡張に位置するブロック Rosenbrock 法は，並列化による高速化や変数係数系に対する十分な安定性を示すので，移流拡散方程式の特性曲線差分法に適用可能な高速かつ安定な解法であるといえる．また，第4～6章においては，次の章への課題を明確にするために各解法の有効性を検証する予備実験を実施する．続く第7章では，本論文の方法を評価するために，移流拡散方程式系に対して特性曲線法を適用し，得られる変数係数線型系に対してブロック Rosenbrock 法を適用する並列化数値実験を実施し，その結果および考察を述べる．第8章では，本論文の結論および今後の課題を述べる．

第2章 並列計算環境と計算手法

近年 CPU の高速化は著しく、より一層の速度向上が期待されるが、電子工学的な見地からはその限界が予想される。そこで期待されるのがアルゴリズムやデータを分割し、複数個の CPU を用いて同時に計算をする並列計算である。単純には n 個のプロセッサを用いれば n 倍の計算速度が期待できるが、そのアーキテクチャや並列化手法を理解することは重要で、現実には理想的な計算速度は得られないことが知られている。本章では、我々が取り扱う並列計算環境や並列化手法を述べ、最後に並列化パフォーマンスの評価方法を導入する。

2.1 並列計算アーキテクチャ

1つの問題を複数のプロセッサを用いて計算する並列プログラムに対して、それを実際に計算する計算機構成を並列計算機と呼ぶ。並列計算機は、複数のプロセッサあるいは複数の独立した計算機を何らかの方法で結合する構成で、その種類は大まかにメモリの構成によって、メモリ共有型とメモリ分散型に分けられる ([3] p.3, [24] pp.3-8)。

前者のメモリ共有型は、複数のプロセッサが結合網を介して共通の主記憶領域を利用できる構成である (図 2.1)。ここでのメモリ空間は、複数からなる構成であっても仮想的に単一アドレス空間を用い、それぞれのプロセッサは各変数領域へのアクセスにそのアドレスを使う。この場合、複数の CPU が同時に同一空間にアクセスすることに拠るオーバーヘッドに注意が必要であるが、CPU 間で共通のデータを利用することが容易である。しかし、構成するのにコストがかかり、また、アーキテクチャが画一化されていないため、現在は主流ではない。しかし、並列計算の入門としてはこの型の並列化手法は重要で、今後何らかの技術革新によって普及するかもしれないので、現段階で安易には捨てがたい。代表的な構成として、Intel Xeon プロセッサを 2^n 個搭載した PC (パーソナルコンピュータ) が挙げられる。

これに対して、分散型アーキテクチャ (図 2.2) は 1CPU につき 1つの RAM が構成され、他の CPU に配置されたメモリ上のデータを利用するためにはプロセッサ間でのデータ通信が必要となる。プロセッサ間通信については種々のアーキテクチャが考えられるが、現在は安価に構成できる LAN を用いるものが主流である。このように LAN に接続する複数の PC によって構成される仮想的な並列計算機を PC クラスタと呼び、安価に構成可能な分散メモリ型の代表である。分散型の場合は、個々のプロセッサで独立にプロセスが実行されるが、相互にデータ交換をするために通信が必要で、ハードウェア的には通信速度の向上が、ソフトウェア的にはアルゴリズムにおいて通信の回数、量を減少させることがより高速な並列計算のための課題である。

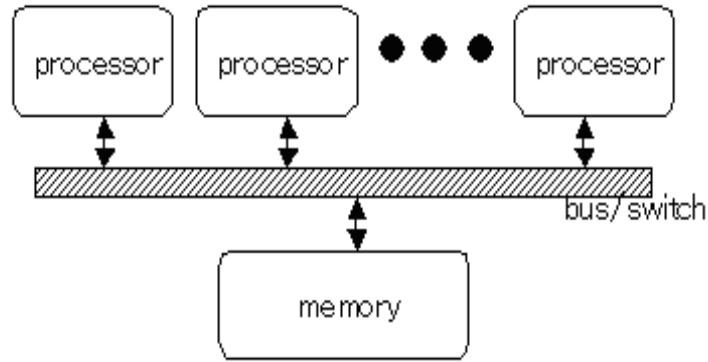


図 2.1: 共有メモリ型並列計算モデル

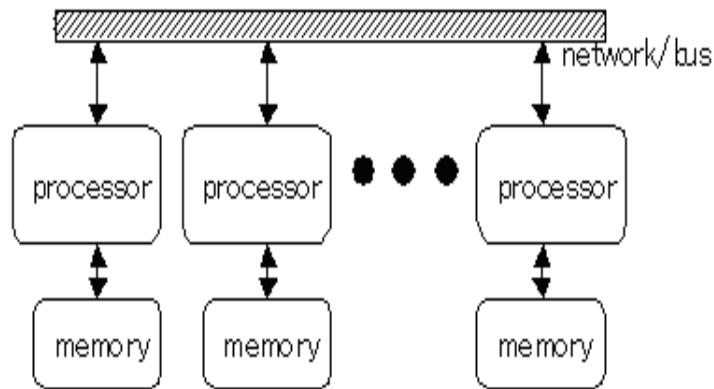


図 2.2: 分散メモリ型並列計算モデル

さらに共有メモリと分散メモリをあわせもつ構成 (図 2.3) や分散メモリを仮想的に共有する分散共有型 (図 2.4) もあり, 一般的には並列計算のコーディングはアーキテクチャ依存といわれるゆえんである.

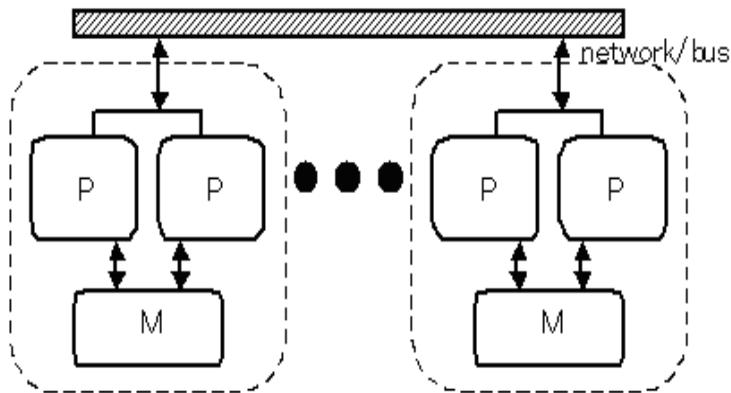


図 2.3: 共有メモリ型を分散する PC クラスタ

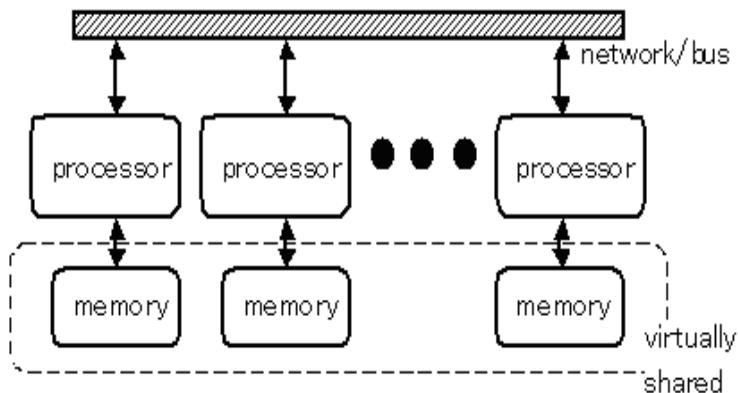


図 2.4: 分散共有型 (分散メモリを仮想的に共有する構成)

本研究では, 共有メモリ型として Silicon Graphics 製 PowerChallenge,4CPU を用い, 分散メモリ型として VisualTechnology 製 VT-Alpha6(2CPU × 2) の 4CPU を用いた. VT-Alpha は, クラスタのノードが 2CPU をもち, その 2 つのプロセッサが共有メモリをもつ共有分散型であるが, 各プロセッサが利用するメモリ領域は共有せずに動作させた.

2.2 プロセスの流れとデータ分割

逐次計算機上では、プログラムから1個の命令ストリームが生成される。M.J. FLYNN はコンピュータを分類して、この単一プロセッサコンピュータを SISD(single instruction stream, single data stream) 型と呼んだ ([3] p.4, [24] pp.8-9)。これに対して、汎用的な並列計算機上のプログラムは、MIMD(multiple instruction stream, multiple data stream) 型であり、各 CPU が別のプログラムをもち、その各プログラムに対して1つずつの命令ストリームが生成されて、各命令は異なるデータに対して作用する。

共有メモリ型においては、マスタプロセッサ上で主となるプログラムが実行され、必要に応じてプロセスが生成 (fork) され、スレーブプロセッサ上で実行される。並列部分が終了すると同期 (join) して、プロセスも終了する。単純にループを分割する場合は、同一の命令ストリームが複数のデータ処理をするという意味で、SIMD(single instruction stream, multiple data stream) 型であるといえる。

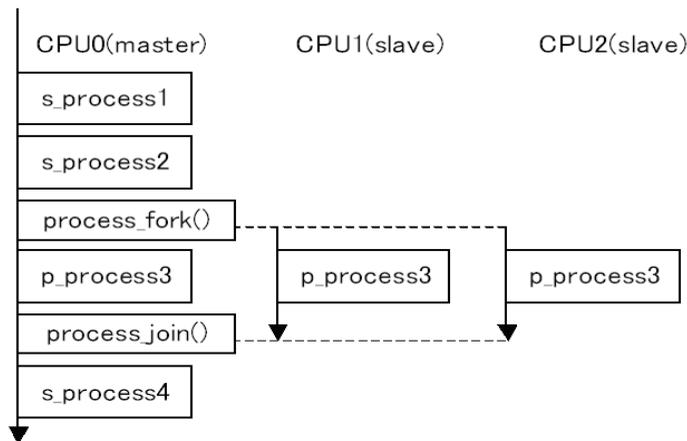


図 2.5: 共有メモリ型のプロセス生成と同期

分散メモリ型においては、プログラムが実行されるとすべてのプロセッサ上でプロセスが生成され、プログラムが終了するまでプロセスは存続する。各 CPU のプロセスは異なる処理を行い (MIMD)、同期はしなくてもよいが、プロセッサ間通信時には正しい順序でデータを送受信する必要があり、オーバーヘッドの小さいプログラムを実現するためには、プロセス同士のバランスを考慮する必要がある。

並列計算において、計算領域を各プロセッサにどのように割り当てるかも重要である。ブロック分割とは、計算領域である配列の要素をあるまとまった数ごとに分割する手法である (図 2.7)。通常は、ブロック数がプロセッサ数になるように均等に分割する。これに対して、隣りの要素を隣のプロセッサに順に割り当てていく分割をサイクリック分割と呼ぶ (図 2.8)。この中間に当たるのがブロックサイクリック分割である。計算する全要素数を N 、ブロックサイズを B 、プロセッサ数を P として、全体を $\frac{N}{B}$ 個の帯に分割する場合、 k 番目の帯を k

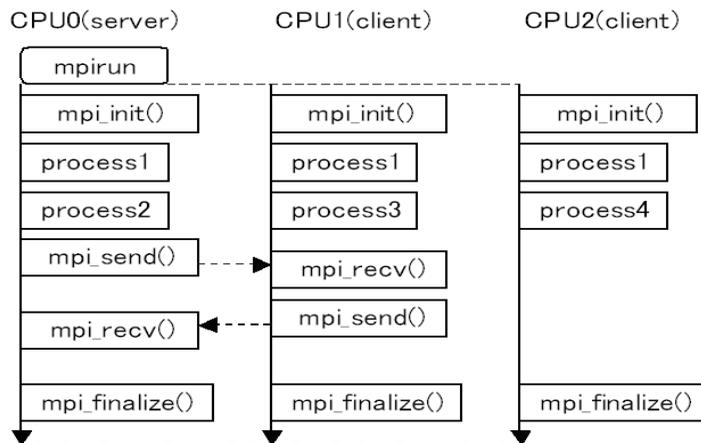


図 2.6: 分散メモリ型の並列実行とメッセージ通信

mod P 番目のプロセッサに割り付けることになる。 $\frac{N}{B} = P$ の時はブロック分割， $B = 1$ の時にはサイクリック分割となる。図 2.7, 2.8 のように要素数がプロセッサ数で割り切れない場合は，ある規則を設けて（例えば，番号の小さいプロセッサから順に）余りのプロセスを分配する。

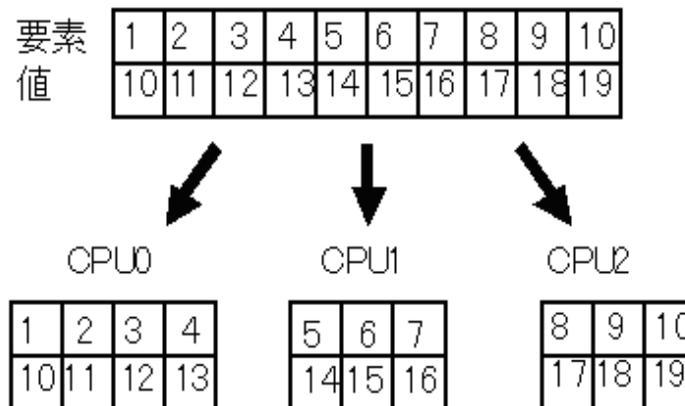


図 2.7: ブロック分割

各計算領域（配列の各要素）ごとのプロセスの負荷がほぼ均等な場合は，ブロック分割が望ましい。また，分散メモリ型でプロセッサ間通信が必要な場合を考慮するときも通信に要するコストを必要最小限に抑えるためには，この分割が適当である。これに対して，各領域ごとの負荷が不均等となるときは，サイクリック分割が適当である。単純なサイクリック分割によってプロセッサ間通信に依存するオーバーヘッドが著しく増加する場合は，適当なブロック・サイクリック分割を用いることでパフォーマンスを高められる。

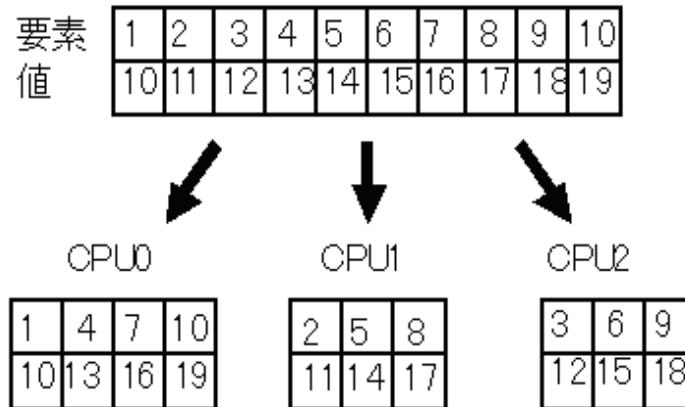


図 2.8: サイクリック分割

2.3 並列化言語

並列計算を実行するプログラムは、逐次プログラムに並列化コードを挿入することで実現する。近年は、自動並列化という手法もあり、単純な反復構造の並列化は容易に適用可能であるが、特定のアルゴリズムの並列化のためには、ある程度の並列化手法を身につける必要がある。また、この並列化手法もアーキテクチャ依存であり、ここでは、共有メモリ型、分散メモリ型の一般的な並列プログラミング技法について、“コーディング”、“オーバーヘッド”、“プロセスの粒度”に焦点を当てて、実際に並列化する際の問題点を明確にする。

共有メモリ型において SIMD 型プログラムを実現するためには、並列計算するループ構造に対して並列化を指示する。この場合、利用する変数が、全 CPU から参照可能な共有メモリ上か各 CPU のみが利用する局所メモリのいずれに割り当てられているかを考慮するだけで、アルゴリズムの並列化にそれほどの苦労は要しない。データ分割についても、通常の場合、ブロック分割とサイクリック分割のどちらでも大差はないが、同一メモリ領域へ同時にアクセスすることに起因するオーバーヘッドに気をつける必要がある。しかし、プロセスの対称性が崩れるような複雑な並列化はコーディングの面でも高速化の面でも適当でない。本研究で利用した PowerChallenge では、並列用言語である MPC(C for multi-processor) を採用したが、最近ではフリーウェアの OpenMP を用いるのが通常である。Java や OpenMP 等の新しい開発系においては、OS が管理する 1 つのプロセス中で処理を並列分配するときには、分けられた流れをプロセスとは区別してスレッドと呼ぶが、本論文では、特に区別を意識する必要がある場合を除いて、分けられた流れをすべてプロセスと称する。

分散メモリ型の場合、1 つのプログラム中で 1 度並列プロセスを生成した後は、プログラムの終了時まで同期しなくてよいが、プロセッサ間の情報交換のために必要に応じてメッセージ通信を行う。この通信時のオーバーヘッドがボトルネックとなるため、分散メモリ型のアルゴリズムを考案するときは、通信回数および通信量をより少なくする工夫が必要である。し

たがって、データ分割は、データのプロセッサごとの分配を考えるとブロック分割が適当で、現在主流である分散メモリ型においては、MPI(Message Passing Interface)を用いて並列計算およびメッセージ通信をするのが通常であり、標準化しているといえる。MPIを用いれば、プログラム中でのif文などの条件文によって、プロセスID毎に異なる処理を指示できるため、より汎用的な並列計算が可能である。もちろん、単純なループ構造に対する並列化も可能である。

以上で述べたように並列化言語の仕様もアーキテクチャに依存するが、並列化するプロセスについての粒度の観点から見るとその立場がより明確になる。ここで、粒度とはプロセスの大きさを意味する。粗粒度では、各プロセスは多数の逐次命令を含み実行にそれ相当の時間がかかり、プロセス毎に異なる処理を多く含む場合、実行時間はプロセス毎に不均等となる可能性が高い。これに対して、細粒度では、各プロセス自体が小さいこととその実行がほぼ均一であるためプロセス毎の実行時間の差が小さいといえる。

粒度を踏まえた上で、今一度、それぞれのオーバーヘッドである同期と通信に焦点を当てる。粗粒度においては、共有メモリ型の場合、同期にかかるオーバーヘッドが問題となり、分散メモリ型の場合、プロセッサ間の通信を減らす工夫でオーバーヘッドを軽減できる。細粒度においては、共有メモリ型の場合、同期にかかるオーバーヘッドがなくなり高速な実行が期待でき、分散メモリ型の場合、プロセッサ間の通信が無視できなくなり有効でない。

以上をまとめると、並列化するプログラムの実装は、OpenMPによる並列化は単純な構造について容易に並列化指示が可能で、さらに、データの分割についても自由度がある反面、複雑な並列化指示が困難なことから、プロセスの同期が必要なことから、各プロセスの負荷にバラつきが大きい粒度の粗いものに適していない。これに対して、MPIによる並列プログラミングは、並列化指示に汎用的な記述が可能で、また、各プロセスは独立に実行される反面、プロセス間のデータのやり取りの通信に多く時間を要したり、プログラムの記述が複雑となりデバッグが困難となるのが欠点であるが、各プロセスの負荷にバラつきが大きい粒度の粗い場合でもプロセスの同期の必要はなく、必要となるプロセッサ間通信の回数やデータ量を少なくするアルゴリズムやデータ構造の工夫が有効である。

以上のことを簡潔に表 2.1 にまとめ、並列計算導入時の採用要件とする。

表 2.1: MPI と OpenMP の比較

	MPI	OpenMP
コーディング時の利点	複雑な処理を自由に記述可能	簡単に並列化可能
オーバーヘッド	プロセッサ間通信回数・量	プロセスの生成・同期
プロセスの粒度	粗粒度向き	細粒度向き

分散メモリ型環境を仮想的に分散共有型とみて、共有メモリ型と同等に並列化指示行を挿入する言語仕様となっているのが富士通製の HPC&F(high performance C and Fortran)である。富士通製 VPP5000 上での開発環境が進みつつあるが、並列化性能が MPI に劣るため、

現段階では普及していない．HPCと同じ考え方で開発されたのが Omni OpenMP である．これは，共有メモリ型で動作するプログラムがほぼそのまま PC クラスタ上，つまり，分散メモリ環境で実行可能である．分散共有メモリ (SCASH) をソフトウェアで実現するため，共有メモリへのアクセスのオーバーヘッドは致命的に大きく，多くのデータを複数のプロセス (OpenMP ではスレッド) で頻繁に更新しあうようなプログラムの場合には十分な性能が得られない．しかし，実際には分散メモリ型であることを意識していさえすれば，容易に並列化可能であるという利点のためこれらのコーディングの将来はそれほど暗くない．

2.4 微分方程式の数値解法に適用する並列化技法

アルゴリズムの並列化技法は，いくつかのレベルにおいて実現することが可能である．単純に配列に対する反復処理をブロック分割することから，逐次アルゴリズムを並列計算向きに記述しなおすことまで様々である．微分方程式系の数値解法の並列化については，大きく分けて3つの手法が考えられ ([3] pp.166-170, [17])，それらは図 2.9 のように “空間”，“時間”，“方法” についてである．

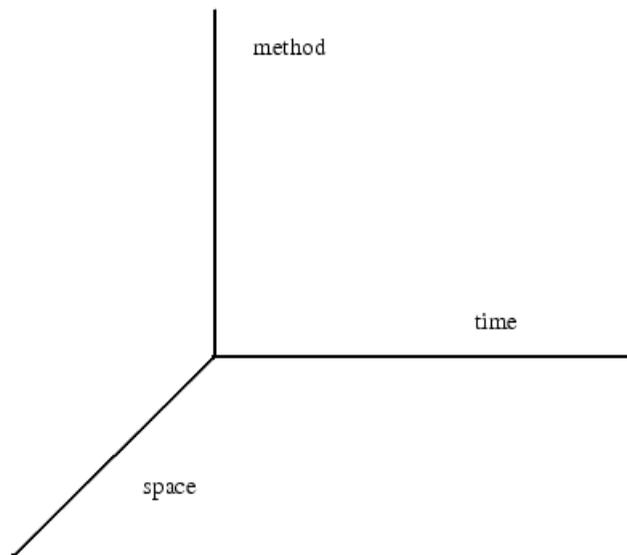


図 2.9: 並列化空間

第一の並列化は各方程式において空間変数の相互依存性が小さい場合，領域を分割し分割した領域ごとの計算を並列プロセッサに割り当てる．これは，空間方向についての並列化とみることができ，C.W. Gear は parallelism across the system と呼んだ．

第二には適用する数値積分法に並列化向きのものを利用することである．各ステップでの値や段値 (stage value) を並列に計算することになるので，時間方向についての並列化とみて，parallelism across the steps と呼ぶ．

第二レベルで述べた並列化向きの“方法”は，parallelism across the method に分類してもよいが，我々は，ほとんどの微分方程式系の数値解法において，各段 (step, stage) において，最終的に大規模疎行列の線型方程式の数値計算に帰結することから，この第三の並列化としては，線型系の解法の高速度化を挙げたい．線型方程式の数値解法の高速度化において，Gauss 消去法の交代代入部分のパイプライン処理や Jacobi 法や Gauss-Seitel 法のような並列化が容易な古典的反復解法の並列化も考えられるが，並列化向きではないが積型反復解法 [14] による高速度化も期待される．

本研究で述べる並列化手法は，第一および第二レベルのものである．第4章の移流拡散方程式の特性曲線差分法は，各方程式の相互依存性が小さいので，空間領域を分割する並列化が可能で，特に共有メモリ型計算機上での並列化が有効である．第5,6章で述べる ROW 型解法やブロック Rosenbrock 法は並列計算向き数値積分法で，段値の計算の逐次性を減らすことで並列性を高めている．やや複雑なアルゴリズムで実現するため，プログラミングに汎用性をもつ分散型計算機上での並列化を採用する．ただし，ROW 型解法やブロック Rosenbrock 法において最終的に大規模疎行列の線型方程式を解くことになるが，安定な数値解を得るために解法を通常の Gauss 消去法に限定し，これについての並列化は本論文では取り扱わない．

2.5 並列化効果の指標

並列計算の効果を評価する指標として，本研究では“速度向上度”を採用する．速度向上度は，スピードアップ率 (speedup ratio) とも呼ばれ，

$$\text{速度向上度 (倍)} = \frac{1\text{CPU での実行時間}}{n\text{CPU での実行時間}} \quad (2.1)$$

によって見積もられる．つまり， n CPU での実行時間が 1 CPU のときの $\frac{1}{n}$ しか要しないならば，速度向上度は n 倍となる．後で述べるが，実際には，理想的な n 倍の速度向上は得られないので，通常 $0.8n$ 倍以上ならば，並列化の効果は高いといえる．この並列化の効果に焦点を当てた指標が“並列化効率”で，

$$\text{並列化効率 (\%)} = \text{速度向上度} \times n \times 100 \quad (2.2)$$

と表わされる．この指標は，並列計算時にどれだけのプロセッサが有効に利用されているかを意味する．並列化効率が $\frac{100}{n}\%$ を下回るとき，速度向上度も 1 倍を下回る．

理想的な並列計算を妨げる主な要因としては，アルゴリズムの逐次部分と CPU 間通信やプロセス同期のオーバーヘッドが挙げられる．ここでは，アルゴリズム全体において，逐次的に計算しなければならない部分によって並列化の効果が損なわれることを考える．図 2.10 に示されるように，逐次計算機上での全体の計算時間を t_s ，逐次計算部分の割合を f とするとき，並列計算機上での計算時間 t_p は，オーバーヘッドを考慮しない場合で， $t_p = ft_s + (1-f)t_s/n$ で表わされる．これを速度向上度で表すと以下ようになる．

$$\frac{t_s}{ft_s + (1-f)t_s/n} = \frac{n}{1 + (n-1)f} \quad (2.3)$$

これは Amdahl の法則 ([3] p.24, [24] p.29) と呼ばれ、例えばアルゴリズム全体において並列化可能部分が 95% であるような場合でも、理想的にプロセッサ数が無限個と仮定する最大速度向上度は 20 倍に留まってしまう。この値には、通信や同期のオーバーヘッドが含まれないので、実際の並列化による速度向上度はさらに劣るものとなる。

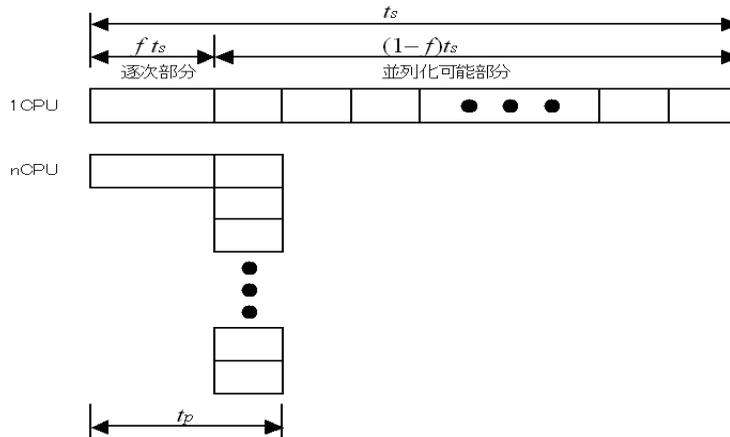


図 2.10: 逐次問題の並列化—Amdahl の法則

並列化の効果を評価するのに同じアルゴリズムの 1CPU での実行時間と n CPU での実行時間の速度向上度を用いるのは、基礎的なレベルでのアルゴリズムの性能の見積もりには有効であるが、時によっては適当でないことがある。K. BURRAGE も述べるように、新しく提案される並列化アルゴリズムは、従来の逐次計算アルゴリズムのうちもっとも効果的なものと比較して評価されるべきである ([3] p.27)。つまり、実用的な並列化の評価のためには、同一アルゴリズムの逐次版とではなく、逐次計算機上で最も効果的なアルゴリズムとの比較が必要である。

2.6 並列計算量の評価

並列計算の計算量を見積もることを考える。実際の並列計算では、図 2.10 のプロセスの分配による速度向上に加えて、プロセッサ間通信やプロセス同期に依存するオーバーヘッドを考慮する必要がある。ただしここでは、本論文でよく用いる分散メモリ型である PC クラスタで、全プロセッサが同じ種類かつ同じ速度のシステムに限って考える。

分散メモリ型では、問題の全実行時間中にメッセージ送受信時間が含まれるので、並列実行時間 t_p は、計算部分 t_{comp} と通信部分 t_{comm} から構成される。

$$t_p = t_{\text{comp}} + t_{\text{comm}} \quad (2.4)$$

計算部分の時間は、逐次アルゴリズムの場合と同様に推定できる。複数のプロセスが同時に実行されているときを除いて、最も複雑なプロセスの計算量が分かればよい。

通信時間はメッセージ長，結合構造，および転送モードに依存するが，1次近似として次の式を用いる．

$$t_{\text{comm}} = t_{\text{startup}} + Nt_{\text{data}} \quad (2.5)$$

ここで， t_{startup} は起動時間で実質的にはデータのないメッセージを送信する時間である． t_{data} は1データ語の転送時間で， t_{comm} においては，通信量 N に依存する量に現われる．

また，複数のプロセッサ間で通信が必要な場合，例えば，サーバから同一のデータをクライアントにブロードキャストする場合でも，マルチキャストに特化したアーキテクチャでない限り，原則的にはそれぞれの通信について1対1通信と同等の時間を要することにも注意が必要である．

これらのことは，並列アルゴリズムの構成において，非常に難しい問題点である．プロセッサ数の増大にともなって，1回の通信における通信量は減少するが，必然的にプロセッサ間の通信回数が増大する．また，PC クラスタの場合，概して， t_{startup} ， t_{data} とともに実行時に無視できないことが多い．

ここで，通信のオーバーヘッドを考慮する場合の計算量を評価する．条件は次のとおりである．計算のサイズを次元数 d とし，領域は使用するプロセッサ数 n 個で分割されるとする．データの分配や集約が複数回必要で，プロセッサ間通信のうち，次元数に依存して $\frac{d}{n}$ 個のデータを送受信する回数を c_1 ，次元数に依存しないデータを送受信する回数を c_2 とする．そして，1CPU 時の逐次部分実行時間を $t_{sp} = ft_s$ ，並列部分実行時間 $t_{pp} = (1 - f)t_s$ とし，これらの計算量はあとで考察する．

逐次計算時間 t_s および通信を考慮した並列計算時間 t_p の見積もりは以下のとおりである．

$$\begin{aligned} t_s &= t_{sp} + t_{pp} \\ t_p &= t_{sp} + \frac{t_{pp}}{n} + c_1 n (t_{\text{startup}} + \frac{d}{n} t_{\text{data}}) + c_2 n (t_{\text{startup}} + \frac{0}{n} t_{\text{data}}) \\ &= t_{sp} + \frac{t_{pp}}{n} + (c_1 + c_2) n t_{\text{startup}} + c_1 d t_{\text{data}} \end{aligned} \quad (2.6)$$

まず，通信に関する項を除けば， t_p は前節で用いた評価式と同値で Amdahl の法則 (2.3) も成り立つ．また，この評価式において， t_{pp} が他より大きい計算量である場合，有限個の n プロセッサによる並列化であっても，並列化の効果は大きい．しかし， t_{pp} が t_{sp} や t_{startup} と同等のオーダである場合，並列化の効果はあまり期待できない．さらに，PC クラスタ等の場合 c_1 や c_2 が大きくなると， t_{startup} の項が無視できなくなり，計算速度がむしろ並列化したほうが遅くなってしまいう可能性がある．

以上より，アルゴリズムの並列化を試みる際には，まず，十分なプロセッサ数でオーバーヘッドを考慮しない理想状態において，逐次実行率 f を減らすことを考慮する．その上で，実用のためには，通信に要する計算量の軽減，PC クラスタにおいては特に通信回数を減らすことを考慮する必要がある．

第3章 移流拡散方程式とその数値解

流体運動の支配方程式は Navier-Stokes 方程式であるが、双曲型、放物型、楕円型の混合偏微分方程式であり、その非線型性のため解析が困難である。しかし、Navier-Stokes 方程式の基礎方程式群は移流拡散方程式であり、自然現象の多くは移流拡散方程式によって支配されているため、議論を移流拡散方程式に限定することは、運動方程式の有効な数値計算手法を検討することを容易にする。したがって、本章では 3.1 節で移流拡散方程式が表わす現象についての安定性等の考察を述べ、3.2 節では本論文の数値実験で取り扱うモデル方程式を、3.3 節では基本的な解法による結果とその他の方法の概要を示す。さらに、3.4 節では、第 4 章、第 5 章および第 6 章での解析を助けるために一般的な Runge-Kutta 法の数値的収束性・安定性の解析を導入し、さらには変数係数線型常微分方程式の安定性解析のため、新しい NA 安定性を定義する。そして、3.5 節においては、変数係数系に適用する数値積分法の先行研究を評価することからも本研究の意義を示す。

3.1 移流拡散方程式が表わす現象

まず、我々が対象としている 1 次元移流拡散方程式を示す。時刻 t 、位置 x における未知変数 u が流体中の物質の濃度等を支配する偏微分方程式の初期値境界値問題で以下のように表わされる。

$$a(x, t)u_t + b(x, t)u_x - \varepsilon u_{xx} = f(x, t), \quad (x, t) \in Q \quad (3.1)$$

ここで、境界条件および初期条件は、

$$u(0, t) = u(1, t) = 0, \quad t \in [0, T], \quad (3.2)$$

$$u(x, 0) = u_0(x), \quad x \in \Omega \quad (3.3)$$

と与えられる。ここで区間 $\Omega = (0, 1)$ 、領域 $Q = \Omega \times (0, T)$ であり、 a, b, f, u_0 は既知の関数である。また、拡散係数 ε は正の定数とする。

この方程式系は、流体の運動方程式である 1 次元 Navier-Stokes 方程式、例えば、

$$u_t + uu_x - \nu u_{xx} = -\frac{1}{\rho} p_x$$

に基づいている ([42] p.11)。しかし、移流項の非線型性および圧力項のため数値解析を行うのはそれほど容易ではないため、この式における圧力項を非斉次項とし、移流項を線型化したものが我々の方程式である。一般に、この系の数値解析においては、不安定現象を示す移

流項の取り扱いが重要であり、移流項優位であるなしにかかわらず、その離散化時の安定性に注意が必要である。それに対して拡散項はスキームを安定化させることが知られている。ゆえに、従来の数値解法は移流項の空間離散化時に風上差分を用いる等の工夫がなされるが、今度はそこで生じる数値粘性が新たな誤差要因となり、この誤差を高次の風上差分を適用することで軽減を試みることになるが、スキームが複雑になることに対して、精度の向上や安定性がそれほど期待されるものでなかった。

方程式系 (3.1) は、 ε の値が相対的にその他の値よりも小さいとき、ほとんど双曲型となり、その双曲部分に対応する特性曲線に沿って解を得るのが自然であると考えられる。したがって、空間の離散化時の節点をたとえ不等間隔であっても一定の位置に固定するよりも、特性曲線に沿う方向に変化させて構成するのが有効と考えられる。次節のモデル問題を含め以下では、 $|\varepsilon| \ll 1$ であることを前提とし、移流拡散方程式の特性曲線とは、拡散項が十分小さいとみたときの双曲部に対応するものであることとする。

3.2 モデル問題

移流拡散方程式は、特別な場合を除いて解析解を得るのは困難である。また、よく知られるモデル問題 ([38] p.21, [42] pp.253-254) も誤差函数を含む解で本研究のテスト問題としては適当でない。したがって、我々は移流拡散問題 (3.1)–(3.3) について、解析解を決定したのちに各偏微分の項を求めることで、 a, b, f, u_0 を以下のように定めるモデル問題を構成した。

$$\begin{aligned} f(x, t) &= d[-qx(1-x)\sin(qt) + b_0(1-2x)\sin(8\pi x)\cos(qt) + 2\varepsilon\cos(qt)], \\ a(x, t) &= 1.0, \quad b(x, t) = b_0\sin(8\pi x), \quad u_0(x) = dx(1-x), \\ d &= 100, \quad q = 0.24\pi, \quad b_0 = 0.05 \end{aligned} \quad (3.4)$$

このときの解析解は以下のとおりである。

$$u(x, t) = dx(1-x)\cos(qt) \quad (3.5)$$

また、拡散係数 ε は数値実験において適当に定めることとし、特に断らない場合、 $\varepsilon = 10^{-3}$ とする。

解析解の3次元表示を図3.1に、空間内の流速函数 $b(x, t)$ の様子を図3.2に示す。式(3.5)より、 $\frac{25}{12} < t < \frac{75}{12}$ のとき $u(x, t) < 0$ となることが分かり、物理現象にそぐわないので数値実験の対象区間は $0 \leq t \leq \frac{25}{12}$ とする。図3.2において、 $b(x, t)$ の値が正のとき物質は空間の正の方向に流れ、負のとき物質は空間の負の方向に流れる。したがって、 $x = \frac{2n-1}{16}$, ($n = 1, 2, \dots, 8$) の近傍で流速が大きく、 $x = \frac{n}{8}$, ($n = 0, 1, 2, \dots, 8$) の近傍で流速が小さい。また、流速の向きから $x = \frac{2n-1}{8}$, ($n = 1, 2, \dots, 4$) の近傍に流束が集中し、 $x = \frac{n}{4}$, ($n = 0, 1, 2, \dots, 4$) の近傍で疎となることがわかる。

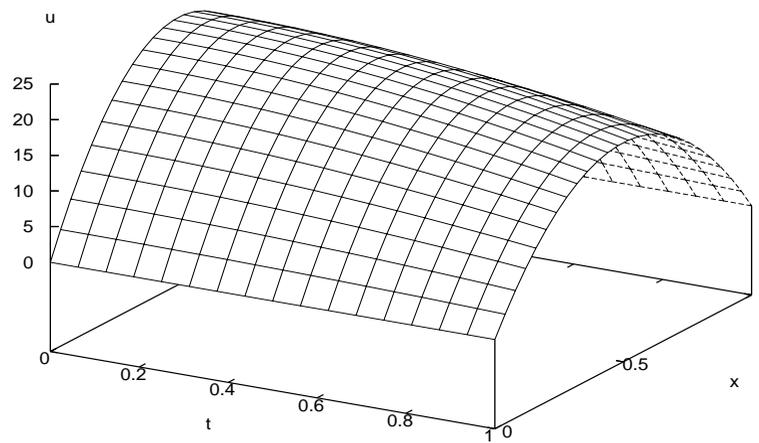
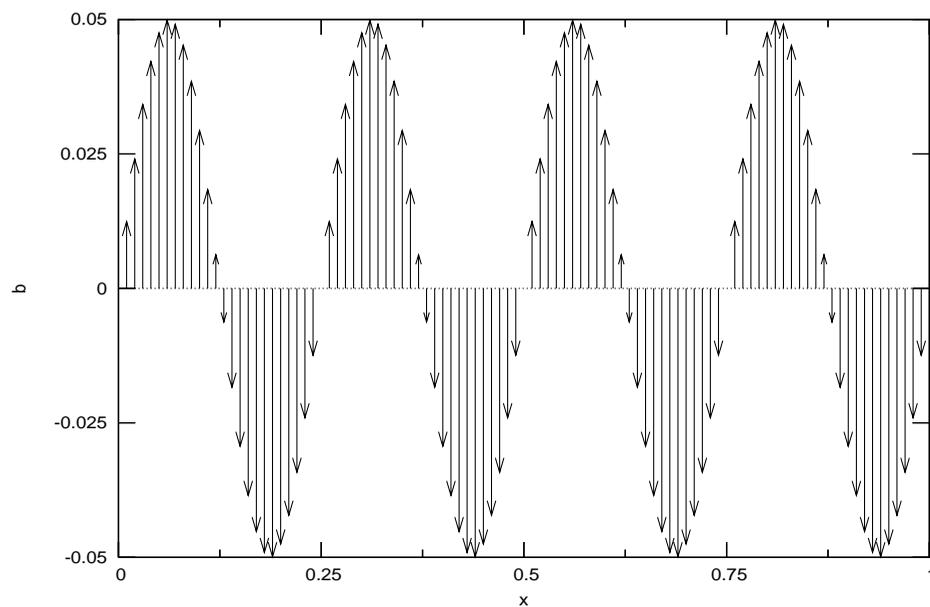


図 3.1: 解析解のプロフィール

図 3.2: 流速函数 $b(x,t)$ のプロフィール

この問題は、次の3.3節においては従来の方法、第4章においては新たに提案する新しい解法のテスト実験に用いる。また、第4章の実験では収束性、安定性の評価のためにも利用し、さらに第7章における数値実験にも用いる。

3.3 従来の結果と最近の研究

例として、上記のモデル方程式をよく知られる FTCS([15] p.15, [42] p.74) の有限差分法 (FDM) で数値実験を行った。FTCSとは、forward-time centered-space の略で、時間方向に前進差分、空間方向に中心差分を用いることである。

この離散化式に対して von Neumann の安定性解析の手法 ([15] p.18, [42] pp.262-264) を採用すると、移流項の安定条件は $c = b(x, t) \frac{\Delta t}{\Delta x}$ を Courant 数とするとき $|c| < 1$ であり、同様に、拡散項の安定条件は $d = \varepsilon \frac{\Delta t}{\Delta x^2}$ を拡散数とするとき $d < \frac{1}{2}$ である。ゆえに対象としている非定常移流拡散方程式の安定条件は $c^2 < 2d < 1$ のように制限される。

モデル問題 (3.4) において $\max |b(x, t)| = 0.05$ であるから、移流項についての安定条件は $\Delta t < 20\Delta x$ となり、実用的には $\Delta t = \Delta x$ 程度の数値実験が妥当で今回のモデルにおいて流れのみに依存する不安定要因は見出すことができない。次に、拡散係数を $\varepsilon = 10^{-3}$ とするとき、 $\Delta t < 500\Delta x^2$ であり、この条件がモデル問題の主要な不安定要因である。これについては、 Δx を $1/2$ にするとき、 Δt を $1/4$ にする割合を保てば、通常の刻み幅について安定である。さらに、 $c^2 < 2d$ からは、 $\Delta t < 0.8$ が得られる。以上より、通常の時間刻み幅として $\Delta t = 1/16$ を仮定するとき、 $(\Delta t, \Delta x) = (1/16, 1/25), (1/64, 1/50), (1/256, 1/100), (1/1024, 1/200)$ の組について安定な数値解を得ることが期待できる。ただし、解析上の安定条件は、 $\Delta t = 1/16$ のとき $\Delta x > 1/90$ 、 $\Delta t = 1/64$ のとき $\Delta x > 1/179$ 、 $\Delta t = 1/256$ のとき $\Delta x > 1/358$ 、 $\Delta t = 1/1024$ のとき $\Delta x > 1/716$ であり、時間方向に比べて空間方向の分割を小さくしすぎると不安定となる。図 3.3 に FTCS の数値解のプロフィールを示す。

本節で基本とする FTCS 法の数値実験結果を表 3.4 の FT-CS の項に示す。先に挙げた安定条件を満たす $\Delta t, \Delta x$ の組について、数値解が安定であり、 $\frac{\Delta x}{\Delta t^2} = \text{const.}$ のもとで、最大誤差が $O(\Delta t)$ にしたがって収束していることが確認できる。また、安定条件を満たさない場合については、最大誤差が急激に増大している。

FTCS 法は、解の単調性を維持できないため不連続面近傍で不安定であり、これが空間方向の収束性を低下させている。これに対して 1 次風上差分法 (upwind FDM) は、移流項の差分化をより自然に行うために、移流の絶対値を考慮した片側 (前進または後退) 差分を適用する解法であり、乱流を含めて移流項優位の問題に有効とされる。ただし、1 次風上差分は拡散が over-damping されるため不自然な数値粘性が現れる。時間方向に前進差分を用いる風上差分の数値実験は、表 3.4 の FT-UW に示す。収束性は、 $O(\Delta t)$ と見積もることができ、空間方向の分割による誤差の減少傾向も見られるが、不自然な数値粘性のために、 $\Delta t = 1/256, 1/1024$ の実験において期待通りの収束性が得られていない。

上記の 2 解法は、時間差分について前進差分を適用したが、ここで後退差分を組み合わせ

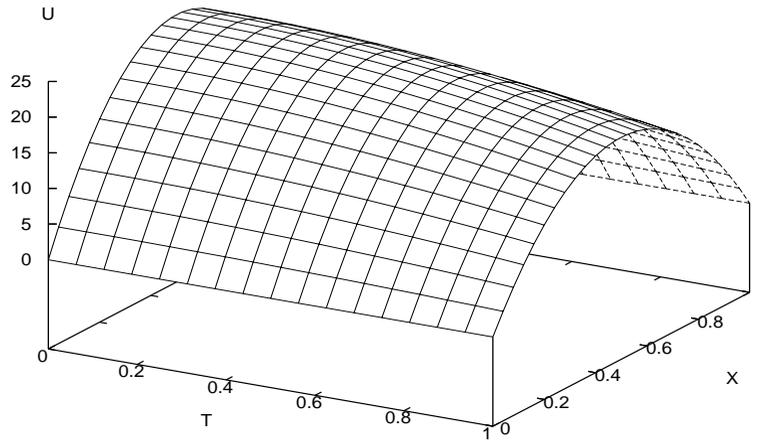


図 3.3: FDM による数値解のプロフィール

表 3.1: 各種解法の $t = 1.0$ における最大絶対誤差

解法	$\Delta t \setminus \Delta x$	1/25	1/50	1/100	1/200
FT-CS	1/16	0.4033E+00	0.4036E+00	0.4036E+00	0.4519E+07
	1/64	0.1006E+00	0.1007E+00	0.1007E+00	0.2711E+02
	1/256	0.2513E-01	0.2515E-01	0.2515E-01	0.2515E-01
	1/1024	0.6282E-02	0.6286E-02	0.6286E-02	0.6286E-02
FT-UW	1/16	0.3002E-00	0.3699E+00	0.6005E+00	0.3358E+10
	1/64	0.9062E-01	0.6730E-01	0.8255E-01	0.2346E+09
	1/256	0.1100E+00	0.5486E-01	0.2583E-01	0.1598E-01
	1/1024	0.1257E+00	0.6083E-01	0.2996E-01	0.1461E-01
CT-CS	1/16	0.1251E-02	0.1252E-02	————	————
	1/64	0.7817E-04	0.7823E-04	0.7823E-04	————
	1/256	0.4885E-05	0.4890E-05	0.4890E-05	0.4890E-05
	1/1024	0.3053E-06	0.3056E-06	0.3056E-06	0.3056E-06
SGA	1/16	0.1484E+01	0.7748E+00	0.5352E+00	0.4224E+00
	1/64	0.1542E+01	0.5756E+00	0.3166E+00	0.1947E+00
	1/256	0.1557E+01	0.5884E+00	0.2623E+00	0.1394E+00
	1/1024	0.1561E+01	0.5917E+00	0.2490E+00	0.1256E+00

ると、数値解の安定性・収束性を向上させることができる。Crank-Nicolson 法は、現在の時刻と次の時刻の数値解の中心差分ともいえる時間差分である。ただし、陰的公式であるため 1 ステップでの計算量が增大する。空間方向は中心差分、時間方向に Crank-Nicolson 法を適用した解法の結果を表 3.4 の CT-CS の項に示す。この解法は、時間・空間共に 2 次収束であるため、 $\frac{\Delta t}{\Delta x} = \text{const.}$ のとき、 $O(\Delta t^2)$ で収束し、数値実験結果も妥当な結果を示している。ただし、実験項目のうち、陰的スキームが正しく解けなかったものについては —— と示し、このスキームを用いても何らかの不安定要因があることがわかる。

以上のように FDM の応用としては、時間差分に後退差分を組み合わせることや高階差分によって人工的な粘性を加えること等の工夫が考えられるが、いずれも古典的な解法であり十分研究がなされていて、適材適所では実用化されているものの、数値解の精度や安定性については限界があるといえる。

また、構造格子による数値計算法である有限差分法の代わりに、非構造格子の有限要素法 (FEM) を用いることも考えられる。非構造格子を用いる解法は、概して、節点平均ではなく要素平均を用いるため、保存系に対して有効であり、複雑な構造に対しても適しているが、離散化において複雑な解析が必要であること、1 ステップあたりの計算量が多いこと、および、計算精度が低いことが欠点である。また、有限差分法と有限要素法は現象のとらえ方が異なるだけで、個々の問題においては、若干の有利・不利が生じるかもしれないが、本質的な挙動は大差ないといえる。FEM のうち、標準 Galerkin 法 (standard Galerkin approach) の結果は表 3.4 の SGA のとおりである。他の解法と比べて、計算精度は低いが、領域の分割に対する安定性が良好であることがわかる。FEM の応用としては、移流項優位の問題に対する風上化を考慮した要素構成や時間積分に Crank-Nicolson を用いる等の工夫も FDM と同様である。

次に、各解法の不安定要因を検証するために、時刻 $t = 1.0$ における相対誤差の分布を示す (図 3.4)。ただし、同一図に表示するために、異なる $\Delta t, \Delta x$ の組のものとしている。空間離散化に中心差分を用いている FTCS および CTCS は、誤差の大きさの差はあるが、他の解法に比べてほぼ一様に誤差が分布している。ただし、移流が大きい箇所では誤差が増大していることがわかる。FTUW は、境界条件の設定の問題からか境界付近の誤差が増大しているが、移流を追跡することによって、FTCS より誤差が減少している。SGA は、誤差の絶対値が大きいことが明確にわかり、流束が集中する、つまり、移流に対して拡散が相対的に大きい場合に誤差の増大傾向が見られ、この誤差の累積による不安定減少が懸念される。

総合報告集 [44] には、移流拡散方程式のテスト用モデル問題とともに、基本的な差分法や有限要素法から実用的な解法までとそれらの数値解が示されている。その中で有効な解法として、ENO (Essentially non-oscillatory) 法が挙げられる。移流拡散方程式系の数値解法に関する研究は、その他にも多くの報告があり、近年の研究として、M. OHLBERGER らの“移流項優位の放物型方程式系の弱結合に対する適応有限体積法 [37]”等が挙げられる。

以上の方法に対して、Y. TOURIGNY ら [43] は、移流拡散方程式の初期値・境界値問題に対して、特性曲線に沿って節点 (node point) を動かす有限要素法を提案し、その収束性を調べた。この有限要素法を念頭に置きながら、我々は“線の方法 (method of line, MOL)”を特

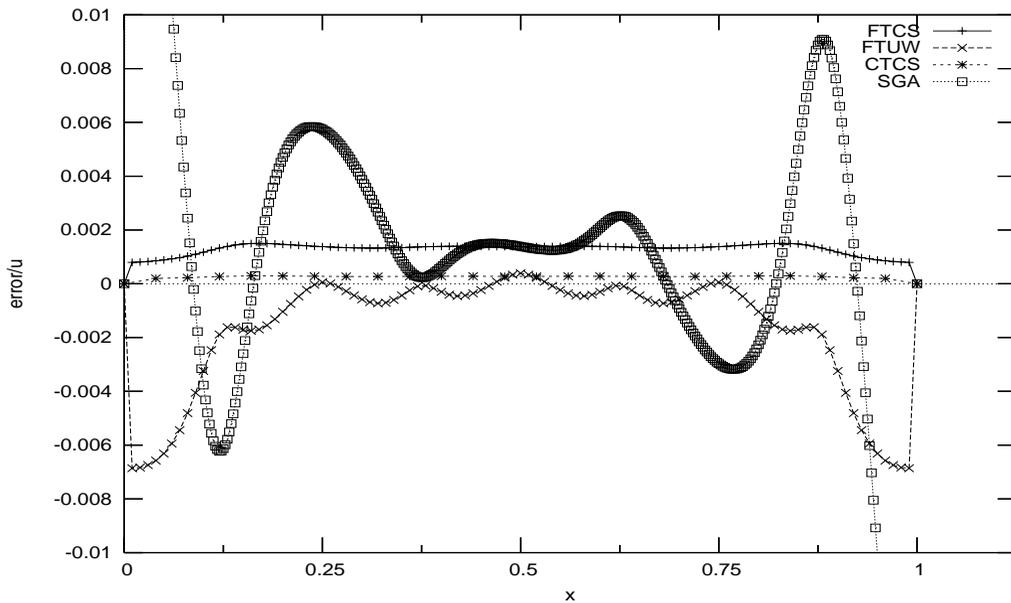


図 3.4: $t = 1.0$ における各種解法の相対誤差の分布 (FTCS: $\Delta t = 1/256, \Delta x = 1/100$, FTUW: $\Delta t = 1/256, \Delta x = 1/100$, CTCS: $\Delta t = 1/8, \Delta x = 1/25$, SGA: $\Delta t = 1/256, \Delta x = 1/400$)

性曲線に沿って適用する移動節点差分法を導く。つまり、空間の節点を特性曲線に沿って動かしていく差分法であり、本節で挙げた方法のような固定された節点上での数値解析との対比となっている。移動節点法の近年の研究は、P.HANSBO らの Crank-Nicolson 型の有限体積法 [23] の報告や田端らの時間刻み 2 次精度特性曲線有限要素法 [16, 36, 41] がある。

3.4 適用する離散変数法の収束性・安定性

本研究においては、解くべき偏微分方程式系 (移流拡散問題) を特性曲線に沿う方向の常微分方程式系に置き換え、ここで得られる常微分方程式系に適切な離散変数法を適用して数値解を得ることになる。ただし、ここで数値積分法として従来 of 解法を適用することは、安定性や計算速度の面で不十分で、新たな近似解法を構成することが必要である。したがって、新しい数値解法については注意深い数値的収束性および安定性の解析が求められる。以下では、一般の常微分方程式の初期値問題

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \quad (3.6)$$

に時間刻み幅 h で適用する Runge-Kutta 法

$$y_{n+1} = y_n + h \sum_{i=1}^s \beta_i k_i,$$

$$k_i = f \left(t_n + \gamma_i h, y_n + h \sum_{j=1}^s \alpha_{ij} k_j \right), \quad \text{for } i = 1, \dots, s, \quad (3.7)$$

$$\mathbf{A} = (\alpha_{ij}), \quad \mathbf{b} = (\beta_i), \quad \mathbf{c} = (\gamma_i)$$

の収束性解析と安定性解析を述べる．

Runge-Kutta 法の収束性解析は、第4章での収束性解析で利用するとともに、第5章、第6章で構成する新しい解法についてはこれに倣って改めて解析する．

安定性解析も同様にここでは、Runge-Kutta 法を適用する場合について解析する．安定性の基準として一般的な線型安定性解析による A 安定性を導入する．第5章、第6章で構成する解法については、これら一般的な安定性解析だけでは不十分なので、新たに NA 安定性を導入することになる．

3.4.1 収束性解析

本節では、古典的 Runge-Kutta 法の大域的な収束性 ([31] pp.47-49) を解析し、その結果から一般的な Runge-Kutta 法の場合についての結論を述べる．

まず、スカラー常微分方程式の初期値問題 (3.6) に適用する Runge-Kutta 法 (3.7) を以下のように表す．

$$y_{n+1} = y_n + h\Phi(t_n, y_n; h).$$

ここで写像 Φ は k_i を含む計算過程を意味し、 f に依存する．ここで局所離散化誤差は、

$$T_{n+1} = \frac{1}{h} \{y(t_{n+1}) - y(t_n) - h\Phi(t_n, y(t_n); h)\}$$

であり、古典的 Runge-Kutta 法は4次の公式であるから、

$$T_{n+1} = A(t_n, y(t_n))h^4 + O(h^5)$$

が成り立つ．また、係数 $A(t_n, y(t_n))$ は f の高階偏導函数の複雑な組合せとなる．ここで、考えている区間で f が少なくとも4回連続偏微分可能であって、ある正数 h_0 をとれば、 $h < h_0$ である限り

$$|T_{n+1}| \leq \bar{A}h^4 \quad (3.8)$$

をみたま正定数 \bar{A} が存在すると仮定する．

真の解および数値解の漸化式

$$y(t_{n+1}) = y(t_n) + h\Phi(t_n, y(t_n); h) + hT_{n+1}$$

$$y_{n+1} = y_n + h\Phi(t_n, y(t_n); h)$$

より，大域誤差 $e_{n+1} = y(t_{n+1}) - y_{n+1}$ については，

$$e_{n+1} = e_n + h\{\Phi(t_n, y(t_n); h) - \Phi(t_n, y_n; h)\} + hT_{n+1}$$

を得る．ここで問題となるのは Φ に関する差の項の評価であるが，各 k_i の y_n に対する依存性を強調して， $k_i(y_n)$ のように記すこととする． f が Lipschitz 連続であると仮定すると，

$$|k_1(y(t_n)) - k_1(y_n)| = |f(t_n, y(t_n)) - f(t_n, y_n)| \leq L|e_n|$$

は自明である．ただし， L は f の y に関する Lipschitz 定数である．同様に k_2, k_3, k_4 についても，

$$\begin{aligned} & |k_2(y(t_n)) - k_2(y_n)| \\ &= \left| f\left(t_n + \frac{h}{2}, y(t_n) + \frac{h}{2}k_1(y(t_n))\right) - f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1(y_n)\right) \right| \\ &\leq L \left(|e_n| + \frac{h}{2}L|e_n| \right) \\ &= L \left(1 + \frac{h}{2}L \right) |e_n| \end{aligned}$$

$$|k_3(y(t_n)) - k_3(y_n)| \leq L \left(1 + \frac{1}{2}hL + \frac{1}{4}(hL)^2 \right) |e_n|$$

$$|k_4(y(t_n)) - k_4(y_n)| \leq L \left(1 + hL + \frac{1}{2}(hL)^2 + \frac{1}{4}(hL)^3 \right) |e_n|$$

を得る．したがって，

$$|\Phi(x_n, y(x_n); h) - \Phi(x_n, y_n; h)| \leq L \left(1 + 3hL + (hL)^2 + \frac{1}{4}(hL)^3 \right) |e_n|$$

であり， h が十分小さいとき hL はある正定数以下であるとみなして，

$$|\Phi(x_n, y(x_n); h) - \Phi(x_n, y_n; h)| \leq L'|e_n| \tag{3.9}$$

と評価することができる．これと式 (3.8) をあわせれば，差分不等式

$$|e_{n+1}| \leq (1 + hL')|e_n| + \bar{A}h^5$$

が得られる．ここで $e_0 = 0$ を考慮すると，

$$|e_n| \leq \frac{\bar{A}h^4}{L'}((1 + hL')^n - 1), \quad (n = 0, 1, 2, \dots)$$

を導くことができる．さらに $(1 + hL')^n < e^{nhL'}$ であるから， h や n に依存しない定数 C を用いて，

$$|e_n| \leq Ch^4 \tag{3.10}$$

と記述できる．つまり， $h \downarrow 0$ とするとき， $|e_n| \rightarrow 0$ となり，古典的 Runge-Kutta 法の大域的収束性が示された．

以上の解析は一般的な Runge-Kutta 法についても適用可能で，十分滑らかな解をもち Lipschitz 連続な問題に対して，解法が適合 (consistent) であれば Runge-Kutta 法は収束し，その次数が p 次であれば大域的収束性も p 次となる．

3.4.2 A 安定性解析

数値計算の過程で生じた誤差が，引き続く計算過程に伝播していくとき，高々有界な範囲にとどまるならば，数値的に安定であるという．ここでは，Runge-Kutta 法について，安定性の指標として一般的である A 安定性を導入する ([31] pp.73-78, p.102)．

線型テスト問題は定数 λ をもつ微分方程式の初期値問題

$$y' = \lambda y, \quad \lambda \in \mathbb{C}, \quad \Re \lambda < 0, \quad y(0) = y_0 \quad (3.11)$$

であって，これに一般の Runge-Kutta 法 (3.7) を適用すると，

$$y_{n+1} = R(z)y_n, \quad R(z) = \frac{\det(\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^T)}{\det(\mathbf{I} - z\mathbf{A})} \quad (3.12)$$

と表わされる．ここで， $z = \lambda h$ を複素数とし， $\mathbf{e} = [1, 1, \dots, 1]^T$ を導入する．また， h は $\mathbf{I} - z\mathbf{A}$ が正則となるようにとられているとする． $R(z)$ は安定性因子と呼ばれ，適用する公式ごとに与えられることがわかる．

ここで，絶対安定領域を複素平面上に

$$\mathcal{R} = \{z \in \mathbb{C}; |R(z)| < 1\}$$

と定義する．また， \mathcal{R} と実軸との共通部分を絶対安定区間という． λh を絶対安定領域に入るように選べば，Runge-Kutta 法の与える数値解は，ステップごとに一定の誤差が混入する

$$y_{n+1} = R(z)y_n + d$$

であっても，その解が $n \rightarrow \infty$ のとき指数函数的に増大することがない．また， \mathcal{R} が複素左半平面をすべて含むとき，その解法は A 安定 (A -stable) な解法と呼ぶ．

2次から4次の次数と段数が等しい陽的 Runge-Kutta 法の安定性因子は，すべて

$$R_s(z) = \sum_{j=0}^s \frac{z^j}{j!}, \quad s = 2, 3, 4$$

と表わされ，その絶対安定領域を図示すると以下のようなになる．領域は，実軸について対称であるから，ここでは上半平面の様子を示す．

図 3.5 からわかるように，適当な絶対安定領域が存在するためには， $z = 0$ が安定領域に含まれ，その境界である必要がある (零安定)．また，陽的 Runge-Kutta 法は，絶対安定区間

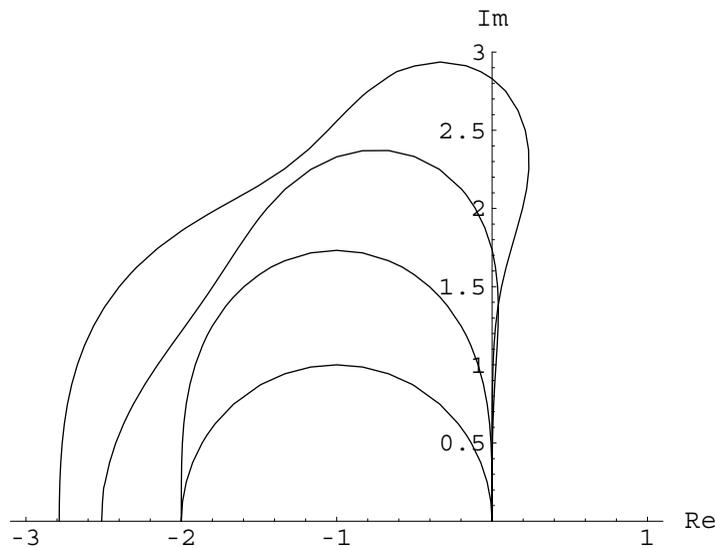


図 3.5: 陽的 Runge-Kutta 法の絶対安定領域
(領域は曲線の内側, 内側の領域から順に $s = 1, 2, 3, 4$)

が有界で, A 安定にはなりえないことが知られている. 我々が採用する古典的 Runge-Kutta 法は 4 段 4 次であるので, 絶対安定区間はおよそ $[-2.7, 0]$ であることがわかる.

一方, A 安定な Runge-Kutta 法として, 陰的解法である s 段 Butcher-Kuntzmann 公式が挙げられる. この s 段公式の次数は $2s$ 次であることが知られるが, 例えば, 2 段 4 次公式の安定性因子は,

$$R(z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2}$$

で表わされ, これは, e^z の Padé 近似有理函数となり, 絶対安定領域は複素左半平面と一致する. その他では, 陰的 Runge-Kutta 法のうち, Radau 型・Lobatto 型の選点法による公式もすべての段数について A 安定であることが知られている.

線型常微分方程式系の係数行列の固有値のうち, 絶対値が最大のものと最小のものとの比を硬度比といい, 硬度比が大きい系を硬い系と呼ぶ [31]. 対象とする常微分方程式系が硬い系である場合, A 安定な解法を適用することが望ましいが, 実用のために A 安定を緩和する指標も定義されている. 原点を頂点とし, 左側に非有界で, 実軸とある角度 α をもつ扇型で, 絶対安定領域に含まれるものが存在するとき, $A(\alpha)$ 安定という. この安定性の狭義で, 実軸上の負の区間全体が安定領域に含まれる場合は, $A(0)$ 安定となる.

3.4.3 NA安定性解析

我々が対象とする変数係数線型常微分方程式系

$$\mathbf{y}'(t) = \mathbf{L}(t)\mathbf{y}(t) + \mathbf{F}(t) \quad (3.13)$$

の場合は, $\mathbf{L}(t)$ が定数であることを想定するテスト方程式 (3.11) を用いる線型安定性解析では不十分で AN 安定性 [7] の解析をしなければならない. すなわち, 非自励系 (non-autonomous) のテスト方程式

$$y'(t) = \lambda(t)y(t) \quad (t \geq 0), \quad \lambda(t) \in \mathbb{C}, \quad \Re\lambda(t) < 0, \quad y(0) = y_0 \quad (3.14)$$

を用いて, これに数値解法を適用したときの安定性因子を導出し, 解法が AN 安定 (非自励系における A 安定, A -stable for non-autonomous) であるかどうかを調べる. しかし, 第5章, 第6章で述べる解法はいずれも, その公式の特徴から特殊な場合を除いて従来の AN 安定であることを期待できない. そこで, 時間刻み幅 h が十分小さいとき, NA 安定領域 \mathcal{R} を定義し, その存在領域を解析する. これを新たに NA 安定性解析 [11] と呼ぶ. NA 安定性解析において, NA 安定領域が, 複素左半平面に相当する領域をすべて含むとき, 解法は AN 安定であり, これを無条件 NA 安定と呼ぶ. これに対して, NA 安定領域が複素左半平面に相当する領域内に存在するが, そのすべてをは含まないとき, 安定領域が時間刻み幅 h に制限されるという意味で, 条件付き NA 安定と呼ぶこととする. NA 安定性因子は, 複数の複素変数で表現されるため解析が困難となる. 詳細は各章の解法それぞれに異なる手法として, 5.5.2節, 6.4節で述べるが, 通常の s 段 Runge-Kutta 法ならば, NA 安定性因子は, s 次元複素ベクトル z の要素を $z_i = \lambda(t_n + \gamma_i h)$, ($i = 1, 2, \dots, s$) と定義し, これを対角要素にもつ対角行列 $Z = \text{diag}(z_i)$ を用いて,

$$R(z) = \frac{\det(\mathbf{I} - \mathbf{Z}\mathbf{A} + \mathbf{Z}\mathbf{e}\mathbf{b}^T)}{\det(\mathbf{I} - \mathbf{Z}\mathbf{A})}$$

であり, NA 安定領域

$$\mathcal{R} \equiv \{z; |R(z)| < 1\} \supset \mathbb{C}^{s,-} \equiv \{z; \Re z_i < 0 \forall i\}$$

の存在範囲を追跡することになる. AN 安定, すなわち, ここでいう無条件 NA 安定な解法である2段4次 Butcher-Kuntzmann 公式の NA 安定性因子は,

$$R(z) = \frac{1 + \frac{1}{4}z_1 + \frac{1}{4}z_2 + \frac{1}{12}z_1z_2}{1 - \frac{1}{4}z_1 - \frac{1}{4}z_2 + \frac{1}{12}z_1z_2}$$

となり, 安定領域の存在範囲の確認は何とか可能であるが, これよりも複雑になる ROW 型公式やブロック Rosenbrock 法の NA 安定性解析の困難さも想像に難くない. 各章の NA 安定性解析においては, NA 安定領域を定義する複素パラメタを近似的に少なくすることで, その領域の図的表現を試みている.

3.5 変数係数線型常微分方程式系と準解析的数値解法

変数係数線型常微分方程式系 (3.13) に適用する数値解法の先行研究として, 準解析的 (Semi-analytic) な数値解法が挙げられる. ここでは, これらの形式と特徴を簡潔に述べる. 多くの手法において, 基本解 $Y(t)$ の同次線型系であり, (3.13) に対応して

$$Y' = L(t)Y, \quad Y(0) = I, \quad (3.15)$$

と表わされる. ここで, I は $d \times d$ 次元単位行列である. 基本解は (3.13) に対応する同次方程式の解を $y(t) = Y(t)y_0$ のように与えるので, このような手法を準解析的と呼ぶ. 準解析的な解法は, 1954 年に W. MAGNUS によって提案されたもので, 基本解 $Y(t)$ を局所的に

$$Y(t) = \exp(\Omega(t))$$

と表し, さらに, Ω を無限級数

$$\begin{aligned} \Omega(t) = & \int_0^t L(t_1) dt_1 + \frac{1}{2} \int_0^t \left[L(t_1), \int_0^{t_1} L(t_2) dt_2 \right] dt_1 \\ & + \frac{1}{4} \int_0^t \left[L(t_1), \int_0^{t_1} \left[L(t_2), \int_0^{t_2} L(t_3) dt_3 \right] dt_2 \right] dt_1 \\ & + \frac{1}{12} \int_0^t \left[\left[L(t_1), \int_0^{t_1} L(t_2) dt_2 \right], \int_0^{t_1} L(t_3) dt_3 \right] dt_1 + \cdots \end{aligned} \quad (3.16)$$

で表現する手法である [29]. ただし上式において, 括弧 $[,]$ は行列の交換子である. より最近の研究では, A. ISERLES らがこの方法について数値解析的な視点から述べている [26].

実際の Magnus 級数展開 (3.16) は再帰的に表わされる. つまり,

$$\Omega(t) = \sum_{n=1}^{\infty} \Omega_n(t) \quad (3.17)$$

において,

$$\Omega_1(t) = \int_0^t L(\tau) d\tau, \quad \Omega_n(t) = \sum_{j=1}^{n-1} \frac{B_j}{j!} \int_0^t S_n^{(j)}(\tau) d\tau \quad (n \geq 2)$$

と記述でき, B_j は Bernoulli 数であるから再帰的であり, さらに $S_n^{(j)}(t)$ も

$$S_n^{(1)}(t) = [\Omega_{n-1}(t), L(t)], \quad S_n^{(j)}(t) = \sum_{m=1}^{n-j} \left[\Omega_m(t), S_{n-m}^{(j-1)}(t) \right], \quad 2 \leq j \leq n-1$$

のような再帰関係となっている. S. BLANES ら ([2]) は, 級数展開 (3.17) をある有限回数で打ち切り, 積分には数値積分を適用して, いくつかの数値計算式を導出した.

一方, A. ISERLES ([25]) は, 基本解 $Y(t)$ を以下のように近似した.

$$\exp(B_0(t)) \exp(B_1(t)) \cdots \exp(B_m(t))$$

そして、これらの行列 $B_0(t), B_1(t), \dots, B_m(t)$ は再帰的に生成する。さらに Iserles は、指数関数の Padé 近似を適用することで、いくつかの数値スキームを導き出した。また、様々な問題に対する数値実験が試みられたが、その問題の次元数 d は、我々が対象としている大きさよりもかなり小さいものであった ([2, 25])。

I. BERNSTEIN らは、行列 $L(t)$ の固有値の対を用いて基本解を展開することを試みた ([1])。彼らは、係数行列 $L(t)$ に比べて数値解が急速に振動するような場合について、数値スキームを導出した。

以上の先行研究の調査から我々は、大規模な線型常微分方程式系に適用可能な並列計算向きの解法はまだ十分には開発されていないと考える。この目的を達成するために、数値解法における以下の問題を解決しなければならない。

- 方程式系 (3.13) の線形性の利点を生かすこと。
- 高次の精度を達成する数値的収束性をもつこと。
- 確実な数値計算のための安定性を達成すること。

第4章 移流拡散方程式の特性曲線差分法

本章は、本論文で取り扱う課題の問題提起となる部分である。我々は、移流項優位の移流拡散方程式に対しては、その流れに沿う方向の微分が自然と考える。そこで、特性曲線を仮定し、その特性曲線上の節点における数値解を差分化により求めていく方法“移流拡散方程式の特性曲線差分法”を提案する。この手法により、移流拡散問題は偏微分方程式から、特性曲線に沿う方向において変数係数線型の常微分方程式に置き換えられる。ここで解くべき方程式系は、領域分割や並列計算による高速化が期待できるが、解析の結果、収束性や安定性のための厳しい条件が必要であることがわかり、これが次章以降に続く大きな課題となる。

本章では、論文 [13] に基づいて、まず、4.1 節において特性曲線法による空間離散化の方法を述べる。続く 4.2 節では解法の数値的収束性および安定性の注意深い解析により、新しい方法としての問題点を確認し、4.3 節で実際に数値計算により解析結果を検証する。以上の問題点を確認したのちに、今後の展望として 4.4 節において、まず、当初に考えていた陽的な離散変数法を用いての並列計算を実施しその限界を示し、そこでの困難を解決する予備実験として、陰的 Runge-Kutta 法適用時の数値解の安定性と計算量の増大の問題点について検討する。

4.1 特性曲線法による空間離散化

移流拡散方程式の初期値境界値問題 (3.1)-(3.3) について、ここでは離散化のための新たな条件を考える。

$$\begin{aligned} \varepsilon &= \text{const.}, \quad 0 < \varepsilon \ll \sqrt{a^2 + b^2}, \quad a(x, t) \geq \alpha > 0 \text{ for } (x, t) \in Q, \\ b(0, t) &= b(1, t) = 0 \text{ for } t \in [0, T]. \end{aligned}$$

以下においては簡単のため、 $a = 1$ と定めることがある。 b の条件は、境界での流速が 0 であることを意味する。移流と拡散の関係は、基本的に移流項優位 (advection dominant) であることを意味するが、流束がいくつかの点に集中し局所的には拡散が強い場合があることも含まれる。

4.1.1 特性曲線に沿った格子

自然数 N に対して $h = 1/N$ と定め、 $t = 0$ における区間 Ω の分割を次のように定義する。

$$\Delta_h(0) : 0 = x_0(0) < x_1(0) < \cdots < x_{N-1}(0) < x_N(0) = 1.$$

ここですべての $i (i = 1, \dots, N - 1)$ についての空間上の点 $x_i = x_i(t)$ は、初期値 $x_i(0)$ の常微分方程式

$$\frac{dx_i(t)}{dt} = \frac{b}{a}(x_i(t), t), \quad t > 0 \quad (4.1)$$

の i 番目の解である．この方程式群の解を特性曲線とする．式 (4.1) をベクトル表記すると、

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{b}}{\mathbf{a}}(\mathbf{x}, t), \quad t > 0 \quad (4.2)$$

となる．ただし、 $\mathbf{x} = [x_1, x_2, \dots, x_{N-1}]^T$ は先に定めた分割の節点であり、 $\frac{\mathbf{b}}{\mathbf{a}}(\mathbf{x}, t) = \left[\frac{b}{a}(x_1, t), \frac{b}{a}(x_2, t), \dots, \frac{b}{a}(x_{N-1}, t) \right]^T$ は、各節点での流速函数のベクトルである．ここで $a(x, t)$ 、 $b(x, t)$ は以下の条件を満たすと仮定する．

$$\left\| \frac{d}{dx} \frac{b}{a} \right\|_{L^\infty(Q)} \leq L = \text{const.} \quad (4.3)$$

このとき、 $x_i(t)$ は大域的に存在し、Gronwall の補題より以下の不等式を得る．

$$(x_i(0) - x_{i-1}(0))e^{-Lt} \leq x_i(t) - x_{i-1}(t) \leq (x_i(0) - x_{i-1}(0))e^{Lt}, \quad \text{for } i = 1, \dots, N. \quad (4.4)$$

これにより、式 (4.1) の解の列は区間 $\Omega = (0, 1)$ において、

$$\Delta_h(t) : 0 = x_0(t) < x_1(t) < \dots < x_{N-1}(t) < x_N(t) = 1$$

のような分割を矛盾なく構成する．ゆえに、 (x, t) 平面におけるすべての時刻 t について、分割 $\Delta_h(t)$ は特性曲線 (4.1) 沿う方向に得られるといえる．つまり、次節で述べる空間離散化の格子点の位置が時間の変化とともに、特性曲線に沿う方向に変化していくのである．

図 4.1 は、特性曲線および分割 $\Delta_h(t)$ の例を示している．

4.1.2 特性曲線に沿った空間差分化

ここで、問題 (3.1)–(3.3) は、式 (4.1) から得られる特性曲線 $x_i(t)$ ($i = 1, \dots, N - 1$) に沿う方向の解 $u_i(t) \equiv u(x_i(t), t)$ を求めることに置き換えられた．つまり、時間とともに移動する節点上の u の常微分方程式についての離散化を考えるのである．

特性曲線 $x_i(t)$ に沿う方向の u の導函数は、微分の連鎖則より、

$$\frac{du_i(t)}{dt} \equiv \frac{d}{dt}u(x_i(t), t) = u_t(x_i(t), t) + u_x(x_i(t), t)\frac{dx_i(t)}{dt}$$

と与えられるので、問題 (3.1)–(3.3) および方程式 (4.1) から、

$$\frac{du_i(t)}{dt} = a^{-1}(x_i(t), t)[\varepsilon u_{xx}(x_i(t), t) + f(x_i(t), t)], \quad (4.5)$$

$$u_i(0) = u_0(x_i), \quad \text{for } i = 1, \dots, N - 1 \quad (4.6)$$

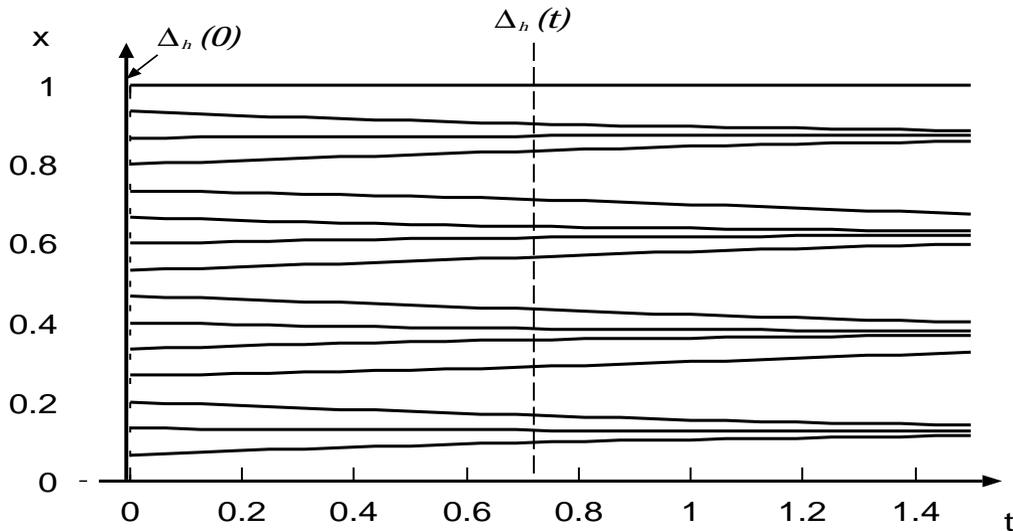


図 4.1: 特性曲線群とある時刻における分割

が得られる .

ここで , 式 (4.5) における 2 階偏導函数 u_{xx} を空間離散化する必要があるので , 我々は不等間隔格子上での有限差分を用いる . $u_i(t)$ の近似値として $U_i(t)$ を導入して , 以下の有限差分近似

$$u_{xx}(x_i(t), t) \approx \frac{1}{h_{i+1}h_i} [(1 + q_{i+1})U_{i-1} - 2U_i + (1 - q_{i+1})U_{i+1}] \quad (4.7)$$

を適用する . ここで $h_i = x_i(t) - x_{i-1}(t)$, $q_{i+1} = \frac{h_{i+1} - h_i}{h_{i+1} + h_i}$ とする .

解 $u(x, t)$ が $C^3(Q)$ 級であると仮定するとき , 空間差分化時の打ち切り誤差すなわち

$$u_{xx}(x_i(t), t) - \frac{1}{h_{i+1}h_i} [(1 + q_{i+1})u(x_{i-1}(t), t) - 2u(x_i(t), t) + (1 - q_{i+1})u(x_{i+1}(t), t)]$$

は以下ようになる ([13] Appendix pp.15-16) .

$$\frac{1}{3}(h_{i+1} - h_i)u_{xxx}(\theta_i(t), t). \quad (4.8)$$

ただし , $\theta_i(t) \in (x_{i-1}(t), x_{i+1}(t))$ である .

ゆえに , 方程式系 (4.5) は式 (4.7) を用いて近似すると ,

$$U'_i = a^{-1}(x_i(t), t) \left[\frac{\varepsilon}{h_{i+1}h_i} \{(1 + q_{i+1})U_{i-1} - 2U_i + (1 - q_{i+1})U_{i+1}\} + f(x_i(t), t) \right], \quad \text{for } i = 1, \dots, N - 1 \quad (4.9)$$

となり , この空間差分化の局所打ち切り誤差 (LTE) は式 (4.8) から ,

$$l_i(t) = \frac{1}{3} \frac{\varepsilon(h_{i+1} - h_i)}{a(x_i(t), t)} u_{xxx}(\theta_i(t), t), \quad \theta_i(t) \in (x_{i-1}(t), x_{i+1}(t)) \quad (4.10)$$

と与えられる．

また，以下の表記を用いると，

$$r_{i+1} = \frac{\varepsilon}{h_{i+1}h_i}, \quad a_i = a(x_i(t), t), \quad f_i = f(x_i(t), t),$$

方程式系 (4.9) は，次のような行列形式で表すことができる．

$$U' = (DA)U + F. \quad (4.11)$$

ここで $U = [U_1, U_2, \dots, U_{N-1}]^T$ であり，行列 D, A およびベクトル F は以下のように与えられる．

$$\begin{aligned} D &= \text{diag} \left(\frac{r_{i+1}}{a_i} \right), \\ A &= \begin{bmatrix} -2 & (1 - q_2) & & & 0 \\ (1 + q_3) & -2 & (1 - q_3) & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & (1 - q_{N-1}) \\ 0 & & & (1 + q_N) & -2 \end{bmatrix} \\ &= \text{tridiag} \{ (1 + q_{i+1}), -2, (1 - q_{i+1}) \}, \\ F &= \left[\frac{f_1}{a_1} + r_2(1 + q_2)u_0, \frac{f_2}{a_2}, \dots, \frac{f_{N-2}}{a_{N-2}}, \frac{f_{N-1}}{a_{N-1}} + r_N(1 - q_N)u_N \right]^T. \end{aligned}$$

ここで D, A と F の要素中にベクトル $x = [x_1, x_2, \dots, x_{N-1}]^T$ が含まれるのを考慮する場合は，必要に応じて，

$$U' = (DA)U + F = B(x, t)U + F(x, t) \quad (4.12)$$

のように表記する．

以上より，我々の方法は特性曲線 (4.2) および u の空間離散化 (4.12) の方程式をさらにあわせた大連立系とみることができ，式 (3.3) から得られる初期値を用い，各方程式に対して適当な数値積分法を適用することで，数値的に解を求めていくことになる．

我々はこの方法を移流拡散方程式の特性曲線差分法と呼ぶ．この新しい方法については，新たな問題を引き起こす可能性があるので，注意深い解析が必要である．

4.2 数値スキームの収束性と安定性

移流拡散方程式の特性曲線差分法の収束性および安定性を解析する．空間および時間方向の局所離散化誤差から大域誤差を評価し，収束性の条件を見積もる．安定性については，粗い解析であるが，Gershgorin の定理を用いて，絶対安定区間を見積もる．この収束性および安定性の解析の結論は，続く 4.3 節において数値実験により検証する．

4.2.1 局所離散化誤差

式 (4.11) に含まれる空間方向の打ち切り誤差は，特性曲線に沿った空間離散化によるので， u_{xx} の差分化に依存する．式 (4.10) からわかるように，この誤差は， $\bar{h} = \max_i h_i$ とするとき，

$$\ell_i(t) = \frac{1}{3} \frac{\varepsilon(h_{i+1} - h_i)}{a(x_i(t), t)} u_{xxx}(\theta_i(t), t) = o(\bar{h}) \quad (4.13)$$

となる．この式は，方程式系 (4.11) の時間方向の離散化誤差についての解析時に再び利用する．

2つの常微分方程式系 (4.2) と (4.11) の組に，ある s 段 p 次の Runge-Kutta 法を適用し，時間刻み幅を Δt として数値解を求めることとする．ここで，Runge-Kutta 法の Butcher 配列で与えられる行列 A ，ベクトル b, c は，通常のように，以下のものを用いる．

$$A = (\alpha_{ij}), \quad b = (\beta_i), \quad c = (\gamma_i).$$

一般的な常微分方程式 $dy/dt = f(y)$ に Runge-Kutta 法を適用するときの局所離散化誤差 T_{n+1} は，前の時間レベル n までの誤差の累積がないものとして，

$$T_{n+1} = y(t_{n+1}) - y_{n+1} = y(t_{n+1}) - \left(y(t_n) + \Delta t \sum_{i=1}^s \beta_i f(Y_i) \right) = O(\Delta t^{p+1}) \quad (4.14)$$

のように得られる．ただし，段値 (stage value) を $Y_i = y(t_n) + \Delta t \sum_{j=1}^s \alpha_{ij} f(Y_j)$ とする．

4.2.2 大域誤差

以後簡単のため $a(x, t) \equiv 1$ と正規化する．また，特性曲線の離散化には誤差が含まれないと仮定する．さらに，行列のノルムはベクトルのノルムに従属することとする．

まず，大域誤差を解析する．大域誤差には，時間レベル n までの空間方向の打ち切り誤差と時間方向の離散化誤差が含まれるが，ここでは u の離散化方程式に打ち切り誤差が含まれているとするときの離散化誤差を評価することになる．つまり，数値解 U_i の離散化方程式 (4.11) が打ち切り誤差を含んでいるので，真の解 $u(x, t)$ は，

$$\frac{d\mathbf{u}}{dt} = (DA)\mathbf{u} + \mathbf{F}(t) + \mathbf{L}(t) \quad (4.15)$$

にしたがう．ここで， $\mathbf{u} = [u(x_1, t), \dots, u(x_{N-1}, t)]^T$ ， $\mathbf{L}(t) = [\ell_1(t), \dots, \ell_{N-1}(t)]^T$ である．これに Runge-Kutta 法を適用すると，

$$\mathbf{u}(t_{n+1}) = \mathbf{u}(t_n) + \Delta t \sum_i \beta_i \left\{ (DA)\mathbf{U}_i^{(n)} + \mathbf{F}(t_n + \gamma_i \Delta t) + \mathbf{L}(t_n + \gamma_i \Delta t) \right\} + T_{n+1} \quad (4.16)$$

となる．ただし，中間の段値は以下ようになる．

$$U_i^{(n)} = \mathbf{u}(t_n) + \Delta t \sum_j \alpha_{ij} \left\{ (DA)U_j^{(n)} + \mathbf{F}(t_n + \gamma_j \Delta t) + \mathbf{L}(t_n + \gamma_j \Delta t) \right\}.$$

一方，式(4.12)の数値解を \mathbf{v}_n とするとき，

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \sum_i \beta_i \left\{ (DA)\mathbf{V}_i^{(n)} + \mathbf{F}(t_n + \gamma_i \Delta t) \right\} \quad (4.17)$$

となる．同様に，段値は以下ようになる．

$$\mathbf{V}_i^{(n)} = \mathbf{v}_n + \Delta t \sum_j \alpha_{ij} \left\{ (DA)\mathbf{V}_j^{(n)} + \mathbf{F}(t_n + \gamma_j \Delta t) \right\}.$$

したがって，大域誤差 $\mathbf{E}^n \equiv \mathbf{u}(t_n) - \mathbf{v}_n$ は，以下の漸化式で得られる．

$$\begin{aligned} \mathbf{E}^{n+1} &= \mathbf{E}^n + \Delta t \sum_i \beta_i \left\{ (DA)\mathbf{E}_i^{(n)} + \mathbf{L}(t_n + \gamma_i \Delta t) \right\} + T_{n+1} \\ &= \mathbf{E}^n + \Delta t (DA) \sum_i \beta_i \mathbf{E}_i^{(n)} + \Delta t \sum_i \beta_i \mathbf{L}(t_n + \gamma_i \Delta t) + T_{n+1}. \end{aligned} \quad (4.18)$$

ただし， $i = 1, \dots, s$ について，

$$\begin{aligned} \mathbf{E}_i^{(n)} &= U_i^{(n)} - V_i^{(n)} \\ &= \mathbf{E}^n + \Delta t \sum_j \alpha_{ij} (DA)\mathbf{E}_j^{(n)} + \Delta t \sum_j \alpha_{ij} \mathbf{L}(t_n + \gamma_j \Delta t) \end{aligned} \quad (4.19)$$

となる．記号

$$\boldsymbol{\mathcal{E}}^{(n)} = \begin{bmatrix} \mathbf{E}_1^{(n)} \\ \vdots \\ \mathbf{E}_s^{(n)} \end{bmatrix}, \quad \mathbf{1}_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

を導入し， \mathbf{I}_n を n 次元の単位行列として，さらに，

$$A \otimes B \equiv \begin{bmatrix} a_{11}B & a_{12}B & \cdots \\ a_{21}B & a_{22}B & \\ \vdots & & \ddots \end{bmatrix}$$

のように Kronecker 積を用いると，式 (4.19) は，以下のように評価できる．

$$\begin{aligned}
\{\mathbf{I}_{(N-1)s} - \Delta t \mathbf{A} \otimes (DA)\} \boldsymbol{\varepsilon}^{(n)} &= \mathbf{1}_s \otimes \mathbf{E}^{(n)} + \Delta t \begin{bmatrix} \sum \alpha_{1j} \mathbf{L}(t_n + \gamma_j \Delta t) \\ \sum \alpha_{2j} \mathbf{L}(t_n + \gamma_j \Delta t) \\ \vdots \\ \sum \alpha_{sj} \mathbf{L}(t_n + \gamma_j \Delta t) \end{bmatrix} \\
&= \mathbf{1}_s \otimes \mathbf{E}^{(n)} + \Delta t (\mathbf{A} \otimes \mathbf{1}_s) \begin{bmatrix} \mathbf{L}(t_n + \gamma_1 \Delta t) \\ \mathbf{L}(t_n + \gamma_2 \Delta t) \\ \vdots \\ \mathbf{L}(t_n + \gamma_s \Delta t) \end{bmatrix} \\
&= \mathbf{1}_s \otimes \mathbf{E}^{(n)} + O(\Delta t) o(\bar{h}).
\end{aligned}$$

ゆえに， Δt が十分小さいとき，行列 $\mathbf{I}_{(N-1)s} - \Delta t \mathbf{A} \otimes (DA)$ が正則であるので，

$$\begin{aligned}
\boldsymbol{\varepsilon}^{(n)} &= \{\mathbf{I}_{(N-1)s} - \Delta t \mathbf{A} \otimes (DA)\}^{-1} \{\mathbf{1}_s \otimes \mathbf{E}^{(n)} + O(\Delta t) o(\bar{h})\} \\
&= \{\mathbf{I}_{(N-1)s} + \Delta t \mathbf{A} \otimes (DA) + \Delta t^2 (\mathbf{A} \otimes (DA))^2 + \cdots\} \\
&\quad \times \{\mathbf{1}_s \otimes \mathbf{E}^{(n)} + O(\Delta t) o(\bar{h})\}
\end{aligned} \tag{4.20}$$

とみることができる．

$a(x, t) \equiv 1$ と仮定したので，行列の内積 DA はより簡潔に表すことができ，

$$\begin{aligned}
DA &= -\varepsilon \begin{bmatrix} \frac{2}{h_1 h_2} & \frac{-2}{h_1(h_1 + h_2)} & & 0 \\ \frac{-2}{h_2(h_2 + h_3)} & \frac{2}{h_2 h_3} & \frac{-2}{h_2(h_2 + h_3)} & \\ & \ddots & \ddots & \ddots \\ 0 & & & \ddots \end{bmatrix} \\
&= -\varepsilon \text{tridiag} \left\{ \frac{-2}{h_i(h_i + h_{i+1})}, \frac{2}{h_i h_{i+1}}, \frac{-2}{h_i(h_i + h_{i+1})} \right\}
\end{aligned} \tag{4.21}$$

のようになる．さらに以下での解析のために， $\mathbf{A} = -\frac{1}{\varepsilon} DA$ とする．

ここで， $\{h_i\}$ の最小値を \underline{h} と記し，やや強い仮定であるが，我々の数値スキームの空間および時間方向の刻み幅の割合を

$$\frac{\Delta t}{\underline{h}^2} = \text{const.} \tag{4.22}$$

のように定義する．これによって， $\Delta t \|DA\| = C(\text{const.})$ が得られ，式 (4.20) から以下の不等式が得られる．

$$\|\boldsymbol{\varepsilon}^{(n)}\| \leq (1 + C) \|\mathbf{E}^{(n)}\| + O(\Delta t) o(\bar{h}). \tag{4.23}$$

Kronecker 積を用いることで，さらに，式 (4.18) を

$$\mathbf{E}^{n+1} = \mathbf{E}^n + \Delta t (DA) (\mathbf{b} \otimes \mathbf{1}_s)^T \boldsymbol{\varepsilon}^{(n)} + O(\Delta t) o(\bar{h}) + O(\Delta t^{p+1})$$

のように変形でき、これに式 (4.23) の見積もりを置き換えると、結果として、

$$\begin{aligned}\|E^{n+1}\| &\leq \|E^n\| + C(1+C)\|E^n\| + O(\Delta t^2)o(\bar{h}) + O(\Delta t)o(\bar{h}) + O(\Delta t^{p+1}) \\ &= \|E^n\| + C'\|E^n\| + O(\Delta t^2)o(\bar{h}) + O(\Delta t)o(\bar{h}) + O(\Delta t^{p+1}) \\ &= (1+C')\|E^n\| + O(\Delta t)o(\bar{h}) + O(\Delta t^{p+1})\end{aligned}\quad (4.24)$$

となる。ここで、 C' は仮定の式 (4.22) を用いて $C(1+C)$ を置き換えたものである。上記の大域誤差の評価式は最終の時間レベル M まで成立し、初期時間レベルにおいては $\|E^0\| = 0$ のように誤差は含まれないから、結論として、大域誤差の評価式

$$\|E^M\| = o(\bar{h}) + O(\Delta t^p) \quad (4.25)$$

を得ることができる。

4.2.3 収束性の評価

初期時間レベルにおいて、空間方向の節点の間隔はすべて等間隔 $h(t=0) = \Delta x$ とする。ここで $\Delta x \downarrow 0$ のとき、 $\underline{h} = O(\Delta x)$ にしたがうと仮定できるならば、刻み幅についての条件式 (4.22) は、

$$\frac{\Delta t}{\Delta x^2} = \text{const.} \quad (4.26)$$

のように数値計算の初期値として設定できる値に置き換えることができる。つまり、条件 (4.26) のもとでは、大域誤差の評価式 (4.25) は、数値スキームが収束することを意味する。この収束の次数は、時間積分に p 次の Runge-Kutta 法を適用するとき、 $O(\Delta t^p)$ である。

4.2.4 数値的安定性の基準

ここでは、やや粗い解析ではあるが、Gershgorin の定理 ([35] p.86, [46] p.333) を用いることでスキームの数値的安定性を評価する。

特性曲線に沿って空間離散化した常微分方程式 (4.11)、すなわち、

$$U' = (DA)U + F$$

の安定性を調べる。式 (4.21) において $DA = -\varepsilon A$ であるから、まず、行列 A の固有値を評価する。これは、Gershgorin の定理によって得られる。

複素平面上の閉円盤 \mathcal{D}_i ($i = 1, 2, \dots, N-1$) は以下のように表わされる。

$$\begin{aligned}\mathcal{D}_j &= \left\{ z \in \mathbb{C} : \left| z - \frac{2}{h_j h_{j+1}} \right| < \frac{2}{h_j h_{j+1}} \right\} \quad \text{for } j = 2, \dots, N-2; \\ \mathcal{D}_1 &= \left\{ z \in \mathbb{C} : \left| z - \frac{2}{h_1 h_2} \right| < \frac{2}{h_2(h_1 + h_2)} \right\}, \\ \mathcal{D}_{N-1} &= \left\{ z \in \mathbb{C} : \left| z - \frac{2}{h_{N-1} h_N} \right| < \frac{2}{h_N(h_{N-1} + h_N)} \right\}.\end{aligned}$$

ここですべての D_j を含むさらに大きい円

$$D = \left\{ z \in \mathbb{C} : \left| z - \frac{2}{h^2} \right| < \frac{2}{h^2} \right\}$$

の領域内に, A のすべての固有値 ξ が存在し, 以下の包含関係が成り立つ.

$$\xi \in \bigcup_j D_j \subset D. \quad (4.27)$$

ゆえに, 行列 A の固有値の実数部について,

$$0 < \Re(\xi) < \frac{4}{h^2}$$

が成り立ち, これによって, 行列 DA の固有値 λ はすべて,

$$\frac{-4\varepsilon}{h^2} < \Re(\lambda) < 0 \quad (4.28)$$

の条件を満たすことになる. また, 行列 DA は正則であるため, $\Re(\lambda) = 0$ にはならない.

ここで, 適用する数値積分法の絶対安定区間が $(-M, 0)$ であるとき, 言い換えると, $\Re(\lambda)\Delta t$ がこの区間内に存在するとき, 数値的安定性が得られるので, 式 (4.28) から得られる時間刻み幅 Δt についての絶対安定条件は,

$$\Delta t < \frac{Mh^2}{4\varepsilon} \quad (4.29)$$

となる. さらにこの条件式は, $M = \infty$ である A 安定な解法を適用する場合は, 刻み幅 Δt についての制限がなくなることも意味している.

以上の数値的収束性および安定性解析を以下のようにまとめる.

1. 本解法の収束の次数は, 適用する数値解法の次数 p と一致して $O(h^p)$ となる. ただし, 条件として, $\Delta t/\Delta x^2$ を一定に保つ必要がある.
2. 行列 DA のすべての固有値 λ は,

$$\frac{-4\varepsilon}{h^2} < \Re(\lambda) < 0$$

の有界の範囲に存在する.

3. 時間積分に陽的な数値積分法を適用する場合, 刻み幅は以下のように制限される.

$$\Delta t < \frac{Mh^2}{4\varepsilon}.$$

ここで, M は (陽的)Runge-Kutta 法の絶対安定区間の下限の絶対値である.

これらについては, 次節の数値実験において検証する.

4.3 方法および解析の数値的検証

ここでは、我々の離散化法およびその解析結果を検証するために、3.2節で述べたモデル問題に対して、一般的な数値積分法である古典的 Runge-Kutta 法を適用する。

4.3.1 古典的 Runge-Kutta 法の適用

移流拡散方程式 (3.1)–(3.3) のモデル問題 (3.4) を特性曲線法によって空間離散化した常微分方程式系 (4.2), (4.12) に対して古典的 Runge-Kutta 法を適用する。拡散係数 ε 、空間方向の分割の初期値 $\Delta x = 1/N$ 、時間方向の刻み幅 Δt を適当に変化させて数値実験し、方法の有効性、解析結果の妥当性を検証する。

1. まず、特性曲線の方程式 (4.2) のみを時刻 $t = t_M$ まで解く。数値実験のパラメタ $\Delta x, \Delta t$ を適当に変化させるとき、行列 $A = -\frac{1}{\varepsilon}DA$ の固有値の上限から得られる安定性条件 $4/h^2$ の振る舞いを調べ、数値的に計算する実際の A の固有値と比較する。
2. 次に、ここで得た安定性条件をもとに時間刻み幅 Δt を決定し、特性曲線に沿う方向の空間差分化にしたがって数値計算を進める。ここで得られる数値解が、安定なものであるかどうかを検証し、また、この解を得るのに必要な総ステップ数等を評価する。
3. 最後に、収束性、安定性の条件は満たすように数値解 U を求め、数値的収束性を確認するために、固定した時間刻み幅 Δt に対する誤差の振る舞いを評価する。

4.3.2 係数行列の固有値

表 4.1 は、行列 A の固有値について、 $t, \Delta t, \Delta x$ を変化させたときの Gershgorin の定理を用いて見積もった固有値の上限および実際の固有値のうち最大のもの、最小のものを示している。数値的な固有値は、富士通製数値計算ライブラリ SSL II を用いて計算した。

これによって、条件 (4.27)

$$0 < \Re(\xi) < \frac{4}{h^2}$$

が確かに満たされていることがわかり、この見積もりは粗い解析によるものであったが、適当な値であったことを示している。実際、ここで調査した $\Delta x, \Delta t$ の組についての結果では、見積もりの上限は最大値のおおよそ 2 倍程度になっている。

A の固有値について、さらにここでわかることは、すべてが実数値であることである。このことは 3 重対角行列においてすべての行の帯要素が同符号であることから示すことができる。また、特に t が増加する場合、 Δx を小さくすれば h は小となり、最大固有値と最小固有値の比が大きいことがわかる。このことについての考察および今後の課題は、4.3.5 節において述べる。

表 4.1: 行列 A の固有値の上限と数値的に求めた最大固有値, 最小固有値

t	Δt	Δx	$4/h^2$	max e-value	min e-value
1.0	1/4	1/25	2.944E+04	1.884E+04	9.833E+00
1.0	1/8	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/16	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/32	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/64	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/128	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/256	1/25	2.945E+04	1.884E+04	9.833E+00
0.5	1/16	1/25	8.471E+03	5.886E+03	9.849E+00
1.0	1/16	1/25	2.945E+04	1.884E+04	9.833E+00
1.5	1/16	1/25	1.032E+05	6.436E+04	9.828E+00
2.0	1/16	1/25	3.621E+05	2.243E+05	9.818E+00
2.5	1/16	1/25	1.272E+06	7.861E+05	9.790E+00
3.0	1/16	1/25	4.469E+06	2.760E+06	9.761E+00
1.0	1/16	1/25	2.945E+04	1.884E+04	9.833E+00
1.0	1/16	1/50	1.214E+05	1.007E+05	9.855E+00
1.0	1/16	1/100	4.926E+05	4.500E+05	9.866E+00
1.0	1/16	1/200	1.970E+06	1.889E+06	9.869E+00
1.0	1/16	1/400	7.896E+06	7.731E+06	9.869E+00

4.3.3 安定性条件を満たす刻み幅

空間方向の節点間隔の最小値 \underline{h} は時間とともに変化するため、安定条件を調整することはそれほど容易でない。ゆえに、ここでは以下のように時間刻み幅を決定することを試みる。ただし、この前提として、 \underline{h} は、特性曲線の方程式 (4.1) を数値的に解くことで得られるが、その精度は、表 4.1 の結果からわかるように、刻み幅 Δt の大きさにそれほど影響されないと考える。

1. まず、特性曲線方程式のみを $t = t_M$ まで、ある $\Delta x, \Delta t$ の組で数値的に解く。ただし、この計算は既に実行済みで、表 4.1 に結果が示されている。
2. 次に、 Δt を安定性条件 (4.29) から決定する。ここで、 ε については、 $t_M, \Delta x$ と組み合わせて数種の組を選定する。 \mathcal{M} については、ここで適用する古典的 Runge-Kutta 法の絶対安定領域に依存する値 $\mathcal{M} = 2.7$ を採用する。
3. ここで決定した Δt を用いて、方程式系 (4.11) と特性曲線の両方を数値的に解く。

表 4.2: 安定条件を満たす固定刻み幅に対する最大誤差

	no1	no2	no3	no4
ε	10^{-1}	10^{-3}	10^{-3}	10^{-3}
t_M	1.0	1.0	2.0	1.0
Δx	1/25	1/25	1/25	1/400
stiffness ratio	1.916E+3	1.916E+3	2.285E+4	7.834E+5
$4/\underline{h}^2$	2.945E+4	2.945E+4	3.621E+5	7.896E+6
sup Δt	9.168E-4	9.168E-2	7.456E-3	3.419E-4
Δt	2^{-11}	2^{-4}	2^{-8}	2^{-12}
total steps	2048	16	256	4096
max error	0.2089E-10	0.1166E-05	0.1312E-09	0.4574E-13

数値実験の結果を表 4.2 に示す。これらは、適当な $\varepsilon, t_M, \Delta x$ の 4 つの組による計算であり、得られた数値解はすべて十分に正確であることがわかる。no2 が基本となる組合せで、それぞれ、no1 は ε に、no3 は t_M に、no4 は Δx に依存する不安定要因を含んでいる。ゆえに、安定条件を満たす時間刻み幅 Δt による数値計算結果は安定であるといえる。しかし、あらかじめ特性曲線のみを適当な Δt で t_M まで解いて、そこで得られる \underline{h} に依存する安定条件から改めて Δt を決定し、主計算をするという今回の手順は実用的でない。また、定められた Δt は、 $0 < t < t_M$ の区間における最も厳しい安定条件に依るものであるから、非常に小さい値となり、この点についても実用的でないといえる。

4.3.4 収束性の数値的確認

表 4.3 および図 4.2 は，収束性条件

$$\frac{\Delta t}{\Delta x^2} = \text{const.}$$

と同時に安定性条件 (4.29) を満たすときの収束性を示している．この結果から以下のことがわかる．

- 時間方向の収束次数は $O(\Delta t^4)$ であり，適用する数値積分法である古典的 Runge-Kutta 法の収束性と一致する．
- 数値実験は倍精度実数にて行ったため， 10^{-12} 程度の高精度の実験において丸め誤差に依存する誤差が確認される．

表 4.3: $t = 1.0$ における最大絶対誤差

ε	Δt	Δx	$\max \text{error} $
1.0E-3	1/16	1/25	0.1166E-05
	1/64	1/50	0.3241E-08
	1/256	1/100	0.1215E-10
	1/1024	1/200	0.7061E-13
	1/4096	1/400	0.4530E-13
1.0E-6	1/16	1/25	0.1989E-06
	1/64	1/50	0.7587E-09
	1/256	1/100	0.2951E-11
	1/1024	1/200	0.1421E-12
	1/4096	1/400	0.2629E-12

以上より，収束性および安定性の条件を満たす場合，数値解は $\Delta t \rightarrow 0$ のとき収束し，その次数は適用する離散変数法に依存することが確認できた．このことは，今後の実験において，本章で適用した古典的 Runge-Kutta 法の代わりに，安定性のよりよい数値積分を適用する場合にも，その解法の収束の次数を達成できることを示唆している．

4.3.5 硬度比の動向

表 4.1 および表 4.2 の結果から，解くべき方程式系が特性曲線の振る舞いによっては，硬い系となる可能性があることがわかる．

具体的には， t が増加する場合，または，節点数 N の増加によって Δx が小さくなる場合， $h^2 \rightarrow 0$ となる．これにより，見積もった A の固有値の上限が急激に増加する．同様に，実

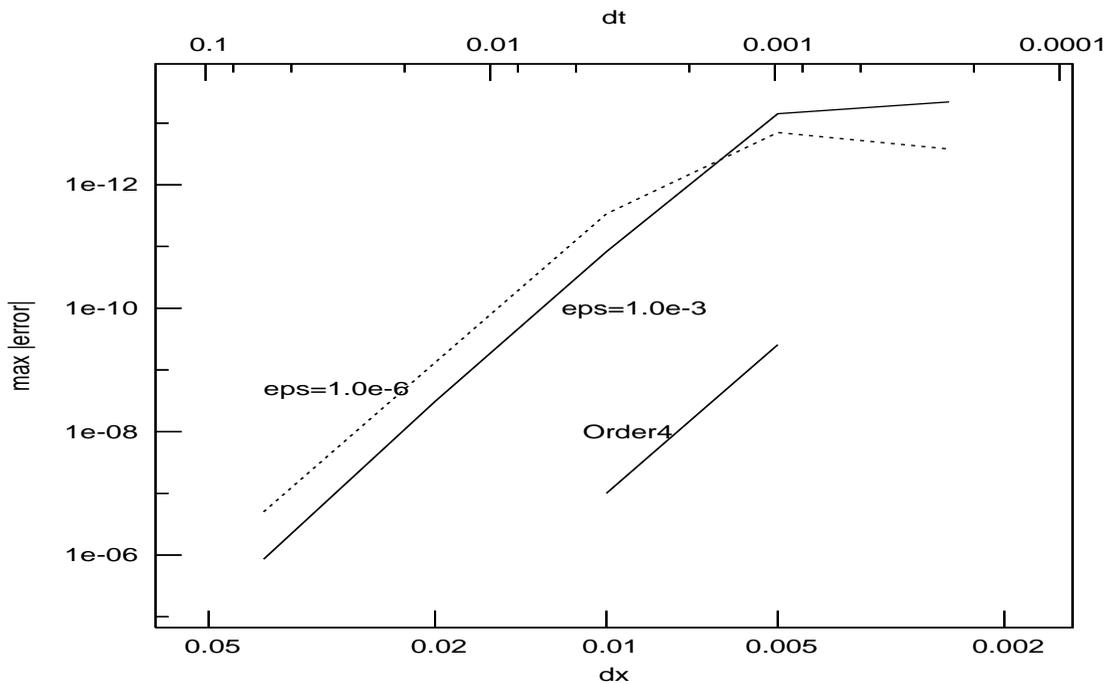


図 4.2: 数値的収束性の評価

際の固有値の最大値も増加している．それに対して，最小の固有値はほぼ一定の値になっている．ゆえに，最大固有値と最小固有値の比が大きいという硬い系であるという定義に当てはまる．

このような場合，硬い系に適用可能な数値解法を適用するのが望ましい．硬い系に適用するのに望ましい解法としては， A 安定性を達成する解法が挙げられる． A 安定性を達成する解法は，安定な解を得るための条件としての時間刻み幅の大きさの制限はなくなるが，1ステップあたりの計算量は増大する．この計算量の増大を軽減するのが並列化アルゴリズムである．本解法で解くべき方程式系に適用可能な並列化向きの離散変数法については，第5章，第6章で新たに開発する．

4.4 陽的解法による並列化実験と陰的解法の適用の展望

特性曲線に沿う空間差分を導入することで，解領域を分割する並列計算が可能となり，特に，共有メモリ型の並列計算機上で有効である．しかし，離散変数法として陽的な解法を用いる限りは，時間刻み幅が制限されて計算量が増大する．それに対して，陰的な解法を用いる場合は，1ステップあたりの計算量が増大することは既に述べたが，さらに，数値計算の相互依存性が増大するため並列性が損なわれる．本節では，今後の展望として陽的な数値積分を用いる場合の並列計算結果とその限界を示し，その後陰的解法を用いる並列化に向けての検討をする．

4.4.1 並列化実験

ここではまず、時刻ごとに安定条件を満たすように時間刻み幅を定めることを考える。つまり、ある時刻 t_n についての刻み幅が、

$$\Delta t_n < \frac{Mh^2}{4\varepsilon}$$

を満たすように、時刻ごとに空間節点間距離の最小値 h を計算し、得られた安定な Δt_n により、次の時刻の値を求めていく。適用する数値積分法として、一段解法である Runge-Kutta 法を用いることで、容易にこの刻み幅可変のアルゴリズムを構成できる。古典的 Runge-Kutta 法の場合、 $M = 2.7$ として計算する。

続いて並列化について述べる。ここでの並列化は、共有メモリ型での構成を考えているので、必要なデータは各プロセッサが共有しているとして容易にコーディングできる。つまり、ここで解くべき $2(N_x - 1)$ 次元常微分方程式系

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \frac{\mathbf{b}}{\mathbf{a}}(\mathbf{x}, t) \\ \frac{d\mathbf{U}}{dt} &= (DA)\mathbf{U} + \mathbf{F} = \mathbf{B}(\mathbf{x}, t)\mathbf{U} + \mathbf{F}(\mathbf{x}, t) \end{aligned}$$

の成分方程式の相互依存性が小さいことにより並列化が可能である。コーディング時に考慮すべきことは、ここでの計算領域のプロセッサへの割り当てをブロック分割として、隣接する方程式の計算が同時に複数のプロセッサで実行されることによる、メモリアクセス時のオーバーヘッドが生じないような工夫をすることである。

並列計算には、第2章で述べたとおり、共有メモリ型計算機の Power Challenge(4CPU) を用い、並列化言語 MPC(C for multi-processor) によって並列化コードを記述した。数値実験は、これまでと同様のテスト問題を $\varepsilon = 10^{-3}$ として $t = 1.0$ まで計算する。節点数 N_x を変化させて、逐次および並列計算の実行時間を計測し、速度向上度を評価する。また、最大絶対誤差を示し、安定条件に基づく刻み幅の再計算の有効性も検討する。

表 4.4: 共有メモリ計算機上での並列実験結果 ($\varepsilon = 10^{-3}$)

N_x	1CPU の実行時間 (HH:MM:SS.)	4CPU の実行時間 (HH:MM:SS.)	速度向上度	$\max \text{error} $ ($t=1.0$)
25	0.08	0.25	0.32	0.1166E-05
100	0.26	0.36	0.72	0.4099E-07
400	10.80	3.46	3.03	0.5587E-12
1600	10:46.84	3:06.17	3.47	0.1538E-11
6400	12:85:52.04	3:38:24.02	3.57	0.1695E-10

表 4.4 は、空間方向の節点数 N_x に対するステップごとに安定条件を計算するアルゴリズムの実行時間、並列計算の速度向上度と最大絶対誤差を示している。方程式の次元数が大きい

場合に速度向上度が增大する．また，最大絶対誤差は十分小さい値となっているので，安定に数値解を得られているといえる．しかし，実行時間は逐次計算の場合も並列計算の場合も同様に，節点数の増大に対して急激に増加している．ここで適用する並列化は，最大 $2(N_x - 1)$ 個のプロセッサで計算可能であるが，理想的に十分な数のプロセッサが用意されていてもこの計算量の増大の問題は解消されない．これは，十分安定な数値解を得るための Δt_n が非常に小さい値となることによるもので，陽的な数値解法を用いる限界であるといえる．次節では，A 安定な離散変数法である陰的 Runge-Kutta 法を適用するアルゴリズムを検討する．この場合，陽的解法適用時のような容易な並列化は困難となるが，並列化向きの数値積分法を用いることによる解法の高速度化が期待できる．

4.4.2 陰的 Runge-Kutta 法の適用

前節での問題点を踏まえて，安定な解を得るために陰的 Runge-Kutta 法を数値解 U の常微分方程式 (4.12)

$$U' = f(x, U, t) = B(x, t)U + F(x, t)$$

に適用する手順を考える．

式 (4.12) は，変数係数ではあるが線型の常微分方程式系であるので， s 段の陰的 Runge-Kutta 法

$$U_{n+1} = U_n + \Delta t \sum_{i=1}^s \beta_i k_i,$$

$$k_i = f \left(x(t_n + \gamma_i \Delta t), U_n + \Delta t \sum_{j=1}^s \alpha_{ij} k_j, t_n + \gamma_i \Delta t \right), \quad \text{for } i = 1, \dots, s$$

を適用するとき，下段の段値 (stage value) k の計算過程である非線型方程式は，

$$\begin{bmatrix} I - \Delta t \alpha_{11} B_1 & -\Delta t \alpha_{12} B_1 & \cdots & -\Delta t \alpha_{1s} B_1 \\ -\Delta t \alpha_{21} B_2 & I - \Delta t \alpha_{22} B_2 & \cdots & -\Delta t \alpha_{2s} B_2 \\ \vdots & & \ddots & \vdots \\ -\Delta t \alpha_{s1} B_s & -\Delta t \alpha_{s2} B_s & \cdots & I - \Delta t \alpha_{ss} B_s \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_s \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_s \end{bmatrix}$$

のように線型となる．これは，適当な線型方程式の数値解法を用いることで解くことができる．ただし，変数係数に対応する時刻 t 依存のパラメタは，

$$B_i = B(x(t_n + \gamma_i \Delta t), t_n + \gamma_i \Delta t),$$

$$f_i = f(x(t_n + \gamma_i \Delta t), U, t_n + \gamma_i \Delta t)$$

であり，さらに $x(t_n + \gamma_i \Delta t)$ は特性曲線の方程式 (4.2) を解くことで求めなければならない．ここでは，特性曲線の方程式は不安定要因を含まないものとし，スケールつき Runge-Kutta 法 ([30], pp.86-90) を用いることで， $0 < \gamma_i < 1$ に対応する $t_n + \gamma_i \Delta t$ における数値解 $X_i \simeq x(t_n + \gamma_i \Delta t)$ を求め，これを以後の計算に利用することになる．

つまり、ここでの手順は、まず、特性曲線を解くのに7段4次のスケールつき Runge-Kutta 法 (Fehlberg 公式に基づく) を適用して必要な段値 X_i ($i = 1, 2$) を計算する。そののちに U の常微分方程式には、 A 安定な陰的 Runge-Kutta 法である2段4次 Butcher-Kuntzmann 公式 (BK24) を適用することになる。また、各段で解くべき線型方程式系は直接法 (Gauss 消去法) を適用する。時間刻み幅 Δt は Fehlberg 公式から自動調節可能であるが、ここでは単純に固定値を用いる。

表 4.5 は、4.3.3 節においての実験 (RK4-fixed) と同じ $\varepsilon, t_M, \Delta x$ の組に対する前小節 (RK4-scalable) および本小節 (BK24) のアルゴリズムによる数値実験の結果である。それぞれの解法について、時間刻み幅、総ステップ数、最大絶対誤差と実行時間を示している。ただし、計算環境は通常のワークステーション (Sun 製 UltraSparcII) を用い、並列計算は適用していない。

表 4.5: BK24 適用時の最大誤差と実行時間 (MM:SS.)

	no1	no2	no3	no4
ε	10^{-1}	10^{-3}	10^{-3}	10^{-3}
t_M	1.0	1.0	2.0	1.0
Δx	1/25	1/25	1/25	1/400
stiffness ratio	1.916E+3	1.916E+3	2.285E+4	7.834E+5
RK4-fixed				
sup Δt	9.168E-4	9.168E-2	7.45575E-3	3.41937E-4
Δt	2^{-11}	2^{-4}	2^{-8}	2^{-12}
total steps	2048	16	256	4096
max error	0.2089E-10	0.1166E-05	0.1312E-09	0.4574E-13
CPU time	00:01.67	00:00.02	00:00.44	00:59.26
RK4-scalable				
Δt_0	2^{-4}	2^{-4}	2^{-4}	2^{-4}
Δt_M	8.634E-4	6.250E-2	1.897E-3	3.252E-4
total steps	427	16	68	1074
max error	0.3871E-09	0.1166E-05	0.2001E-06	0.5587E-12
CPU time	00:00.34	00:00.02	00:00.06	00:15.09
BK24				
Δt	2^{-4}	2^{-4}	2^{-4}	2^{-4}
total steps	16	16	16	16
max error	0.5745E-04	0.7002E-06	0.4607E-06	0.7757E-06
CPU time	00:00.12	00:00.12	00:00.23	05:51.65

no1,no3,no4 の組は問題が硬い系であるが、それぞれ安定な数値解が得られていることがわかる。硬い系に対しては、前節で述べたとおり陽的解法の場合、安定な刻み幅を1ステップ

ブ毎に決定できるため、従来の固定刻み幅の方法よりは実用的で計算速度も向上している。しかし、安定な解を得るためには時間刻み幅 Δt_n を十分小さい値にする必要があるため、結果として計算時間が増大する。今回用いた問題の組は、極端に厳しい安定条件ではないので、計算時間を極端に要することはなかったが、計算時間が安定条件に依存することは陽的解法を用いる限界であるといえる。

これに対して、陰的解法を適用する場合は、適当な固定刻み幅による数値解が安定で、安定性のために刻み幅を変える必要がない。今回の結果ではすべて刻み幅 $\Delta t = 2^{-4}$ で固定し、残りの2つの陽的解法のように極端に小さい値でなくても安定な数値解が得られている。しかし、解くべき方程式系の次元数が増大する no4 の場合、最終的に解くことになる線型方程式の解法の計算量 ($O(N_x^3)$) に依存して計算量が増大する。

つまり、A 安定な解法として陰的 Runge-Kutta 法を採用する場合、安定性に依存するステップ数の増大はないが、問題のサイズが大きくなるようなときには計算量が増大するという新たな問題が生じる。

4.5 本章のまとめ

本章では、移流拡散方程式の特性曲線差分法を提案し、数値的収束性および安定性を解析するとともに、数値実験によりその妥当性や問題点を検討した。その要約は以下のとおりである。

1. 一般的な等間隔、固定点の節点上ではなく、特性曲線 (4.1) に沿う方向に節点を移動し、そこで構成される格子上に数値解を求めていく。
2. 問題を特性曲線 (4.2) を解く部分と、差分化した常微分方程式 (4.12) を解く部分に分離し、それらを同時に解く。
3. Gershgorin の定理による解析は、粗い解析ではあるが、今回取り扱う問題については、この固有値の評価方法は有効である。

4. 安定性条件

$$\Delta t < \frac{Mh^2}{4\varepsilon}$$

を満足するとき、数値解は指数函数的な誤差の増大を抑えることができる。

5. やや厳しい条件であるが、

$$\frac{\Delta t}{\Delta x^2} = \text{const.}$$

を満たすとき数値解は収束し、その次数は適用する数値積分法の次数と一致する。

6. 特性曲線上に構成された不等間隔の節点における中心差分に依存して、特性曲線法によって得られた常微分方程式系は硬い系となる可能性がある。

7. 陽的な Runge-Kutta 法を適用する場合，一段解法であることの利点による安定条件をステップごとに決定するアルゴリズムやそれぞれの方程式の相互依存性が小さいことによる並列化が容易であるが，問題が硬い系となる場合に計算量が増大するため，陽的な数値積分法の適用は実用的でない．
8. 特性曲線法によって空間差分した常微分方程式系は， A 安定な陰的 Runge-Kutta 法を適用するとき変数係数常微分方程式系となり，段値の非線型方程式は，線型方程式に置き換えられるが，段値に対応する特性曲線方程式の解をスケールつき Runge-Kutta 法等で計算する必要がある．これによって，安定な数値解を得ることが期待できる．
9. A 安定の陰的解法によって，安定な数値解を得ることができるが，反面，1 ステップでの計算量が増大するため，なんらかの高速化をはかる必要がある．

以上より，次章以降では，変数係数線型常微分方程式系

$$\mathbf{y}' = \mathbf{L}(t)\mathbf{y} + \mathbf{F}(t)$$

に焦点を移し，この系に適用可能な並列化向き離散変数法を新たに開発する．

第5章 変数係数線型系に対する並列化向き ROW 型解法

第4章において，常微分方程式系の初期値問題

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0 \quad (5.1)$$

が式 (4.12) のように (d 次元) 線型の常微分方程式系で，その係数行列が変数係数であり，さらに大連立でかつ硬い系であるようなものに対して，安定でかつ高速な数値積分法を新たに構成する必要があった．新たに構成する数値解法については，3.4 節で述べたような収束性および安定性の解析が必要である．さらにその上，ここで取り扱う変数係数線型系の安定性の議論は絶対安定から NA 安定に焦点を移す必要がある．本章では，論文 [10] に基づいて，高速化を目的とし，取り扱う系に特化して適用する数値積分法として並列化向きの ROW 型スキームを構成する．この解法は，O.A. KARAKASHIAN らの並列化に倣って，変数係数線型系に適用するために，陰的 Runge-Kutta 法をさらに近似するスキームである．また，構成したスキームの収束性，安定性を解析し，得られた結果をテスト問題の数値実験により確認をする．さらに並列計算機上で実験を行い，その並列化の効果を従来の逐次計算機向きの解法と比較することで検証する．

5.1 陰的 Runge-Kutta 法による試み

常微分方程式系 (5.1) を陰的 Runge-Kutta 法を用いて数値的に解く場合， s 段公式を適用するスキームは以下のように表わされる．

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s \beta_i \mathbf{k}_i, \quad (5.2)$$

$$\mathbf{k}_i = \mathbf{f} \left(t_n + \gamma_i h, \mathbf{y}_n + h \sum_{j=1}^s \alpha_{ij} \mathbf{k}_j \right), \quad \text{for } i = 1, \dots, s. \quad (5.3)$$

ここで， h は時間方向の刻み幅とし， \mathbf{y}_j は時刻 $t_j = t_0 + j \cdot h$ における数値解を意味する．この公式は Butcher 配列を用いて

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

と記述する．ここで， $s \times s$ 次元行列 $A = (\alpha_{ij})$ であり， $b = (\beta_i)$ ， $c = (\gamma_i)$ は s 次元ベクトルを表す．

陰的 Runge-Kutta 法は硬い系に適用可能な方法として知られるが，大きな欠点として，段値 (stage-value) k の方程式は， $d \times s$ 次元の非線形方程式であり，何らかの反復解法を用いて解く必要があるため，1 ステップでの計算時間は増大する．

この欠点を改善するのに種々の並列計算アルゴリズムが考案されてきた．熱方程式の差分法に現れるような定数係数線型系

$$y' = Ly$$

に適用する並列化向き陰的 Runge-Kutta 法として，O.A. KARAKASHIAN らの 2 段 3 次公式 [27] や後藤らの 3 段 4 次公式 [18] がある．これらは，この特定の問題に対してのみ適用可能で，非線形方程式 (5.3) を Kronecker 積を用いて，

$$(I - hA \otimes L)K = \Phi$$

のように記述できる場合に並列化可能な解法である．これらは， A 安定を達成する公式が構成できるのが利点である．しかし，我々の要求である変数係数線型系の場合は，ここでの並列化の着想はもはや利用できない．ゆえに，従来の離散解法をさらに近似する新たな解法を構成することになる．この解法の並列化は，ここで紹介した並列化向き陰的 Runge-Kutta 法のそれに倣ったものである．

5.2 ROW 型の近似公式

常微分方程式 (5.1) が変数係数線型系であるとき，

$$y' = L(t)y + F(t), \quad y(0) = y_0, \quad (5.4)$$

陰的 Runge-Kutta 法の段値方程式 (5.3) は，大規模疎行列の線型方程式系となり，以下の行列形式で与えられる．

$$\begin{aligned} & \begin{bmatrix} I - h\alpha_{11}L(t_n + \gamma_1h) & -h\alpha_{12}L(t_n + \gamma_1h) & \cdots & -h\alpha_{1s}L(t_n + \gamma_1h) \\ -h\alpha_{21}L(t_n + \gamma_2h) & I - h\alpha_{22}L(t_n + \gamma_2h) & \cdots & -h\alpha_{2s}L(t_n + \gamma_2h) \\ \vdots & & \ddots & \vdots \\ -h\alpha_{s1}L(t_n + \gamma_sh) & -h\alpha_{s2}L(t_n + \gamma_sh) & \cdots & I - h\alpha_{ss}L(t_n + \gamma_sh) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_s \end{bmatrix} \\ &= \begin{bmatrix} L(t_n + \gamma_1h)y_n + F(t_n + \gamma_1h) \\ L(t_n + \gamma_2h)y_n + F(t_n + \gamma_2h) \\ \vdots \\ L(t_n + \gamma_sh)y_n + F(t_n + \gamma_sh) \end{bmatrix}. \end{aligned} \quad (5.5)$$

すべての $\{L(t_n + \gamma_ih)\}$ (for $i = 1, 2, \dots, s$) の値が異なるので，KARAKASHIAN らの行列 A の対角化による並列化技法 [27] はもはや適用できない．そこで我々は，行列内に現れる d 次

線型系の変数係数行列が、各時刻においてすべて同じものになるように近似して置き換える。つまり、パラメタ C を用いて、式 (5.5) の左辺に現れるすべての $L(t_n + \gamma_i h)$ を同時に、 $L(t_n + Ch)$ のように置き換える近似である。そしてここでは、 $L(t_n + Ch)$ を L_C のように書く。これによって、式 (5.5) は以下のように変換される。

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} - h\alpha_{11}L_C & -h\alpha_{12}L_C & \cdots & -h\alpha_{1s}L_C \\ -h\alpha_{21}L_C & \mathbf{I} - h\alpha_{22}L_C & \cdots & -h\alpha_{2s}L_C \\ \vdots & & \ddots & \vdots \\ -h\alpha_{s1}L_C & -h\alpha_{s2}L_C & \cdots & \mathbf{I} - h\alpha_{ss}L_C \end{bmatrix} \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} \\ &= \begin{bmatrix} L(t_n + \gamma_1 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_1 h) \\ L(t_n + \gamma_2 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_2 h) \\ \vdots \\ L(t_n + \gamma_s h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_s h) \end{bmatrix}. \end{aligned} \quad (5.6)$$

そしてここで、Kronecker 積を導入すると、式 (5.6) は以下のように簡単に表すことができる。

$$(\mathbf{I} - h\mathbf{A} \otimes L_C)\mathbf{K} = \Phi. \quad (5.7)$$

ただし、 $(d \times s)$ 次元ベクトル \mathbf{K} 、 Φ は、

$$\mathbf{K} = \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{bmatrix}, \quad \Phi = \begin{bmatrix} L(t_n + \gamma_1 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_1 h) \\ L(t_n + \gamma_2 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_2 h) \\ \vdots \\ L(t_n + \gamma_s h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_s h) \end{bmatrix} \quad (5.8)$$

のように定義する。この近似による線型方程式 (5.7) においては、離散化パラメタ行列 \mathbf{A} の対角化による並列化が可能となる。並列化の詳細は 5.4 節で述べる。

この我々の新しい離散変数法 (5.2), (5.6) を陰的 ROW 型 (ROW-type, Rosenbrock-Wanner-type) 解法と呼ぶ。この解法において、まず、式 (5.6) によって段値が得られる。そして次に、式 (5.2) によって次のステップでの数値解を計算し、逐次的に計算を進めていく。この方法を構成する規則は以下のとおりである。

- 自由パラメタを用いて、公式の次数条件が可能な限り高次になるようにスキームを決定する。
- 自由パラメタは、行列 \mathbf{A} を行列の相似変換によって対角化可能とするためにも利用される。
- 決定したスキームが適当な数値的安定性を達成しているかを確認する。

次節においては、もっとも最小の構成である 1 段 2 次公式、および、最高到達次数を達成する最小の構成 2 段 3 次公式を、上記の規則に従って代数方程式を解くことにより決定する。

5.3 3次公式のパラメタ決定

我々のスキームに現れる新しいパラメタ C を Runge-Kutta 法の Butcher 配列に

$$\begin{array}{c|c} c & A \\ \hline C & b^T \end{array}$$

のように付け加えることとする。ただし、 C はスカラー値である。

スキームの次数条件を達成するために、真の解 $\mathbf{y}(t_{n+1})$ と数値解 \mathbf{y}_{n+1} の両方を Taylor 展開し、 $\mathbf{y}_n = \mathbf{y}(t_n)$ の仮定のもとでその係数を比較する。次の時刻 $(t_n + h)$ における真の解は以下のように展開される。

$$\mathbf{y}(t_n + h) = \mathbf{y}(t_n) + h\mathbf{y}'(t_n) + \frac{h^2}{2}\mathbf{y}''(t_n) + \frac{h^3}{6}\mathbf{y}'''(t_n) + \frac{h^4}{24}\mathbf{y}^{(4)}(t_n) + O(h^5). \quad (5.9)$$

以下においては簡単のため、 t_n における \mathbf{y}, L, F の評価をそれぞれ、 $\mathbf{y}(t_n) = \mathbf{y}, L(t_n) = L, F(t_n) = F$ という様に記述する。

ここで、4階までの導関数は適用する常微分方程式の線型性により次のように書き下せる。

$$\begin{aligned} \mathbf{y}'(t_n) &= L\mathbf{y} + F, \\ \mathbf{y}''(t_n) &= L(L\mathbf{y} + F) + L'\mathbf{y} + F', \\ \mathbf{y}'''(t_n) &= L^2(L\mathbf{y} + F) + 2L'(L\mathbf{y} + F) \\ &\quad + L(L'\mathbf{y} + F') + L''\mathbf{y} + F'', \\ \mathbf{y}^{(4)}(t_n) &= L^3(L\mathbf{y} + F) + 5LL'(L\mathbf{y} + F) \\ &\quad + 3L''(L\mathbf{y} + F) + L^2(L'\mathbf{y} + F') \\ &\quad + 3L'(L'\mathbf{y} + F') + L(L''\mathbf{y} + F'') \\ &\quad + L'''\mathbf{y} + F'''. \end{aligned}$$

同様に、数値解 \mathbf{y}_{n+1} は、

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + h \sum_{i=1}^s \beta_i (L\mathbf{y} + F) \\ &\quad + h^2 \left\{ \sum_{i=1}^s \beta_i \gamma_i (L'\mathbf{y} + F') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} L(L\mathbf{y} + F) \right\} \\ &\quad + h^3 \left\{ \frac{1}{2} \sum_{i=1}^s \beta_i \gamma_i^2 (L''\mathbf{y} + F'') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j L(L'\mathbf{y} + F') \right. \\ &\quad \left. + C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} L'(L\mathbf{y} + F) + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} L^2(L\mathbf{y} + F) \right\} \end{aligned} \quad (5.10)$$

$$\begin{aligned}
& +h^4 \left\{ \frac{1}{6} \sum_{i=1}^s \beta_i \gamma_i^3 (L'''y + F''') + \frac{1}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j^2 L(L''y + F'') \right. \\
& + C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j L'(L'y + F') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \gamma_l L^2(L'y + F') \\
& + \frac{C^2}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} L''(Ly + F) + 2C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} LL'(Ly + F) \\
& \left. + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \sum_{m=1}^s \alpha_{lm} L^3(Ly + F) \right\} + O(h^5) \quad (5.11)
\end{aligned}$$

のように展開される．式(5.9)と式(5.11)の同じ次数の h の項について，係数を比較することで，公式の次数条件を得ることができる．

5.3.1 次数条件式

2次のスキームを構成するための次数条件式は以下のとおりである．

$$\sum_{i=1}^s \beta_i = 1 \quad (5.12)$$

$$\sum_{i=1}^s \beta_i \gamma_i = \frac{1}{2} \quad (5.13)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} = \frac{1}{2} \quad (5.14)$$

3次公式のための条件は式(5.12)–(5.14)に

$$\frac{1}{2} \sum_{i=1}^s \beta_i \gamma_i^2 = \frac{1}{6} \quad (5.15)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j = \frac{1}{6} \quad (5.16)$$

$$C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} = \frac{1}{3} \quad (5.17)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} = \frac{1}{6} \quad (5.18)$$

を加えたものである．同様に，4次公式のための条件は式(5.12)–(5.18)に以下のものに加えられたものである．

$$\frac{1}{6} \sum_{i=1}^s \beta_i \gamma_i^3 = \frac{1}{24} \quad (5.19)$$

$$\frac{1}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j^2 = \frac{1}{24} \quad (5.20)$$

$$C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j = \frac{3}{24} \quad (5.21)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \gamma_l = \frac{1}{24} \quad (5.22)$$

$$\frac{C^2}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} = \frac{3}{24} \quad (5.23)$$

$$2C \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} = \frac{5}{24} \quad (5.24)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \sum_{m=1}^s \alpha_{lm} = \frac{1}{24} \quad (5.25)$$

まずここでわかるのが，式 (5.14) および (5.17), (5.23) を同時に満たす解 C が存在しないので，4次を達成する公式が構成できないことである．ゆえに，ROW 型解法の到達可能次数は3次であるといえる．以上をまとめると，式の数と未知数の数から，表 5.1 のように，段数と到達可能次数 $p^*(s)$ の必要条件がわかるので，これに基づいて1段2次および2段3次の公式を構成する．

表 5.1: 段数と到達可能次数

段数 s	1	2	3 以上
$p^*(s)$	2	3	3

5.3.2 2次および3次の公式

1段2次公式のパラメタはほとんど一意に決定される．式 (5.12)–(5.14) を満たすその値は以下のとおりである．

$$\beta_1 = 1, \quad \gamma_1 = \frac{1}{2}, \quad \alpha_{11} = \frac{1}{2}, \quad C = \text{任意}.$$

これらのパラメタは，もっとも単純な陰的 Runge-Kutta 公式である陰的 midpoint のそれとほぼ同様のものである． $0 \leq C \leq 1$ について試みた数値実験の結果， $C = \frac{1}{2}$ とするのが最も適当

と判断しこれを採用する．また，計算量の観点からも $C = \gamma_1$ とするのが有利といえる．ゆえに我々は，1段2次解法を以下のように表現する．

$$\frac{c}{C} \left| \begin{array}{c|c} \mathbf{A} & \\ \hline \mathbf{b}^T & \end{array} \right. = \frac{\frac{1}{2}}{\frac{1}{2}} \left| \begin{array}{c|c} & \frac{1}{2} \\ \hline \frac{1}{2} & \end{array} \right. .$$

次に，2段3次公式を考える．式(5.14)と式(5.17)から容易に $C = \frac{2}{3}$ が得られる．式(5.12), (5.13), (5.15)において， β_1 と β_2 の組が解をもつためには， γ_1 と γ_2 についての以下の条件が必要である．

$$(\gamma_1 - \gamma_2) \left(\gamma_1 \gamma_2 - \frac{1}{2}(\gamma_1 + \gamma_2) + \frac{1}{3} \right) = 0. \quad (5.26)$$

この条件のもとで，解 (β_i) は以下のように表わされる．

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \frac{1}{\gamma_2 - \gamma_1} \begin{bmatrix} \gamma_2 - \frac{1}{2} \\ \frac{1}{2} - \gamma_1 \end{bmatrix}. \quad (5.27)$$

ここで，自然な仮定として， $\gamma_1 = \alpha_{11} + \alpha_{12}$, $\gamma_2 = \alpha_{21} + \alpha_{22}$ となるとすると，それによって，式(5.14)および式(5.18)は，それぞれ，式(5.13)および式(5.16)にまとめられる．以上から得られる解 (α_{ij}) についての条件式は，

$$\begin{aligned} \alpha_{11} + \alpha_{12} &= \gamma_1, \\ \alpha_{21} + \alpha_{22} &= \gamma_2, \\ \beta_1(\alpha_{11}\gamma_1 + \alpha_{12}\gamma_2) + \beta_2(\alpha_{21}\gamma_1 + \alpha_{22}\gamma_2) &= \frac{1}{6} \end{aligned}$$

のようになる．ここで， α_{11} を自由パラメタとして，残りの α_{ij} は以下のように解くことができる．

$$\begin{cases} \alpha_{22} = \frac{-1 + 6\gamma_1\gamma_2 + 6\alpha_{11}\beta_1(\gamma_1 - \gamma_2)}{6\beta_2(\gamma_1 - \gamma_2)} \\ \alpha_{12} = \gamma_1 - \alpha_{11} \\ \alpha_{21} = \gamma_2 - \alpha_{22} \end{cases}. \quad (5.28)$$

以上をまとめると γ_1 と α_{11} を自由パラメタとして残して，その他のパラメタは代数方程式の解として式(5.26)および式(5.27), 式(5.28)から決定することができる．自由パラメタ γ_1 および α_{11} は，次節において，安定性や対角化可能性の条件として決定する．

5.3.3 対角化可能および NA 安定性のための条件

ここでは，2段3次公式の残りの自由パラメタに対しての，対角化可能性と NA 安定性 (non-autonomous stability)[11] のための条件を示す．ただし，安定性の条件については，一般的なパラメタに対する条件を示すのは難しいので，適当に定めた Butcher 配列の値から，

そのスキームが安定性を満たすのかどうかを検討する．本小節では安定性のための条件のみを示し，パラメタの決定および安定性の検討は，次の 5.3.4 節において述べる．

まず，我々の ROW 型解法は，一段解法であるから， γ_1, γ_2 については，条件

$$0 \leq \gamma_1, \gamma_2 \leq 1$$

を満たすのが望ましい．

次に，行列 A を相似変換によって， $A = S^{-1}\Lambda S$ と対角化することを考える．実数係数の行列に対しての対角化可能の十分条件は，すべての固有値が互いに異なる値であることである．これにより，2 段公式の係数行列 A の場合，条件式は以下ようになる．

$$(\alpha_{11} - \alpha_{22})^2 + 4\alpha_{12}\alpha_{21} > 0. \quad (5.29)$$

さらにここで，数値解法の NA 安定性について考える．我々の ROW 型解法は変数係数のために構成されたスキームであるから，定数係数線型系をテスト方程式とする従来の A 安定性解析では十分でない．その代わりに用いるのが NA 安定性解析である．5.5.2 節で述べる NA 安定性解析の結果，NA 安定性因子 $R(z)$ を導入し，そこで条件付き NA 安定性という NA 安定性に関する新たな指標を提案した．実際，構成したスキームの NA 安定性因子は，

$$R(z) = \frac{\det(\mathbf{I} - z_0\mathbf{A} + b\bar{z}^T)}{\det(\mathbf{I} - z_0\mathbf{A})} \quad (5.30)$$

のように表わされ， $z = (z_0, z_1, \dots, z_s)^T \in \mathbb{C}^{s+1}$ は複素変数のベクトルであり， $\bar{z} = (z_1, \dots, z_s)^T$ はその部分ベクトルである．ここで，無条件 NA 安定とは，複素領域内の z について，それらすべての実部が $\Re z_i < 0$ ($i = 0, 1, \dots, s$) となる領域において，安定性因子 $R(z)$ が $|R(z)| < 1$ を満たすことで，一般的には AN 安定と呼ばれる．この条件は非常に厳しく，これを達成する公式で知られているものには，陰的 Runge-Kutta 法のうち Butcher-Kunzmann 公式がある．我々は，ある条件のもとで，NA 安定を達成することを条件付き NA 安定と呼び，我々が構成したスキームについてはこの指標について，NA 安定性を解析する．

1 段 2 次公式の NA 安定性因子は， $C = \gamma_1$ とするとき， $z_0 = z_1$ となるため，

$$R(z) = \frac{1 + \frac{z_0}{2}}{1 - \frac{z_0}{2}} \quad (5.31)$$

と表わされる．これは， e^{z_0} の Padé 近似有理函数にほかならず，これによりこの公式は，無条件に NA 安定であることを意味する．

5.3.4 パラメタの決定

2 段公式の場合，相似変換を容易にするために，自由パラメタのひとつを $\gamma_2 = 1$ とおく．これにより， (γ_i) および (β_i) は一意に決する．次に， $\alpha_{11} = \frac{2}{3}$ とおいて，式 (5.28) から残りのパラメタ (α_{ij}) を決定する．以上より，公式のすべてのパラメタを以下のように決定する．

$$\begin{array}{c|cc} \frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ \hline \frac{1}{3} & 0 & \frac{1}{3} \\ \hline \frac{2}{3} & \frac{3}{4} & \frac{1}{4} \end{array}.$$

ここで決定したパラメタについて，5.3.3 節で述べた対角化可能と NA 安定性の条件を満たすかどうかを確認する．まず，式 (5.29) を満たすので，行列 A は以下のように対角化可能である．

$$\begin{aligned} A &= S^{-1}\Lambda S \\ &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (5.32)$$

次に， NA 安定性について考える．詳しい解析は，5.5.2 節で述べるが，ここで構成した 2 段 3 次公式は，無条件には NA 安定ではない．しかし，時間刻み幅 h が十分小さい範囲で NA 安定となる，条件付き NA 安定であるといえる．

これまでに決定した 1 段 2 次公式および 2 段 3 次公式を採用して，以下の 5.6 節においてテスト問題に対する数値実験を行う．

5.4 ROW 型解法の並列化

ここで，我々の解法の並列化手法について述べる．前出の式 (5.7)

$$(I - hA \otimes L_C)K = \Phi$$

において，行列 A が以下のように対角化可能であると仮定する．

$$A = S^{-1}\Lambda S, \quad \Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_s \end{bmatrix}.$$

これを用いると，式 (5.7) において左辺は以下のように変形できる．

$$\begin{aligned} (I - h(S^{-1}\Lambda S) \otimes L_C)K &= \Phi \\ (I - h(S^{-1} \otimes I)(\Lambda \otimes L_C)(S \otimes I))K &= \Phi. \end{aligned} \quad (5.33)$$

ここでさらに， $(S \otimes I)$ を左から掛けると，

$$(I - h(\Lambda \otimes L_C))(S \otimes I)K = (S \otimes I)\Phi \quad (5.34)$$

のように変形できる．ここで，以下のようにベクトルを定義すると，

$$\begin{aligned} (\mathbf{S} \otimes \mathbf{I})\mathbf{K} &\equiv \mathbf{U}, \quad \mathbf{U} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_s^T]^T, \\ (\mathbf{S} \otimes \mathbf{I})\Phi &\equiv \mathbf{V}, \quad \mathbf{V} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_s^T]^T, \end{aligned}$$

式 (5.7) の変形は，

$$(\mathbf{I} - h(\Lambda \otimes \mathbf{L}_C))\mathbf{U} = \mathbf{V}, \quad (5.35)$$

のように得られる．ここで，この式を行列表記に戻すと，

$$\begin{bmatrix} \mathbf{I} - h\lambda_1 \mathbf{L}_C & & & 0 \\ & \mathbf{I} - h\lambda_2 \mathbf{L}_C & & \\ & & \ddots & \\ 0 & & & \mathbf{I} - h\lambda_s \mathbf{L}_C \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_s \end{bmatrix} \quad (5.36)$$

のようにブロック対角要素のみの方程式であることがわかる．ゆえに，実際に計算すべき線型方程式は，

$$(\mathbf{I} - h\lambda_i \mathbf{L}_C)\mathbf{u}_i = \mathbf{v}_i, \quad \text{for } i = 1, 2, \dots, s \quad (5.37)$$

であり，各 i 番目の計算は独立であるため， s 段の解法については， s 個のプロセッサを用いて並列計算可能である．しかし，時間発展の逐次計算には，

$$\mathbf{K} = (\mathbf{S}^{-1} \otimes \mathbf{I})\mathbf{U}$$

の計算が必要で，

$$\mathbf{k}_i = \sum_{j=1}^s (\mathbf{S}^{-1})_{ij} \mathbf{u}_j, \quad \text{for } i = 1, 2, \dots, s$$

のように分散メモリ上の数値データを集約する必要がある．

並列計算アルゴリズムにおける命令とデータの流れを図 5.1 に示す．ここでは主に対象としている 2 段解法についての，分散メモリ型並列計算機上での並列化を述べるが，2 個の CPU を 2 個の d 次元線型方程式 (5.37) それぞれを解くのに用いる．それぞれの線型方程式を解いた後に，2 番目のプロセッサ (CPU2) は，方程式の解 \mathbf{u}_2 を 1 番目の CPU (CPU1) に渡す必要がある．CPU1 は次の時間での値 \mathbf{y}_{n+1} を計算し，その値を次の時刻 t_{n+1} での計算のために CPU2 に送り返す．また，各時間においての線型方程式は，式 (5.4) の時間依存性のために，

$$\mathbf{G}_i \equiv (\mathbf{I} - h_i \lambda_i \mathbf{L}_C)$$

を各時間ごとに再計算しなければならない．

5.5 数値スキームの収束性と安定性

本章で構成した離散変数法についても，新たな近似については注意深い収束性および安定性の解析が必要である．収束性については従来の方法と同様に，解くべき方程式系が Lipschitz

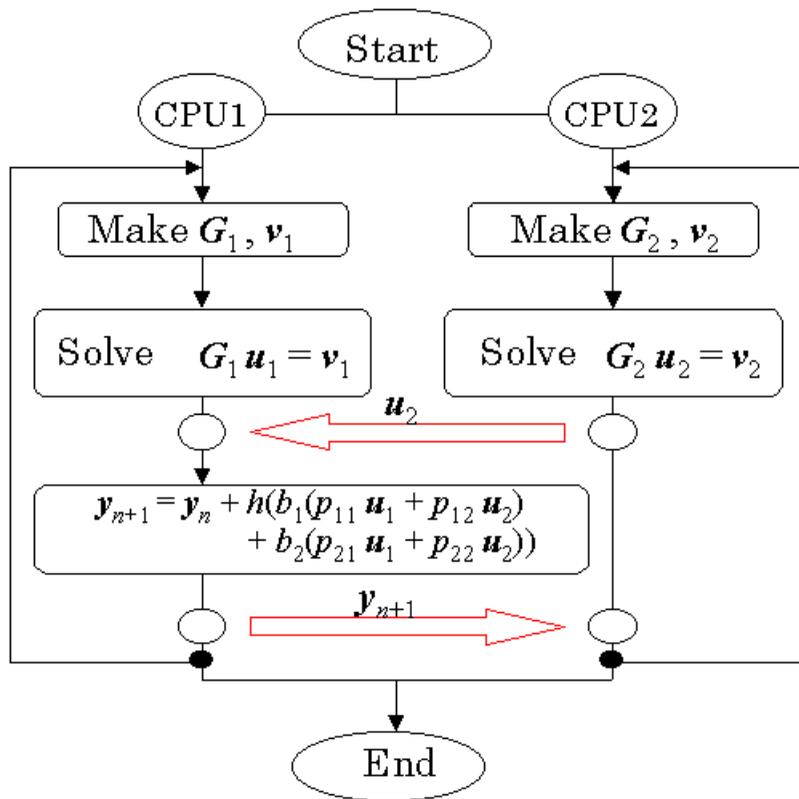


図 5.1: 命令とデータの流れ図

連続性を保つとするときの, $h \downarrow 0$ のときの大域誤差の振る舞いを解析する. 安定性については, NA 安定性を解析し, 無条件 NA 安定を達成できないが, 条件付きでスキームが NA 安定であることを示す.

5.5.1 収束性解析

ここで, ROW 型解法の大域的な収束性を解析する. まず, 一般的な常微分方程式系 (5.1) の右辺の函数 f に適用する一段解法の離散変数法を

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\Phi(t_n, \mathbf{y}_n; h) \quad (5.38)$$

と表す. ここで写像 Φ はすべての k_i を含む計算過程を意味し, 解くべき方程式 f に依存する. ここで, 我々が適用する ROW 型解法の局所離散化誤差は,

$$\mathbf{T}_{n+1} = \frac{1}{h} \{ \mathbf{y}(t_{n+1}) - \mathbf{y}(t_n) - h\Phi(t_n, \mathbf{y}(t_n); h) \}$$

であり, これを真の解についての式に書き直すと,

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + h\Phi(t_n, \mathbf{y}(t_n); h) + h\mathbf{T}_{n+1} \quad (5.39)$$

となるから, p 次公式ならば以下のように記述できる.

$$\mathbf{T}_{n+1} = \mathbf{A}(t_n, \mathbf{y}(t_n))h^p + O(h^{p+1}). \quad (5.40)$$

ただし, 係数 $\mathbf{A}(t_n, \mathbf{y}(t_n))$ は f の高階偏導函数の複雑な組合せとなる. ここで, f が p 回偏微分可能で, ある正値 h_0 が $h < h_0$ であるとき, 不等式

$$\|\mathbf{T}_{n+1}\| \leq \bar{A}h^p \quad (5.41)$$

を満たす \bar{A} が存在すると仮定する. この条件を満たすとき, 式 (5.39) と式 (5.38) から, $\mathbf{y}(t_{n+1})$ と \mathbf{y}_{n+1} の差分として大域誤差 e_{n+1} は

$$e_{n+1} = e_n + h\{\Phi(t_n, \mathbf{y}(t_n); h) - \Phi(t_n, \mathbf{y}_n; h)\} + h\mathbf{T}_{n+1} \quad (5.42)$$

と記述できる.

ここで具体的に, 解くべき方程式系として式 (5.4) の変数係数線型系を用い, それに適用する数値積分として 2 段 3 次 ROW 型解法を用いることを考える. このとき $L(t)$ および $F(t)$ が 3 回微分可能で, さらに Lipschitz 定数 \bar{L} が, $\|L(t)\mathbf{y} - L(t)\mathbf{y}^*\| \leq \bar{L}\|\mathbf{y} - \mathbf{y}^*\|$ を満たすとする. ここで段値方程式の解, i 番目の計算過程 k_i を解くのに線型方程式 (5.6) が誤差なく計算できると仮定するとき, k_i について,

$$k_i = hL(t_n + Ch) \sum_{j=1}^s \alpha_{ij} k_j + L(t_n + \gamma_i h) \mathbf{y}_n + F(t_n + \gamma_i h)$$

と表すことができ，一方，対応する $\mathbf{k}_i^* = \mathbf{k}_i(\mathbf{y}(t_n))$ も同様に記述すると，

$$\mathbf{k}_i^* = h\mathbf{L}(t_n + Ch) \sum_{j=1}^s \alpha_{ij} \mathbf{k}_j^* + \mathbf{L}(t_n + \gamma_i h) \mathbf{y}(t_n) + \mathbf{F}(t_n + \gamma_i h)$$

となる．実際，それぞれの線型方程式が解をもつための条件は， $(\mathbf{I} - h\mathbf{A} \otimes \mathbf{L}(t_n + Ch))$ が逆行列をもつことである．つまり，収束する必要条件として，解くべき常微分方程式系が適当な形の線型系であることが挙げられる．この条件のもとで，Lipschitz 連続性により，不等式

$$\|\mathbf{k}_i - \mathbf{k}_i^*\| \leq \bar{L} \|\mathbf{y}(t_n) - \mathbf{y}_n\| + h\bar{L} \sum_{j=1}^s |\alpha_{ij}| \|\mathbf{k}_j - \mathbf{k}_j^*\|$$

が得られ，この $\|\mathbf{k}_i - \mathbf{k}_i^*\|$ についての連立不等式の解は，ある正の定数 M を用いて，

$$\|\Phi(t_n, \mathbf{y}(t_n); h) - \Phi(t_n, \mathbf{y}_n; h)\| \leq M \|\mathbf{e}_n\| \quad (5.43)$$

と評価することができる．これと式 (5.41) をあわせれば，差分不等式

$$\|\mathbf{e}_{n+1}\| \leq (1 + hL') \|\mathbf{e}_n\| + \bar{A}h^{p+1}$$

が得られる． $\mathbf{e}_0 = 0$ を考慮すると，

$$\|\mathbf{e}_n\| \leq \frac{\bar{A}h^p}{L'} ((1 + hL')^n - 1), \quad \text{for } n = 0, 1, 2, \dots$$

が導出され，さらに $(1 + hL')^n \leq e^{nhL'}$ であるから， h や n に依存しない定数 C を用いて，

$$\|\mathbf{e}_n\| \leq Ch^p \quad (5.44)$$

と記述できる．つまり， $h \downarrow 0$ とするとき， $\|\mathbf{e}_n\| \rightarrow 0$ となり，ROW 型解法の大域的収束性が示される．ただし，適用する方程式系が適当な (変数係数) 線型常微分方程式系である必要がある．

5.5.2 NA 安定性解析

5.3.4 節では結論だけを述べたが，ここでは NA 安定性解析の詳細を述べる．

NA 安定性解析には以下の非自励系 (non-autonomous) のスカラー線型テスト方程式 (3.14)

$$y'(t) = \lambda(t)y(t) \quad (t \geq 0), \quad \lambda(t) \in \mathbb{C}, \quad y(0) = y_0$$

を用いる．このテスト方程式に ROW 型解法を適用して得られる数値解は，

$$\begin{aligned} y_{n+1} &= \{1 + hb(\mathbf{I} - h\lambda_n \mathbf{A})^{-1} \boldsymbol{\lambda}_n^T\} y_n \\ &= \frac{\det(\mathbf{I} - h\lambda_n \mathbf{A} + hb\boldsymbol{\lambda}_n^T)}{\det(\mathbf{I} - h\lambda_n \mathbf{A})} y_n \end{aligned} \quad (5.45)$$

の漸化式で得られる．ただし， λ_n と λ_n は，

$$\begin{aligned}\lambda_n &= \lambda(t_n + Ch), \\ \lambda_n &= (\lambda(t_n + \gamma_1 h), \dots, \lambda(t_n + \gamma_s h))^T\end{aligned}$$

を意味する．ここで， $(s+1)$ 次元の複素ベクトル z

$$\begin{aligned}z &= (z_0, z_1, \dots, z_s)^T = (z_0, \bar{z}^T)^T \\ &= (h\lambda_n, h\lambda_n^T)^T\end{aligned}$$

を導入すると，ROW 型解法を適用するときの NA 安定性因子は，以下のように表わされる．

$$R(z) \equiv \frac{\det(\mathbf{I} - z_0 \mathbf{A} + \mathbf{b} \bar{z}^T)}{\det(\mathbf{I} - z_0 \mathbf{A})}. \quad (5.46)$$

ここでさらに，以下のような集合

$$\mathcal{R} \equiv \{z; |R(z)| \leq 1\} \subset \mathbb{C}^{s+1}$$

を NA 安定領域と呼び，以下の包含関係

$$\mathcal{R} \supset \mathbb{C}^{s+1,-} \equiv \{z; \Re z_i < 0, i = 0, 1, \dots, s\}$$

が保持される時，つまり， z の存在領域が複素左半平面に相当する $(s+1)$ 次元の領域をすべて含むとき，適用する解法が無条件に NA 安定であるという．そうでない場合，つまり，NA 安定領域は存在するが，それがすべての $\mathbb{C}^{s+1,-}$ をは含まない場合は，条件付き NA 安定と呼び，定義どおりの解析は非常に困難であるため，何らかの新たな指標や仮定が必要である．

本節における 2 段 3 次の ROW 型解法についてと次章のブロック Rosenbrock 法の NA 安定性解析，6.4 節においてとは，それぞれ異なる条件付き NA 安定性解析の手法を提案している．これらは，ROW 型についての解析をより一般的にしたものが一般型陰的 Runge-Kutta 法やブロック Rosenbrock 法の解析であるということが出来る．

ここで改めて，我々が構成した 2 つの ROW 型解法のスキームについて解析する．1 段 2 次公式については，5.3.3 節において既に，無条件 NA 安定であることを示したので，ここでは 2 段 3 次スキームについて解析する．NA 安定性因子は以下のとおりである．

$$R(z) = \frac{1 - \frac{5}{3}z_0 + \frac{3}{4}z_1 + \frac{1}{4}z_2 + \frac{2}{3}z_0^2 - z_0z_1 - \frac{1}{6}z_0z_2}{1 - \frac{5}{3}z_0 + \frac{2}{3}z_0^2}. \quad (5.47)$$

まず， $z_0 = 0$ および $z_1 = 0$ ， $z_2 < -8$ の組のように明らかに $|R(z)| > 1$ となる場合があるので，この公式が無条件 NA 安定でないことがわかる．しかし，実際には複素変数 z_0 およ

び z_1, z_2 は独立でなく, 時間刻み幅 h に依存している. このことから h が十分小さいとき, Taylor 展開を適用すると,

$$z_1 = z_0 - \frac{1}{3}h^2\lambda'(t_n + \frac{2}{3}h) + O(h^3), \quad (5.48)$$

$$z_2 = z_0 + \frac{1}{3}h^2\lambda'(t_n + \frac{2}{3}h) + O(h^3) \quad (5.49)$$

となり, 差分 $(z_1 - z_0)$ および $(z_2 - z_0)$ のオーダーは, $O(h^2)$ といえる. これを踏まえて $z_1 = z_0 - \hat{h}^2$, $z_2 = z_0 + \hat{h}^2$, $\hat{h} \in \mathbb{C}$ とし, 変数を z_0 のみとする安定性因子を新たに $\hat{R}(z_0)$ とすると,

$$\begin{aligned} \hat{R}(z_0) &\equiv R(z_0, z_0 - \hat{h}^2, z_0 + \hat{h}^2) \\ &= \frac{1 - \frac{2}{3}z_0 - \frac{1}{2}z_0^2 - \frac{1}{2}\hat{h}^2 + \frac{5}{6}\hat{h}^2z_0}{1 - \frac{5}{3}z_0 + \frac{2}{3}z_0^2} \\ &= \frac{|Q(z_0)|}{|P(z_0)|} \end{aligned} \quad (5.50)$$

は十分小さな \hat{h} についての NA 安定領域を近似する. つまり, NA 安定性解析は, 条件付きという意味で, $\hat{h} \in \mathbb{R}$ の区間を変化させるときの $|\hat{R}(z_0)| \leq 1$ を満たす複素平面上の領域を評価することに置き換えられる.

まず, $\hat{R}(0) = 1 - \hat{h}^2/2$ を保持するので, 区間 $0 \leq \hat{h} \leq 2$ において, 条件 $|\hat{R}(0)| \leq 1$ が得られる. 一方, $P(z_0)$ の根は $z_0 = \frac{3}{2}, 1$ であり, $\hat{R}(z_0)$ の極は実軸上の正の値のみとなる (零安定). 次に, 実軸の負の範囲において, $|\hat{R}(z_0)|$ の大きさを評価する. この実軸の $-\infty$ の極限は,

$$\lim_{z_0 \rightarrow -\infty} \hat{R}(z_0) = -\frac{3}{4}$$

のようになり, 実軸上の z_0 が $\infty < z_0 < 0$ の範囲で, $|\hat{R}(z_0)| \leq 1$ を保持する可能性を示す. 実際, そのための \hat{h} の条件は以下のとおりである ($A(0)$ 安定).

$$0 \leq \hat{h} \leq \frac{1}{5}\sqrt{2(32 + 3\sqrt{11})} \simeq 1.8319.$$

ここでさらに, 虚軸上における $|\hat{R}(z_0)|$ の大きさを評価する. $\hat{R}(z_0)$ において, $z_0 = iy$ ($y \in \mathbb{R}$) と置き換えると

$$\hat{R}(iy) = \frac{1 + \frac{17}{18}y^2 - \frac{1}{3}y^4 - \frac{1}{2}h^2 - \frac{19}{18}h^2y^2 + i\left(y + \frac{23}{18}y^3 - 59h^2y^3\right)}{1 + \frac{13}{9}y^2 + \frac{4}{9}y^4} \quad (5.51)$$

となる. ここで, 実軸の $-\infty$ の極限の場合と同様に, 虚軸上における $\pm\infty$ の極限は,

$$\lim_{y \rightarrow \pm\infty} |\hat{R}(iy)| = \frac{3}{4}$$

となり, \hat{h} についての条件

$$0 \leq \hat{h} \leq \bar{h} \equiv \frac{1}{5} \sqrt{\frac{2}{3} \left(58 + \frac{652}{\sqrt[3]{1693 + 189i\sqrt{7679}}} + \sqrt[3]{1693 + 189i\sqrt{7679}} \right)} \quad (5.52)$$

を満たすとき, すべての実数 y について $|\hat{R}(iy)| < 1$, つまり, 左半平面すべてが絶対安定領域となる (A 安定). ただしここで, $\frac{652}{\sqrt[3]{1693 + 189i\sqrt{7679}}}$ は $\sqrt[3]{1693 + 189i\sqrt{7679}}$ の共役複素数であるから, \bar{h} は実数であり, その値はおよそその値で $\bar{h} \simeq 1.6579$ である.

図 5.2 は複素平面上の領域,

$$\left\{ z_0 \in \mathbb{C}; |\hat{R}(z_0)| < 1, \hat{h} = \bar{h} \right\}$$

が絶対安定領域 (条件付き NA 安定領域) を示し, そこで, 黒く塗られている領域が左半平面全体 \mathbb{C}^- を含んでいることがわかる.

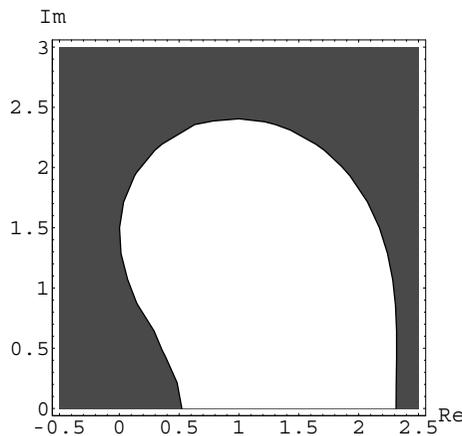


図 5.2: 条件付き NA 安定領域 ($\bar{h} \simeq 1.6579$ のとき)

また, 図 5.3 は, \hat{h} の変化による条件付き NA 安定領域の変化の様子を示している. 左側は $\hat{h} = 0$ で実際の A 安定性を示す絶対安定領域と同値である. 中央は $\hat{h} \simeq 1.8319$ で NA 安定領域という意味での $A(0)$ 安定性と同値である. 右側は $\hat{h} = 2$ で同様に NA 安定領域という意味で零安定と同値である. しかし, 実際には \hat{h} は $(z_1 - z_0), (z_2 - z_0)$ を評価する指標でしかなく, 時間刻み幅 h に依存するものの h によって制御できるものでないことに注意が必要である.

ここまでの NA 安定性解析の結果, 時間刻み幅 h が十分小さいとき, 我々が構成した 2 段 3 次の ROW 型解法の公式の NA 安定性因子 $|R(z)| < 1$ となり, この解法が条件付き NA 安定であるといえる.

また, ここで述べた従来の NA 安定性因子 $R(z)$ および新たに提案する条件付き NA 安定性因子 $\hat{R}(z_0)$ は, A 安定性因子の上位集合であり, $z_0 = z_1 = \dots = z_s$ のときこれらは一致

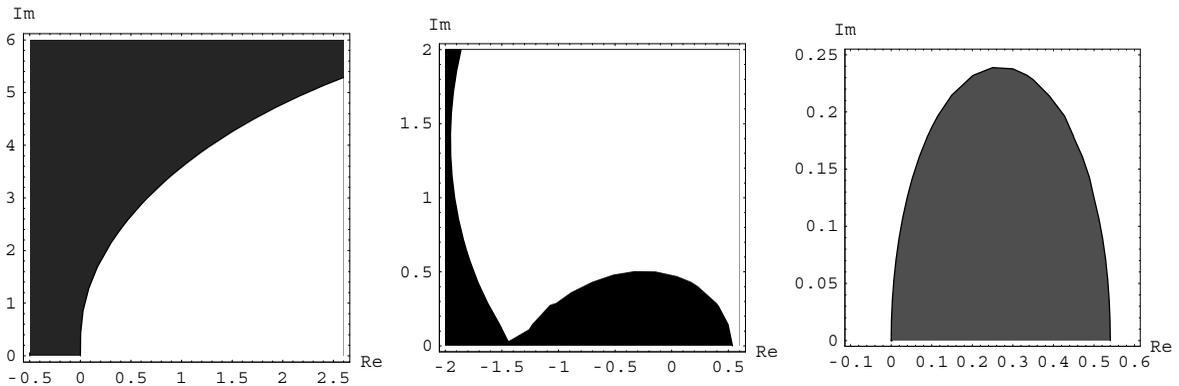


図 5.3: 条件付き NA 安定領域の変化 (左: $\hat{h} = 0.0$, 中: $\hat{h} \simeq 1.8319$, 右: $\hat{h} = 2.0$)

する．言い換えると，式 (5.47) および式 (5.50) は， $\hat{h} = 0$ のとき A 安定性因子そのものになる．ゆえに (5.52) の条件を保つとき，我々の方法は A 安定であるともいえる．

条件付き NA 安定については，第 6 章においては，本章での解析を拡張して定義することになる．

5.6 数値実験

本節では，構成したスキームの収束性および並列性を数値実験により検証する．実験を行う上で数値的に解くテスト方程式系を，以下の d 次元変数係数線型系とする．

$$\begin{aligned} \frac{d\mathbf{y}(t)}{dt} &= \mathbf{L}(t)\mathbf{y}(t) + \mathbf{g}'(t) - \mathbf{L}(t)\mathbf{g}(t), \\ \mathbf{y}(0) &= \mathbf{g}(0). \end{aligned}$$

このテスト向けに構成した方程式系の解析解は， $\mathbf{y}(t) = \mathbf{g}(t)$ である．また，硬度比は $10^2 \sim 10^6$ 程度で変化し， 10^6 程度となるのは $d = 200$, $t = 0.875$ のときである．以下では，それぞれの係数行列およびベクトルを

$$\begin{aligned} \mathbf{L}(t) &= \text{tridiag} \left(1 - \frac{1}{2} \sin t, 1, 1 - \frac{1}{2} \cos t \right), \\ \mathbf{g}(t) &= e^{-2t} \begin{bmatrix} 1 \\ 2 \\ \vdots \\ d \end{bmatrix} \end{aligned}$$

のように定める．

5.6.1 数値的収束性の検証

構成した ROW 型解法の収束性を確認する．ここで，解くべき方程式系は上記のテスト方程式とし，適用する解法は本章で構成した2つの ROW 型解法，1段2次および2段3次のものである．ここで，2段3次公式については並列化アルゴリズムを採用するが，数値実験は逐次計算機上において実施した．収束性の振る舞いとして，時間刻み幅 h に対する $t = 1.0$ における絶対誤差の変化の様子を図 5.4 に示す．また，テスト方程式の次元数は $d = 200$ とする．

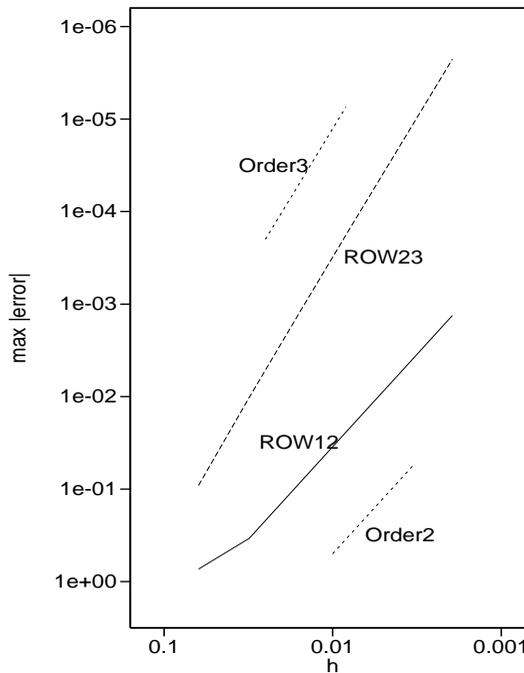


図 5.4: $t = 1.0$ における最大絶対誤差

図中の ROW12 と ROW23 は，1段2次および2段3次それぞれの解法の収束性を示している．また，Order2 は $O(h^2)$ ，Order3 は $O(h^3)$ の傾斜に対応している．それぞれの実験結果は，期待通りの収束の次数を示している．その他の条件の実験として t や d を変化させた場合についても，誤差の振る舞いは本質的には違いがなく，同様の収束性を示した．

5.6.2 並列計算による速度向上度

次に，並列アルゴリズムの性能評価をする．並列計算環境は 2.1 節で述べた VT-Alpha6 を用いた PC クラスタ 4CPU のものである．今回の計算は2個のプロセッサを用いる並列化であるが，そのプロセスの割り当ては既定の状態では，1台目の PC の2つの CPU を利用することになり，本実験ではこの設定を用いる．プロセスを2台の PC 別々に分配することも可能であるが，実験結果からは両者に違いは見られなかった．

数値実験は、先に述べたテスト問題について方程式の次元数 d を適当な値に定めて $t = 1.0$ まで計算をするが、適用する数値解法は、今回の2段3次のROW型解法については、1CPUモードと2CPUモードそれぞれで実行し、その速度向上度を見積もった。また、同様の実験をAN安定で2段4次の陰的Runge-Kutta法であるButcher-Kuntzmann公式 [21](以下でBK24と表記する)を1CPUモードで実行し、これを従来の有効な方法として、新しい並列化の実行時間と比較した。K. BURRAGE [3]も述べるように、新しく提案される並列化アルゴリズムは、従来の逐次計算アルゴリズムのうちもっとも効果的なものと比較して評価されるべきである。2段4次Butcher-Kuntzmann公式は少ない段数で、十分な収束性や安定性を示す解法として知られているが、反面一般的な微分方程式に対しては、その計算過程で現れる非線形方程式を解く何らかの反復解法のために実用的でないと言われる。しかし、我々が適用する変数係数線型系については、5.1節で述べたように、計算過程が $2d$ 次元の線型方程式となり計算量は $O((2d)^3)$ となる。この計算量は決して少ないものではないが、今回のようにAN安定性が要求されている場合においてはよく利用されるものである。

表 5.2: 実行時間 (秒) と速度向上度

d	$\max \text{error} $	BK24	ROW23	ROW23	Speedup	
		1-CPU	1-CPU	2-CPU	(1)/(3)	(2)/(3)
		(1)	(2)	(3)		
200	1.0E-3	1.20	1.49	0.76	1.58	1.96
200	1.0E-4	2.40	3.26	1.70	1.41	1.91
200	1.0E-5	4.20	6.75	3.53	1.19	1.91
200	1.0E-6	8.27	14.35	7.48	1.11	1.92
400	1.0E-3	10.08	10.23	5.20	1.93	1.96
400	1.0E-4	20.21	22.38	11.48	1.76	1.94
400	1.0E-5	35.31	46.35	23.75	1.49	1.95
400	1.0E-6	69.26	98.46	59.38	1.37	1.95

表 5.2 は、それぞれのプログラムの実行時間 (秒) および並列化による速度向上度を示している。ここで、すべてのアルゴリズムは行列 $L(t)$ の要素がすべて存在するとして、要素が 0 の場合であっても同様に計算している。数値実験は以下のように実施した。まず、解くべき方程式系の次元数を $d = 200$ とした。この条件で実行し、 $t = 1.0$ における最大絶対誤差が同程度の大きさになる固定時間刻み幅 h についての実行時間を計測する。ここで述べたように、誤差のレベルを揃えるような場合、各プログラムの刻み幅が異なるから、要求される精度によってそれぞれの解法が要する総ステップ数は異なる。ここでの要求精度は、 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} と変化させて実験を行った。また、同様の実験を $d = 400$ の場合についても実施した。

表 5.2 の Speedup の項目の (2)/(3) が ROW23 の並列化による速度向上度 (speedup) を示している。その値が 1.91 ~ 1.96 となっているので、2CPU での理想値 2 に近い値であり、解法の並列化が有効であるといえる。一方、Speedup の項目の (1)/(3) は、従来の逐次解法から

の速度向上度を示している．4次公式である BK24 に対して3次公式の ROW23 は，同精度の解を得るための総ステップ数が多く必要である．しかし，ROW23 は計算過程で解くべき線型方程式のサイズが d 次元で済み，それに対して BK24 は $2d$ 次元の線型系を解く必要がある．ゆえに，たとえば，次元数 $d = 400$ ，誤差レベル $1.0E-3$ である場合，BK24 に対する ROW23 の速度向上度は 1.93 である．この値は，ここで示した実験結果の中ではもっとも有効である．

以上より，この2つの速度向上度の結果から，今回の2CPUを用いる ROW 型解法の並列計算が効果的であるといえる．特に，行列 $L(t)$ のサイズが大きく，数値解の精度がそれほど高いものが要求されていないとき，並列化が有効であるといえる．

次に，係数行列 $L(t)$ が疎行列で，特にここでは3重対角行列であることの利点を用いるようにプログラムを修正する．これによって，ROW23 の場合は計算過程において， d 次元の3重対角の線型方程式系を2つ解くことになる．ここでは，3重対角の線型方程式を効率よく解く Thomas アルゴリズム [28] が適用可能である．しかし，BK24 に対しては3重対角性が十分には利用できず，帯幅が $(d + 1)$ の帯行列の線型系を解く必要がある．ここでの線型方程式の解法の改良は，それぞれの場合に対して，計算量を大幅に削減する．特に Thomas アルゴリズムを適用する ROW23 の場合は顕著で，計算量の見積もりで $O(d^3)$ から $O(d)$ にまで落ちる．BK24 の計算量が依然として $O(d^3)$ であることから，ROW23 は次元数 d の増加に対してより有効となる．この点も ROW 型解法の有効な点の一つである．この結果を表 5.3 に示す．アルゴリズムの計算量の違いによる実行時間の増加の違いが明らかにみることができる．

表 5.3: 疎行列向けアルゴリズムの実行時間 (秒)

d	$\max \text{error} $	BK24	ROW23	ROW23
		1-CPU	1-CPU	2-CPU
200	1.0E-6	3.57	0.31	0.26
400	1.0E-6	27.25	0.63	0.49
800	1.0E-6	293.80	1.26	0.80
1600	1.0E-6	3342.18	2.51	1.57

5.7 本章のまとめ

本章では，変数係数線型常微分方程式系に適用可能な並列化向き ROW 型の数値積分法を提案した．ここで，具体的な公式として最も単純な解法である1段2次公式と，2個のプロセッサで並列計算が可能な2段3次の公式を構成した．それぞれは，有効な NA 安定性を示し，また，PC クラスタ型の並列計算環境において，2段3次公式の並列化の速度向上度は良好な結果を示した．

しかし，その到達可能次数は3で高次の精度が要求されるような実用的な場合においては，十分な成果を示さないといえる．したがって，今後は ROW 型解法を拡張して，我々が実用

的と考える4次の公式を構成することを目指す。ただし、次の新たな拡張近似において、本章で得られたROW型解法の並列性をなるべく損なわないことが必要である。

第6章 ブロック Rosenbrock 法とその並列化

第5章で提案した新しい ROW 型の離散変数法は，変数係数線型系に適用可能な解法で，良好な安定性や並列化効果を示すものであったが，その到達可能次数は3であり，実用の点では十分なものではなかった．

本章では，論文 [11] に基づいて，前章の ROW 型解法に比して並列性をなるべく損なわずに，4 次以上を達成するような拡張を考える．これを新たにブロック Rosenbrock 法と呼ぶ．その過程で，ROW 型解法や陰的 Runge-Kutta 法をより一般的に記述し直し，その上での収束性・安定性についての解析を行う．また，この解法の有効性を示すテスト実験を実施し，従来の陰的 Runge-Kutta 法や ROW 型解法との比較をする．

6.1 ブロック Rosenbrock 法の一般的な記述と高次達成のための拡張

前章で述べたとおり，変数係数線型常微分方程式系

$$\mathbf{y}' = \mathbf{L}(t)\mathbf{y} + \mathbf{F}(t), \quad \mathbf{y}(0) = \mathbf{y}_0 \tag{6.1}$$

に適用する陰的 Runge-Kutta 法は，

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s \beta_i \mathbf{k}_i, \tag{6.2}$$

$$\begin{aligned}
 & \begin{bmatrix} \mathbf{I} - h\alpha_{11}\mathbf{L}(t_n + \gamma_1 h) & -h\alpha_{12}\mathbf{L}(t_n + \gamma_1 h) & \cdots & -h\alpha_{1s}\mathbf{L}(t_n + \gamma_1 h) \\ -h\alpha_{21}\mathbf{L}(t_n + \gamma_2 h) & \mathbf{I} - h\alpha_{22}\mathbf{L}(t_n + \gamma_2 h) & \cdots & -h\alpha_{2s}\mathbf{L}(t_n + \gamma_2 h) \\ \vdots & & \ddots & \vdots \\ -h\alpha_{s1}\mathbf{L}(t_n + \gamma_s h) & -h\alpha_{s2}\mathbf{L}(t_n + \gamma_s h) & \cdots & \mathbf{I} - h\alpha_{ss}\mathbf{L}(t_n + \gamma_s h) \end{bmatrix} \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{L}(t_n + \gamma_1 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_1 h) \\ \mathbf{L}(t_n + \gamma_2 h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_2 h) \\ \vdots \\ \mathbf{L}(t_n + \gamma_s h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_s h) \end{bmatrix} \tag{6.3}
 \end{aligned}$$

のように，次のステップへの漸化式と段値 (stage value) の線型方程式で記述される．本節では，これをもとに ROW 型公式を拡張する．

まず，ROW 型公式の拡張を考える上で，より一般化した陰的 Runge-Kutta 型形式 (general IRK) として，変数係数線型系に適用するときの段値 k の方程式を以下のように記述する．

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} - h\alpha_{11}\mathbf{L}(t_n + C_1h) & -h\alpha_{12}\mathbf{L}(t_n + C_1h) & \cdots & -h\alpha_{1s}\mathbf{L}(t_n + C_1h) \\ -h\alpha_{21}\mathbf{L}(t_n + C_2h) & \mathbf{I} - h\alpha_{22}\mathbf{L}(t_n + C_2h) & \cdots & -h\alpha_{2s}\mathbf{L}(t_n + C_2h) \\ \vdots & \vdots & \ddots & \vdots \\ -h\alpha_{s1}\mathbf{L}(t_n + C_sh) & -h\alpha_{s2}\mathbf{L}(t_n + C_sh) & \cdots & \mathbf{I} - h\alpha_{ss}\mathbf{L}(t_n + C_sh) \end{bmatrix} \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}(t_n + \gamma_1h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_1h) \\ \mathbf{L}(t_n + \gamma_2h)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_2h) \\ \vdots \\ \mathbf{L}(t_n + \gamma_sh)\mathbf{y}_n + \mathbf{F}(t_n + \gamma_sh) \end{bmatrix}. \end{aligned} \quad (6.4)$$

この記述は，陰的 Runge-Kutta 法，従来の ROW 型解法そして今回の ROW 型解法の拡張をも含んでいる．つまり， $C_i = \gamma_i$ ($i = 1, 2, \dots, s$) である場合は陰的 Runge-Kutta 法となり，すべての C_i について $C_i = C$ とおく場合は ROW 型である．以下では，この式 (6.2), (6.4) の組について次数条件を調べてパラメタを決定し，収束性・安定性を解析する．

本章で提案する ROW 型解法の拡張は，ブロック半陰的な拡張である．新しく拡張する解法は，変数係数線型系 (6.1) に適用するときを得られる k についての線型方程式において，各ブロックの計算を，式 (5.7)

$$(\mathbf{I} - h\mathbf{A} \otimes \mathbf{L}_C)\mathbf{K} = \Phi$$

のような ROW 型とし，そこでの計算が並列化可能とする．各ブロックごとの計算がこれと同等となるようにするためには，ブロック半陰的な公式を構成すればよく，線型方程式としては，係数行列がブロック上三角行列となるようなものである (式 (6.5))．つまり，一般化陰的 Runge-Kutta 公式 (6.4) を b 個のブロックに分割し，そのブロック内の C_i をすべて等しい C_k と定めるものである．

$$\begin{bmatrix} \mathbf{I} - h\mathbf{A}_{11} \otimes \mathbf{L}_{C_1} & -h\mathbf{A}_{12} \otimes \mathbf{L}_{C_1} & \cdots & -h\mathbf{A}_{1b} \otimes \mathbf{L}_{C_1} \\ & \mathbf{I} - h\mathbf{A}_{22} \otimes \mathbf{L}_{C_2} & \cdots & -h\mathbf{A}_{2b} \otimes \mathbf{L}_{C_2} \\ & & \ddots & \vdots \\ 0 & & & \mathbf{I} - h\mathbf{A}_{bb} \otimes \mathbf{L}_{C_b} \end{bmatrix} \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \\ \vdots \\ \mathbf{K}_b \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_b \end{bmatrix}. \quad (6.5)$$

ここで， \mathbf{A}_{kl} は $(s_k \times s_l)$ 行列 ($k, l = 1, 2, \dots, b$) で， \mathbf{L}_{C_k} は $\mathbf{L}(t_n + C_kh)$ ($k = 1, 2, \dots, b$) を意味する．また， \mathbf{K}_k , Φ_k は，適当な段値と右辺のベクトルに対応している．このように新たに構成する数値積分法 (6.2), (6.5) をブロック Rosenbrock 法と呼ぶ．ブロック Rosenbrock 法の公式を特定する段数についての表記は，ブロック数 b , i 番目の対角要素に対応する段数を s_i とするとき， $(b, s_1, s_2, \dots, s_b)$ 型と定義する．

ここで， $(2, 2, 2)$ 型ブロック Rosenbrock 公式の場合を例にとって実際の計算手順を説明する．また，本章ではこの $(2, 2, 2)$ 型の 4 次公式を構成することを主題とする．まず，下段の

ブロック方程式は,

$$(I - hA_{22} \otimes L_{C_2})K_2 = \Phi_2 \quad (6.6)$$

のように式 (5.7) と同等となる。したがって、 A_{22} が相似変換による対角化が可能なとき、この線型方程式に対して第 5 章で述べたような、2 個のプロセッサを用いた並列化が可能である。次に、上段の計算に移る。ここで解くべきブロック方程式も

$$(I - hA_{11} \otimes L_{C_1})K_1 = \Phi_1 + h(A_{12} \otimes L_{C_1})K_2 \quad (6.7)$$

のように式 (5.7) と同等である。したがって、これについても ROW 型の並列計算が適用できる。ただし、下段で計算した K_2 を利用することに注意が必要である。つまり、各ブロック内の計算において ROW 型解法に倣った並列化が可能であるが、ブロック間の計算は逐次的に実行することになる。

以上より、我々は上記で述べた (2, 2, 2) 型スキームを構成し、2CPU の並列計算機を用いる場合の 1 ステップに要する計算時間のオーダは、従来の ROW 型解法の同等のサイズの 4 段公式 (4CPU) に対して 2 倍となる。このオーダは、ブロックの段数 b に依存するが、各ブロックサイズに関する s_i には依存しない。つまり、 i 番目のブロックの計算は s_i 個のプロセッサを用いて ROW 型と同じオーダで計算可能であり、その各々の計算については逐次的に b 回処理する必要があるのである。ここでの拡張により、1 ステップに要する計算時間は b に依存して線型に増加することになるが、高次公式を構成することによって、同精度の数値解を得るために必要な総ステップ数が ROW 型解法よりも少なく済むことによる計算量の軽減が期待できる。

6.2 4次公式の構成

一般化陰的 Runge-Kutta 型公式 (6.2), (6.4) についての Butcher 配列は以下のように拡張する。新たなパラメタはベクトル $C = (C_i)$ である。

$$\frac{C \quad c \mid A}{\quad \quad \mid b^T}.$$

5.3 節の場合と同様に、スキームの次数条件を達成するために、真の解 $y(t_{n+1})$ と数値解 y_{n+1} の両方を Taylor 展開し、 $y_n = y(t_n)$ の仮定のもとでその係数を比較する。真の解についての展開は

$$y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \frac{h^3}{6}y'''(t_n) + \frac{h^4}{24}y^{(4)}(t_n) + O(h^5) \quad (6.8)$$

でその各導関数の展開

$$\begin{aligned}
 \mathbf{y}'(t_n) &= Ly + F, \\
 \mathbf{y}''(t_n) &= L(Ly + F) + L'y + F', \\
 \mathbf{y}'''(t_n) &= L^2(Ly + F) + 2L'(Ly + F) \\
 &\quad + L(L'y + F') + L''y + F'', \\
 \mathbf{y}^{(4)}(t_n) &= L^3(Ly + F) + 5LL'(Ly + F) \\
 &\quad + 3L''(Ly + F) + L^2(L'y + F') \\
 &\quad + 3L'(L'y + F') + L(L''y + F'') \\
 &\quad + L'''y + F'''
 \end{aligned}$$

も 5.3 節で述べたとおりである．一方，数値解 \mathbf{y}_{n+1} の展開は以下ようになる．

$$\begin{aligned}
 \mathbf{y}_{n+1} &= \mathbf{y}_n + h \sum_{i=1}^s \beta_i (Ly + F) \\
 &\quad + h^2 \left\{ \sum_{i=1}^s \beta_i \gamma_i (L'y + F') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} L(Ly + F) \right\} \\
 &\quad + h^3 \left\{ \frac{1}{2} \sum_{i=1}^s \beta_i \gamma_i^2 (L''y + F'') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j L(L'y + F') \right. \\
 &\quad \left. + \sum_{i=1}^s \beta_i C_i \sum_{j=1}^s \alpha_{ij} L'(Ly + F) + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} L^2(Ly + F) \right\} \\
 &\quad + h^4 \left\{ \frac{1}{6} \sum_{i=1}^s \beta_i \gamma_i^3 (L'''y + F''') + \frac{1}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j^2 L(L''y + F'') \right. \\
 &\quad + \sum_{i=1}^s \beta_i C_i \sum_{j=1}^s \alpha_{ij} \gamma_j L'(L'y + F') + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \gamma_l L^2(L'y + F') \\
 &\quad + \frac{1}{2} \sum_{i=1}^s \beta_i C_i^2 \sum_{j=1}^s \alpha_{ij} L''(Ly + F) \\
 &\quad + \left(\sum_{i=1}^s \beta_i C_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} C_j \sum_{l=1}^s \alpha_{jl} \right) LL'(Ly + F) \\
 &\quad \left. + \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \sum_{m=1}^s \alpha_{lm} L^3(Ly + F) \right\} + O(h^5). \tag{6.9}
 \end{aligned}$$

式 (6.8) と式 (6.9) の同じ次数の h の項について，係数を比較することで，公式の次数条件を得ることができる．ただし，ここで得られる次数条件式は，我々が対象としている変数係数線型系についてのみ成立するものである．

6.2.1 次数条件式

ここではまず，自然な仮定として，

$$\gamma_i = \sum_{j=1}^s \alpha_{ij}, \quad \text{for } i = 1, 2, \dots, s \quad (6.10)$$

と定める．以下では，この仮定を用いて式や項を適当に消去している．

式 (6.8), (6.9), (6.10) から得られる 4 次公式のための条件式は以下のものである．

$$\sum_{i=1}^s \beta_i = 1 \quad (6.11)$$

$$\sum_{i=1}^s \beta_i \gamma_i = \frac{1}{2} \quad (6.12)$$

$$\frac{1}{2} \sum_{i=1}^s \beta_i \gamma_i^2 = \frac{1}{6} \quad (6.13)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j = \frac{1}{6} \quad (6.14)$$

$$\sum_{i=1}^s \beta_i C_i \gamma_i = \frac{2}{6} \quad (6.15)$$

$$\frac{1}{6} \sum_{i=1}^s \beta_i \gamma_i^3 = \frac{1}{24} \quad (6.16)$$

$$\frac{1}{2} \sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \gamma_j^2 = \frac{1}{24} \quad (6.17)$$

$$\sum_{i=1}^s \beta_i C_i \sum_{j=1}^s \alpha_{ij} \gamma_j = \frac{3}{24} \quad (6.18)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} \sum_{l=1}^s \alpha_{jl} \gamma_l = \frac{1}{24} \quad (6.19)$$

$$\frac{1}{2} \sum_{i=1}^s \beta_i C_i^2 \gamma_i = \frac{3}{24} \quad (6.20)$$

$$\sum_{i=1}^s \beta_i \sum_{j=1}^s \alpha_{ij} C_j \gamma_j = \frac{2}{24} \quad (6.21)$$

ここで，式 (6.21) は h^4 についての恒等式における $LL'(Ly + F)$ の項と式 (6.18) から得ることができる．

6.2.2 パラメタの決定

ここで未知数と式の数の関係から 4 次ブロック Rosenbrock 公式の最小構成は (2, 2, 1) 型または (2, 1, 2) 型であるが, s_i に並列計算オーダが依存しないことと行列の対称性を考慮して, (2, 2, 2) 段公式の構成, つまり, $b = 2, s_1 = 2, s_2 = 2$ の場合を考える. このときブロック Rosenbrock 公式を一般化陰的 Runge-Kutta 型公式とみるときの段数は $s = s_1 + s_2 = 4$ であり, まず, 上三角であるために $\alpha_{31} = 0, \alpha_{32} = 0, \alpha_{41} = 0, \alpha_{42} = 0$ が定まり, さらに, (2, 2, 2) 型の 2 つのブロックになるように $C_1 = C_2, C_3 = C_4$ が決定する. 次に $\{\gamma_i\}$ を, 定義域を $[-1, 1]$ から $[0, 1]$ に変換した, 4 次のずらし Legendre 多項式の根として以下のように定める.

$$\gamma_1 = \frac{1}{70} \left(35 - \sqrt{35(15 - 2\sqrt{30})} \right) \quad (6.22)$$

$$\gamma_2 = \frac{1}{70} \left(35 + \sqrt{35(15 - 2\sqrt{30})} \right) \quad (6.23)$$

$$\gamma_3 = \frac{1}{70} \left(35 - \sqrt{35(15 + 2\sqrt{30})} \right) \quad (6.24)$$

$$\gamma_4 = \frac{1}{70} \left(35 + \sqrt{35(15 + 2\sqrt{30})} \right). \quad (6.25)$$

これによって, $\{\beta_i\}$ は式 (6.11), (6.12), (6.13), (6.16) から一意に決めることができ, 倍精度実数値で表現すると,

$$\beta_1 = \beta_2 \simeq 0.32607257743127307 \quad (6.26)$$

$$\beta_3 = \beta_4 \simeq 0.17392742256872692 \quad (6.27)$$

のようになる.

次に, C_1 と C_3 については, 2 次の 2 元連立方程式 (6.15), (6.20) から 2 組の解

$$\begin{cases} C_1 = 0.83881017107725915 \\ C_3 = 0.34393851177186564 \end{cases} \quad (6.28)$$

$$\begin{cases} C_1 = 0.49452316225607418 \\ C_3 = 0.98939482156146769 \end{cases} \quad (6.29)$$

が得られ, ここでは式 (6.28) を採用する.

残りのパラメタ $\{\alpha_{ij}\}$ は, (6.14), (6.17), (6.18), (6.19), (6.21) および (6.10) から決定する (未知数 12, 式 9).

まず, (6.10) より,

$$\alpha_{12} = \gamma_1 - \alpha_{11} - \alpha_{13} - \alpha_{14} \quad (6.30)$$

$$\alpha_{21} = \gamma_2 - \alpha_{22} - \alpha_{23} - \alpha_{24} \quad (6.31)$$

$$\alpha_{34} = \gamma_3 - \alpha_{33} \quad (6.32)$$

$$\alpha_{43} = \gamma_4 - \alpha_{44} \quad (6.33)$$

とし, 4つの未知数と式を消去する.

式 (6.13), (6.16), (6.17), (6.21) を用いて消去を繰り返すと,

$$\alpha_{13} = -0.29985410339729551 - \alpha_{23} \quad (6.34)$$

$$\alpha_{24} = 0.29985410339729551 - \alpha_{14} \quad (6.35)$$

が導かれ, これを用いると, (6.14), (6.17), (6.19) より,

$$\alpha_{22} = -1.12641964692177122 + \alpha_{11} + \alpha_{14} - \alpha_{23} \quad (6.36)$$

$$\begin{aligned} \alpha_{33} = & -3.08467944220248923 + 4.07011715697389420\alpha_{11} \\ & -3.11953127651679558\alpha_{14} - 7.18964843349068978\alpha_{23} \end{aligned} \quad (6.37)$$

$$\begin{aligned} \alpha_{44} = & -3.03503906717323509 + 4.07011715697389420\alpha_{11} \\ & -3.11953127651679558\alpha_{14} - 7.18964843349068978\alpha_{23} \end{aligned} \quad (6.38)$$

となる. 以上より, 自由度3でパラメタを決定できる. これらの自由パラメタは, 対角化および安定性を達成するために用いる. A_{11} , A_{22} の対角化可能条件は, それぞれ,

$$(\alpha_{11} - \alpha_{22})^2 + 4\alpha_{12}\alpha_{21} > 0 \quad (6.39)$$

$$(\alpha_{33} - \alpha_{44})^2 + 4\alpha_{34}\alpha_{43} > 0 \quad (6.40)$$

である. ここで, 簡単のため $\alpha_{14} = 0, \alpha_{23} = 0$ と定めるとき, 残りの α_{11} についての対角化可能条件は $\alpha_{11} \leq 0.77513030203711903$ または $0.97413593153469024 \leq \alpha_{11}$ となる. これを踏まえて, $\alpha_{11} = 1.00625$ とするとき, A 安定性を達成するパラメタをすべて決定できる. 安定性についての解析は, 6.4節で述べるが, A 安定性達成条件は $1.00613647058398071 \leq \alpha_{11}$ である.

以上をまとめて, (2, 2, 2) 型4次公式のすべてのパラメタを倍精度実数で以下に示す.

$$\begin{array}{ll} \alpha_{11} = 1.00625 & \alpha_{31} = 0.0 \\ \alpha_{12} = -0.37638641839513261 & \alpha_{32} = 0.0 \\ \alpha_{13} = -0.29985410339729551 & \alpha_{33} = 1.01087594700249180 \\ \alpha_{14} = 0.0 & \alpha_{34} = -0.94144410279951808 \\ \alpha_{21} = 0.49030606531690384 & \alpha_{41} = 0.0 \\ \alpha_{22} = -0.12016964692177122 & \alpha_{42} = 0.0 \\ \alpha_{23} = 0.0 & \alpha_{43} = -0.12994816623471965 \\ \alpha_{24} = 0.29985410339729551 & \alpha_{44} = 1.06051632203174594 \end{array}$$

$$\begin{array}{ll}
\beta_1 = 0.32607257743127307 & \gamma_3 = 0.0694318442029737 \\
\beta_2 = 0.32607257743127307 & \gamma_4 = 0.9305681557970262 \\
\beta_3 = 0.17392742256872692 & C_1 = 0.83881017107725915 \\
\beta_4 = 0.17392742256872692 & C_2 = 0.83881017107725915 \\
\gamma_1 = 0.3300094782075718 & C_3 = 0.34393851177186564 \\
\gamma_2 = 0.6699905217924281 & C_4 = 0.34393851177186564
\end{array}$$

さらに，対角成分ブロックの A_{11}, A_{22} は， $A_{11} = S_1^{-1} \Lambda_1 S_1$, $A_{22} = S_2^{-1} \Lambda_2 S_2$ のように対角化可能で，対応する行列の成分パラメタ $\Lambda_i = \text{diag}(\lambda_j^{(i)})$, $S_i = \begin{pmatrix} s_{j\ell}^{(i)} \end{pmatrix}$, $S_i^{-1} = \begin{pmatrix} t_{j\ell}^{(i)} \end{pmatrix}$ は以下のようなになる．

$$\begin{array}{ll}
\lambda_1^{(1)} = 0.80726642682978542 & \lambda_1^{(2)} = 1.38634549852559605 \\
\lambda_2^{(1)} = 0.07881392624844334 & \lambda_2^{(2)} = 0.68504677050864169 \\
s_{11}^{(1)} = 1.44012843462329139 & s_{11}^{(2)} = 0.50019556522965889 \\
s_{12}^{(1)} = -0.58445514346259248 & s_{12}^{(2)} = -1.44525475035481424 \\
s_{21}^{(1)} = -0.72639611344244829 & s_{21}^{(2)} = -0.56655017298169639 \\
s_{22}^{(1)} = 1.37401106593291927 & s_{22}^{(2)} = -1.42055545417733843 \\
t_{11}^{(1)} = 0.88405955099841603 & t_{11}^{(2)} = 0.92885320219021638 \\
t_{12}^{(1)} = 0.37604730014123471 & t_{12}^{(2)} = -0.94500323721970348 \\
t_{21}^{(1)} = 0.46737427217218432 & t_{21}^{(2)} = -0.37044801090163920 \\
t_{22}^{(1)} = 0.92660046840938308 & t_{22}^{(2)} = -0.32706097542244446
\end{array}$$

6.3 大域収束性

ここで，一般化陰的 Runge-Kutta 型公式についても大域収束性を解析する．5.5.1 節の ROW 型解法の場合と同様に，離散変数法は，写像 Φ を用いて，

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\Phi(t_n, \mathbf{y}_n; h) \quad (6.41)$$

と表わされる． p 次公式についての局所離散化誤差

$$\mathbf{T}_{n+1} = \mathbf{A}(t_n, \mathbf{y}(t_n))h^p + O(h^{p+1}) \quad (6.42)$$

は，ある正数 h_0 をとるとき $h < h_0$ の条件で，正定数 \bar{A} を用いて，

$$\|\mathbf{T}_{n+1}\| \leq \bar{A}h^p \quad (6.43)$$

によって制限できる．また大域誤差 e_{n+1} は，

$$e_{n+1} = e_n + h\{\Phi(t_n, \mathbf{y}(t_n); h) - \Phi(t_n, \mathbf{y}_n; h)\} + h\mathbf{T}_{n+1} \quad (6.44)$$

と記述できる．

ここで具体的に (2, 2, 2) 型 4 次ブロック Rosenbrock 法を変数係数線型常微分方程式系 (6.1) に適用する場合について解析する． $L(t)$ および $F(t)$ が 4 回連続微分可能であるとするとき，上記の条件を満たし，さらに，Lipschitz 連続性 $\|L(t)\mathbf{y} - L(t)\mathbf{y}^*\| \leq \bar{L}\|\mathbf{y} - \mathbf{y}^*\|$ も満たす．また， Φ に関する差の項は，段値 \mathbf{k}_i とその真値 $\mathbf{k}_i^* = \mathbf{k}_i(\mathbf{y}(t_n))$ の差に依存し，ブロック Rosenbrock を一般化陰的 Runge-Kutta 法とみるとき，

$$\mathbf{k}_i = h \sum_{j=1}^s \alpha_{ij} L(t_n + C_j) \mathbf{k}_j + L(t_n + \gamma_i h) \mathbf{y}_n + F(t_n + \gamma_i h) \quad (6.45)$$

$$\mathbf{k}_i^* = h \sum_{j=1}^s \alpha_{ij} L(t_n + C_j) \mathbf{k}_j^* + L(t_n + \gamma_i h) \mathbf{y}(t_n) + F(t_n + \gamma_i h) \quad (6.46)$$

から以下のように表わされる．

$$\|\mathbf{k}_i - \mathbf{k}_i^*\| \leq \bar{L}\|\mathbf{y}(t_n) - \mathbf{y}_n\| + h\bar{L} \sum_{j=1}^s |\alpha_{ij}| \|\mathbf{k}_j - \mathbf{k}_j^*\|. \quad (6.47)$$

ここにおいても，時刻 t_n における段値方程式 (6.4) が矛盾なく解くことができる場合に

$$\|\Phi(t_n, \mathbf{y}(t_n); h) - \Phi(t_n, \mathbf{y}_n; h)\| \leq M\|e_n\| \quad (6.48)$$

という 5.5.1 節における解析と同様の結果に達する．したがって，結論も同様になり， h や n に依存しない定数 C を用いて，

$$\|e_n\| \leq Ch^4 \quad (6.49)$$

であるから， $h \downarrow 0$ とするとき， $\|e_n\| \rightarrow 0$ となり，(2, 2, 2) 型ブロック Rosenbrock 法についても大域的収束性が示される．

6.4 NA 安定性解析

NA 安定性解析の流れは，5.5.2 節における従来の ROW 型解法についての解析と同様に，非自励系 (non-autonomous) のスカラー線型テスト方程式 (3.14)

$$y'(t) = \lambda(t)y(t) \quad (t \geq 0), \quad \lambda(t) \in \mathbb{C}, \quad y(0) = y_0$$

に公式を適用して安定性因子を求め，そこで得られる安定領域を解析する．本質的な結論も同様で，無条件には NA 安定にはならないが A 安定であり，制限された刻み幅 h のもとでは

NA 安定であることを示す．しかし，条件付き NA 安定についての解析の手法は，同様のものを用いることができないので，新たな手法を考案する必要がある．

s 段一般化陰的 Runge-Kutta 型公式の NA 安定性因子は，段値方程式 (6.4) の i 段目の左辺に対応する複素変数を $z_i^{(\ell)} = h\lambda(t_n + C_i h)$ ，同様に右辺に対して $z_i^{(r)} = h\lambda(t_n + \gamma_i h)$ ($z_i^{(\ell)}, z_i^{(r)} \in \mathbb{C}$ for $i = 1, 2, \dots, s$) と記述するとき，以下のように表わされる．

$$R(\mathbf{Z}_\ell, \mathbf{Z}_r) = \frac{\det(\mathbf{I} - \mathbf{Z}_\ell \mathbf{A} + \mathbf{Z}_r \mathbf{e} \mathbf{b}^T)}{\det(\mathbf{I} - \mathbf{Z}_\ell \mathbf{A})}. \quad (6.50)$$

ただし，新たに s 次元の対角行列 $\mathbf{Z}_\ell = \text{diag}(z_i^{(\ell)})$ ， $\mathbf{Z}_r = \text{diag}(z_i^{(r)})$ および要素がすべて1のベクトル $\mathbf{e} = [1, 1, \dots, 1]^T$ を定義する．また，行列 \mathbf{Z}_ℓ ， \mathbf{Z}_r は，変数 $z_i^{(\ell)}$ ， $z_i^{(r)}$ に依存するのでこれらをあわせた $2s$ 次元ベクトル $\mathbf{z} = [z_\ell^T, z_r^T]^T$ を定義し，これによって， NA 安定性因子を $R(\mathbf{z}) = R(\mathbf{Z}_\ell, \mathbf{Z}_r)$ と記述する．

解法が無条件に NA 安定であることの定義は，

$$\mathcal{R} \equiv \{\mathbf{z}; |R(\mathbf{z})| \leq 1\} \supset \mathbb{C}^{2s,-} \equiv \{\mathbf{z}; \Re z_i < 0 \forall i\}$$

であり，一般には AN 安定条件と呼ばれる．この集合 \mathcal{R} を NA 安定領域と呼び， \mathcal{R} が存在するが， $\mathbb{C}^{2s,-}$ の領域すべてを含まない場合，その解法が条件付き NA 安定であると定める．

我々が構成するブロック Rosenbrock 法については， $|z_i^{(r)}|$ の増大に対して安定性因子の振る舞いが $|R(\mathbf{Z}_\ell, \mathbf{Z}_r)| \leq 1$ となるような制限が困難であることから，無条件 NA 安定になりえないといえる．ゆえに以下では条件付き NA 安定についての解析をする．しかし，厳密に定義にしたがって NA 安定領域を追跡する場合， $2s$ 次元複素ベクトル \mathbf{z} についての解析が必要でこの計算の実行や領域の表示は非常に困難である．したがって，条件付き NA 安定性のための新たな仮定をし，そこでの安定領域を追跡する．

まず，時間刻み幅 h が十分小さいと仮定し，ブロック Rosenbrock 公式を適用するテスト方程式の函数値 $\lambda(t_n + C_i h)$ および $\lambda(t_n + \gamma_i h)$ が Taylor 展開可能であるとする．ここで， c_i を C_i ， γ_i のどちらかを表すとして，

$$z_i = h\lambda(t_n + c_i h) \approx h\lambda(t_n) + c_i h^2 \lambda'(t_n)$$

のように3次以降の項を打ち切って近似可能である．したがって，新たな複素パラメタを $z_0 = h\lambda(t_n)$ ， $z'_0 = h^2 \lambda'(t_n)$ のように導入し，すべての $z_i^{(\ell)}$ ， $z_i^{(r)}$ を上記の2次近似で置き換えると，近似された NA 安定性因子 $\hat{R}(z_0, z'_0)$ は A 安定性因子 (3.12)

$$R(z) = \frac{\det(\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^T)}{\det(\mathbf{I} - z\mathbf{A})}$$

に対して，

$$\hat{R}(z_0, z'_0) = R(z) + O(h^3) \quad (6.51)$$

と表わされる．ここで， $\hat{R}(z, 0)$ とするとき $R(z)$ に一致し，この近似が自然なものであることがわかる．ゆえに，我々が取り扱う $\hat{R}(z_0, z'_0)$ は， h が十分小さいとき， NA 安定性因子の

2次近似であるといえる．ここでの安定領域の解析は，複素変数が2個にまで削減できたので解析が容易になったといえる．

ここで，6.2節で構成した具体的な(2, 2, 2)型ブロック Rosenbrock 公式について，近似 NA 安定性因子 $\hat{R}(z_0, z'_0)$ を得る．そして，安定領域として，

$$\{z_0 \in \mathbb{C}; |\hat{R}(z_0, z'_0)| \leq 1\}$$

をいくつかの固定した複素数 z'_0 について描画する．この結果を図 6.1 に示す．この図においては， $z'_0 = 0, -0.2, -0.1 + 0.5i, 0.1$ とするものをそれぞれ示している．

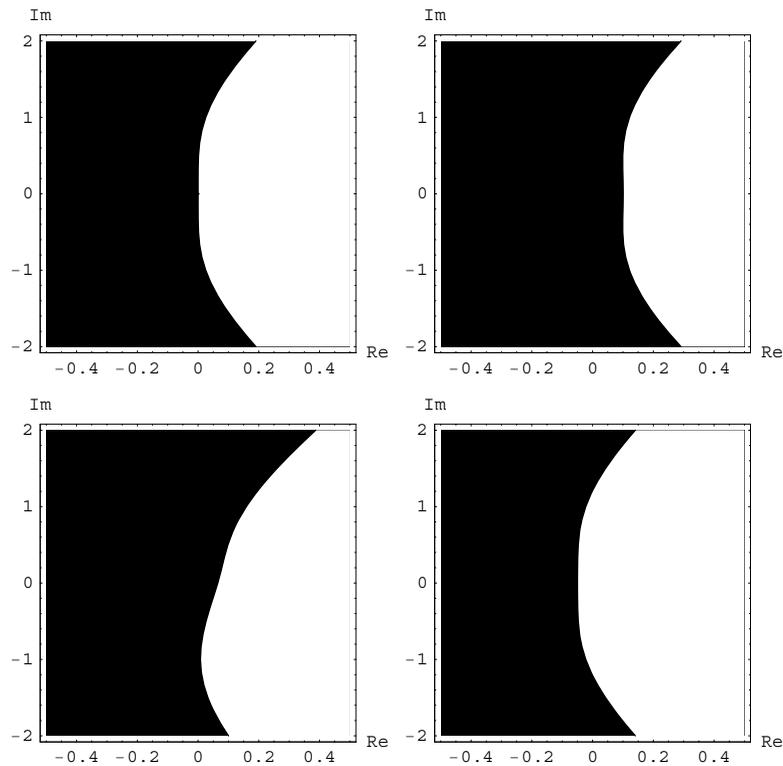


図 6.1: (2, 2, 2) 型ブロック Rosenbrock 公式の近似 NA 安定領域の様子 (左上: $z'_0 = 0$, 右上: $z'_0 = -0.2$ 左下: $z'_0 = -0.1 + 0.5i$, 右下: $z'_0 = 0.1$)

図 6.1 からわかることを以下に述べる．左上は $z'_0 = 0$ のときの様子で A 安定領域そのものを示し，これは複素左半平面すべてを含む．右上の $z'_0 = -0.2$ の様子のように， z'_0 が負の実数値である場合，近似 NA 安定領域は左半平面すべてを含む．左下のように z'_0 について，実数部は負で虚数部が 0 でない場合は，実軸に対して非対称となり，虚数部の増大とともに安定領域が縮小する傾向である．図の $z'_0 = -0.1 + 0.5i$ の場合は，安定領域は左半平面をすべて含んでいる様子である．残りの右下の図， z'_0 の実数部が正である場合は，近似 AN 安定領域はもはや左半平面すべてを含まない．ゆえに， z_0, z'_0 の組が $\Re z_0, \Re z'_0 < 0$ のような組の場合条件付き AN 安定領域が複素左半平面全体を含み，虚数部の増大や $\Re z'_0 > 0$ のような場合は不安定要因を含む傾向となる．

同様の解析および計算を無条件 NA 安定な解法である 2 段 4 次 Butcher-Kuntzmann 公式 [21] について行った結果を図 6.2 に示す．虚数部の変化によって安定領域が縮小しない点を除くと，全体的な様子はブロック Rosenbrock 法の場合と同様の傾向を示している．

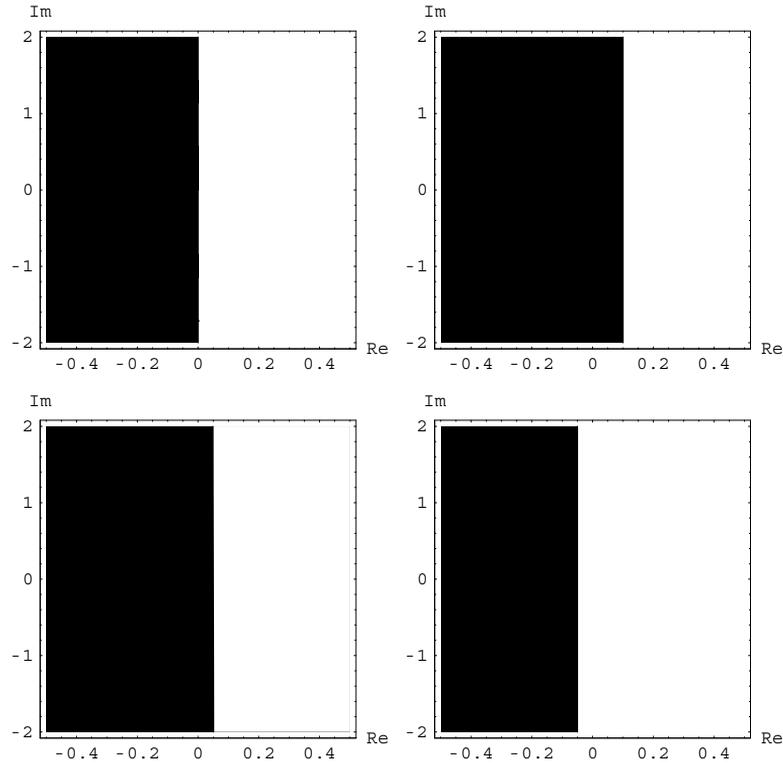


図 6.2: BK24 公式の近似 NA 安定領域の様子 (左上: $z'_0 = 0$, 右上: $z'_0 = -0.2$ 左下: $z'_0 = -0.1 + 0.5i$, 右下: $z'_0 = 0.1$)

以上より，この解析を用いた結果， $(2, 2, 2)$ 型ブロック Rosenbrock 法は，無条件には NA 安定ではないが，十分小さい時間刻み幅 h のとき NA 安定であることがわかった．しかし，この刻み幅の上限を考慮することは未だ困難な課題である．

6.5 数値実験

本節では，ブロック Rosenbrock 法として，前節までに取り扱った $(2, 2, 2)$ 型公式を用いて数値実験を実施する．実験には，第 5 章で用いたテスト方程式

$$\begin{aligned} \frac{dy(t)}{dt} &= L(t)y(t) + g'(t) - L(t)g(t), \\ y(0) &= g(0) \end{aligned}$$

を利用する．係数行列およびベクトルも同様に，

$$L(t) = \text{tridiag} \left(1 - \frac{1}{2} \sin t, 1, 1 - \frac{1}{2} \cos t \right),$$

$$g(t) = e^{-2t} \begin{bmatrix} 1 \\ 2 \\ \vdots \\ d \end{bmatrix}$$

とする．

収束の次数や並列化の効果を従来の方法，2 段 4 次 Butcher-Kuntzmann 公式 (BK24)，2 段 3 次 ROW 型解法 (ROW23) と比較する．また，移流拡散方程式の特性曲線差分法に適用する場合の問題点を検討する．

6.5.1 数値的収束性の検証

収束性の振る舞いとして，時間刻み幅 h に対する $t = 1.0$ における絶対誤差の変化の様子を図 6.3 に示す．

図中の BK24 は 2 段 4 次 Butcher-Kuntzmann 公式を，bR224 は (2, 2, 2) 型ブロック Rosenbrock 公式それぞれの収束性を示している．また，Order4 は $O(h^4)$ ，Order5 は $O(h^5)$ の傾斜に対応している．bR224 の実験結果は，期待通り 4 次収束を示しており，BK24 と比較すると，誤差の絶対値は大きい収束の次数は同等である．その他の条件の実験として t や d を変化させた場合についても，誤差の振る舞いは本質的には違いがなく，同様の収束性を示した．

6.5.2 並列計算による速度向上度

次に，並列アルゴリズムの性能評価をする．並列計算環境は 2.1 節で述べた VT-Alpha6 を用いた PC クラスタ 4CPU のものである．今回の計算は 2 個のプロセッサを用いる並列化であるが，そのプロセスの割り当ては既定の状態では，1 台目の PC の 2 つの CPU を利用することになる．プロセスを 2 台の PC 別々に分配することも可能であるが，実験結果からは両者に違いは見られない．

数値実験は，先に述べたテスト問題について方程式の次元数 d を適当な値に定めて $t = 1.0$ まで計算をするが，適用する数値解法は，本章の (2, 2, 2) 型ブロック Rosenbrock 解法 (bR224) および前章の 2 段 3 次 ROW 型公式 (ROW23) については，1CPU モードと 2CPU モードそれぞれで実行し，その速度向上度を見積もった．また，同様の実験を AN 安定で 2 段 4 次の陰的 Runge-Kutta 法である Butcher-Kuntzmann 公式 (BK24) を 1CPU モードで実行し，これを従来の有効な方法として，新しい並列化の実行時間と比較した．すなわち，実験結果は第 5 章の実験結果と比較している．

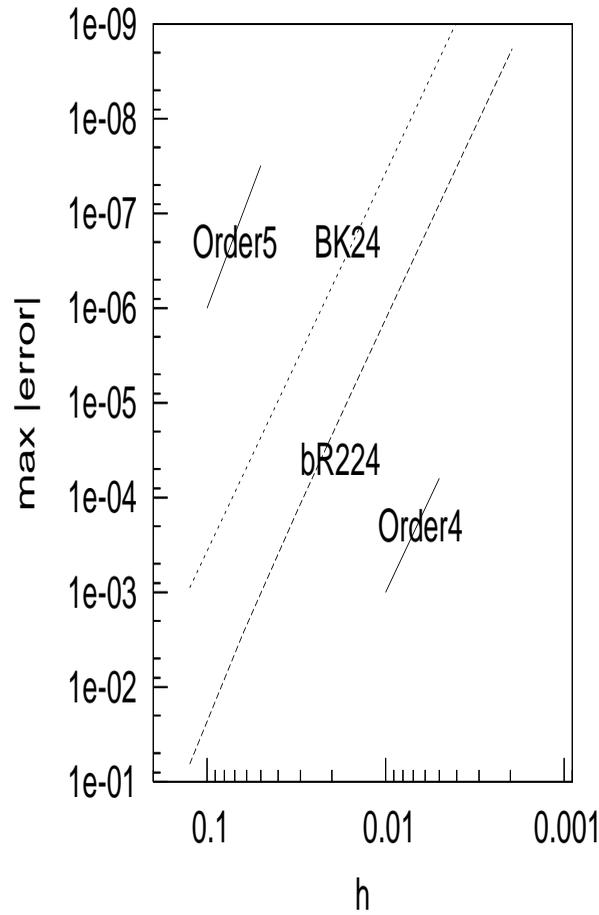
図 6.3: $t = 1.0$ における最大絶対誤差

表 6.1: 計算実行の経過時間 (秒)

d	max error	bR224	bR224	BK24	ROW23	ROW23
		1-CPU	2-CPU	1-CPU	1-CPU	2-CPU
		(1)	(2)	(3)	(4)	(5)
200	1.0E-3	2.00	1.14	1.20	1.49	0.76
200	1.0E-4	3.20	1.85	2.40	3.26	1.70
200	1.0E-5	5.59	3.22	4.20	6.75	3.53
200	1.0E-6	9.97	5.74	8.27	14.35	7.48
400	1.0E-3	13.26	7.33	10.08	10.23	5.20
400	1.0E-4	21.16	11.66	20.21	22.38	11.48
400	1.0E-5	37.06	20.37	35.31	46.35	23.75
400	1.0E-6	66.15	36.35	69.26	98.46	59.38

表 6.2: 経過時間から求めた速度向上度 (倍)

d	$\max \text{error} $	(1)/(2)	(4)/(5)	(3)/(2)	(5)/(2)	(3)/(5)
200	1.0E-3	1.75	1.96	1.05	0.66	1.58
200	1.0E-4	1.72	1.91	1.29	0.91	1.41
200	1.0E-5	1.73	1.91	1.30	1.09	1.19
200	1.0E-6	1.73	1.92	1.44	1.30	1.11
400	1.0E-3	1.80	1.96	1.37	0.70	1.93
400	1.0E-4	1.81	1.94	1.73	0.98	1.76
400	1.0E-5	1.81	1.95	1.73	1.16	1.49
400	1.0E-6	1.81	1.95	1.90	1.63	1.37

表 6.1 は各実験の経過時間 (秒) を示し, その結果から見積もった速度向上度を表 6.2 に示している. ここで, すべてのアルゴリズムは行列 $L(t)$ の要素がすべて存在するとして, 要素が 0 の場合であっても同様に計算している. 数値実験は以下のように実施した. まず, 解くべき方程式系の次元数を $d = 200$ とした. この条件で実行し, $t = 1$ における最大絶対誤差が同程度の大きさになる固定時間刻み幅 h についての実行時間を計測する. ここで述べたように, 誤差のレベルを揃えるような場合, 各プログラムの刻み幅が異なるから, 要求される精度によってそれぞれの解法が要する総ステップ数は異なる. ここでの要求精度は, $10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ と変化させて実験を行った. また, 同様の実験を $d = 400$ の場合についても実施した.

bR224 自身の速度向上度は表 6.2 の (1)/(2) の項で示されるように値は 1.73–1.81 で, ROW23 自身の速度向上度 (4)/(5) の値 1.91–1.96 に比べると劣るものではあるが, 並列化の効率として適当であることがわかる. 同表の右部は, BK24 に対する bR224(2CPU), ROW23(2CPU) に対する bR224(2CPU), BK24 に対する ROW23(2CPU) の速度向上度を示している. 前章で述べたように, ROW23 は高次元, 低精度のとき有効であったが, bR224 は今回達成した 4 次収束のため, 高精度において他の解法よりも有効で, さらに並列化の効果によって, その様子はやはり高次元のときに特に見られる. ROW23 と比較するとき, 低精度の場合は ROW23 が勝るが, 高次元かつ高精度になると bR224 が有利となり, 本章における解法の開発が有効であったといえる.

次に, 係数行列 $L(t)$ が疎行列で, 特にここでは 3 重対角行列であることの利点を用いるようにプログラムを修正する. これによって, ROW23 と bR224 の場合は計算過程において, d 次元の 3 重対角の線型方程式系を解くことになる. ここでは, 3 重対角の線型方程式を効率よく解く Thomas アルゴリズム [28] が適用可能である. しかし, BK24 に対しては 3 重対角性が十分には利用できず, 帯幅が $(d + 1)$ の帯行列の線型系を解く必要がある. ここでの線型方程式の解法の改良は, それぞれの場合に対して, 計算量を大幅に削減する. この結果を表 6.3 に示す. BK24 に対して, ROW23 と bR224 の結果は, アルゴリズムの計算量の違いによる実行時間の増加の違いが明らかにみることができる. bR224 と ROW23 の並列計算の結果を比較すると, bR224 の結果は, 解法の速度向上度では ROW23 に劣るが, 同精度の

実行結果を得るための経過時間では勝っている．

表 6.3: 疎行列に対するアルゴリズム実行の経過時間 (秒)

d	$\max \text{error} $	BK24	ROW23	ROW23	bR224	bR224
		1-CPU	1-CPU	2-CPU	1-CPU	2-CPU
200	1.0E-6	3.57	0.31	0.26	0.19	0.20
400	1.0E-6	27.25	0.63	0.49	0.38	0.38
800	1.0E-6	293.80	1.26	0.80	0.75	0.63
1600	1.0E-6	3342.18	2.51	1.57	1.52	1.26

6.6 移流拡散方程式の特性曲線差分法への適用の検討

本章で構成した離散変数法を 4.4.2 節における 2 段 4 次 Butcher-Kuntzmann 公式の代わりに適用することで，従来課題となっていた，陰的 Runge-Kutta 法の 1 ステップでの計算量の軽減が期待できる．ただし，1 ステップでの必要な段値の組が増大することに注意が必要である．ここでは，2.6 節で述べた並列計算量の見積りを用いて議論する．

計算部分はおおまかに，“特性曲線を解く部分”，“並列化係数行列を構成する部分”，“線型方程式を解く部分”，“次の数値解を求める部分”である．このうち並列化によって計算量が軽減されるのは，線型方程式を解く部分でこれを t_{pp} とおき，残りは逐次実行であるので t_{sp} である．一方，通信によるオーバーヘッドは，“線型計算結果の集約”と“次の数値解の配信”の 2 回が必要で，これらはデータ数に依存する通信である．

以上より，3 重対角向きのブロック Rosenbrock を用いるときは，線型方程式を解く部分が $t_{pp} = O(N_x)$ となるため，計算全体の計算量も $O(N_x)$ となり，データ通信にかかるオーバーヘッドは $t_{startup} = O(1)$ ， $t_{data} = O(N_x)$ が懸念される．つまり，逐次実行時間

$$t_s = t_{sp} + t_{pp} = O(N_x)$$

に対して，並列実行時間

$$t_p = t_{sp} + \frac{t_{pp}}{2} + 2t_{startup} + \left(\frac{1}{2} + 1\right) N_x t_{data} = O(N_x) + O(1)$$

であるため，特性曲線を解く部分や通信によるオーバーヘッドが顕著となる．

この見積もりは線型方程式を解く部分の計算量のオーダが $O(N_x^3)$ の場合以下のようになる．

$$t_s = O(N_x^3) + O(N_x)$$

$$t_p = \frac{O(N_x^3)}{2} + O(N_x) + O(1)$$

以上をまとめると，移流拡散方程式の特性曲線差分法にブロック Rosenbrock 法を適用するアルゴリズムの計算量の見積もりは，ブロック Rosenbrock 法の並列化や3重対角化により計算量が軽減できるが，今度はこれまでは線型計算部分に隠れていた，特性曲線を解く部分や通信のオーバーヘッドが現れてくる可能性を示している．詳細は，次章で検討する．

6.7 本章のまとめ

本章においては，変数係数線型常微分方程式系に適用する新しい離散変数法として，第5章で述べた並列化向き陰的 ROW 型解法を拡張して，さらに高次を実現することが可能なブロック Rosenbrock 法を構成した．ここで，一般化陰的 Runge-Kutta 形式で4段4次となる公式である， $(2, 2, 2)$ 型スキームを構成し，収束性および NA 安定性を解析した．数値実験においては，従来の方法である2段4次 Butcher-Kuntzmann 公式と2段3次陰的 ROW 型解法と比較して，特に，高次元，高精度が要求される問題において， $(2, 2, 2)$ 型ブロック Rosenbrock 法が実行時間の観点では有効であることを示した．

ゆえに第7章では，これまでに開発してきた変数係数線型常微分方程式系に適用する並列化向き離散変数法を，その実用として，当初の課題であった移流拡散方程式の特性曲線差分法に適用する．4.4.2 節で考案した手順において，陰的 Runge-Kutta 法の代わりに ROW 型解法やブロック Rosenbrock 法を適用するときの数値的安定性や並列計算速度向上度を実験・考察する．

第7章 移流拡散問題に対するブロック Rosenbrock 法の並列化数値実験

本章では，移流拡散問題に対してブロック Rosenbrock 法を適用する並列化数値実験を行い，数値解の妥当性と並列化の効果を考察する．3.1, 3.2 節で示した移流拡散方程式のモデル問題に対して，第4章の特性曲線差分法を適用し，偏微分系を常微分系に記述し直す．この常微分方程式系は硬い系であるので，4.4.2 節のように A 安定な解法を適用するのが望ましい．このような場合，1 ステップごとの計算量の増大が課題であったが，第5章，第6章で構成した並列化向きの解法を適用することで速度向上が期待できる．そして，数値実験結果から数値解の安定性と並列化による速度向上度を考察する．ゆえに改めて，7.1 節でモデル問題の移流拡散方程式とその構成を，7.2 節で特性曲線法による離散化式を，7.3 節で常微分方程式系に適用する離散変数法を示す．そして，7.4 節で実施する数値実験は，第4～6章に倣って実施し，数値的収束性や速度向上度等の図や表で結果を示す．7.5 節では本章の考察を述べ，最終章の結論へ導く．

7.1 モデル問題

本節では，移流拡散方程式のモデル問題を示す．移流拡散方程式は時間方向1階，空間方向1階および2階の偏微分方程式の初期値境界値問題で以下のように表わされる．

$$a(x, t)u_t + b(x, t)u_x - \varepsilon u_{xx} = f(x, t), \quad (x, t) \in Q \quad (7.1)$$

ただし，境界条件および初期条件は，

$$u(0, t) = u(1, t) = 0, \quad t \in [0, T], \quad (7.2)$$

$$u(x, 0) = u_0(x), \quad x \in \Omega \quad (7.3)$$

と与えられる．ここで，区間 $\Omega = (0, 1)$ ，領域 $Q = \Omega \times (0, T)$ であり，また， a, b, f, u_0 は既知の関数で，以下のように定めるモデル問題を構成した．

$$\begin{aligned} f(x, t) &= d[-qx(1-x)\sin(qt) + b_0(1-2x)\sin(8\pi x)\cos(qt) + 2\varepsilon\cos(qt)], \\ a(x, t) &= 1.0, \quad b(x, t) = b_0\sin(8\pi x), \quad u_0(x) = dx(1-x), \\ d &= 100, \quad q = 0.24\pi, \quad b_0 = 0.05 \end{aligned} \quad (7.4)$$

また，拡散係数 ε は正の定数で，数値実験において $10^{-1} \sim 10^{-6}$ で適当に定めるとし，通常は， 10^{-3} とする．

このときの解析解は以下のとおりである．

$$u(x, t) = dx(1 - x) \cos(qt)$$

特性曲線に沿った解析解の3次元表示は図7.1のとおりである．図3.1と比較すると，格子点の位置が流れに沿う方向に変化していることがわかる．

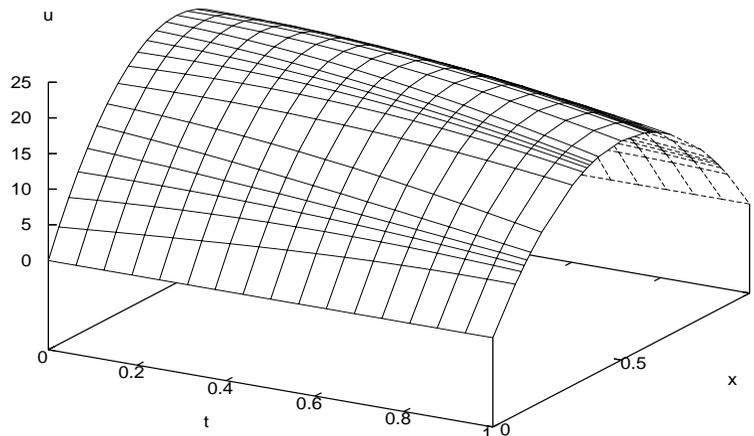


図 7.1: 特性曲線に沿う方向の解析解のプロフィール

7.2 特性曲線差分法の適用

移流拡散方程式の特性曲線差分法の詳細は4.1節において述べ，数値的収束性や安定性は4.2において解析した．問題(7.1)–(7.3)は，拡散係数が小さい場合，移流項から与えられる特性曲線に沿う保存系が自然であると考え，空間差分を適用してこの方向の連立常微分方程式の初期値問題に置き換える．これによって，問題は特性曲線の方程式を解く部分と数値解 U の連立常微分方程式を解く部分に分けられる．また，数値解の方程式系は変数係数線型系となる．

つまり，自然数 N に対して $h = 1/N$ と定め， $t = 0$ における区間 Ω を等間隔に分割する $i (i = 1, \dots, N - 1)$ 番目の特性曲線を常微分方程式

$$\frac{dx_i(t)}{dt} = \frac{b}{a}(x_i(t), t), \quad t > 0 \quad (7.5)$$

の解と定義する．ここで，問題 (7.1)–(7.3) は，式 (7.5) から得られる特性曲線 $x_i(t)$ ($i = 1, \dots, N-1$) に沿う方向の解 $u_i(t) \equiv u(x_i(t), t)$ を求めることに置き換えられ，さらに，空間方向の 2 階偏導関数 u_{xx} を不等間隔格子上的有限差分を用いて近似すると，

$$U'_i = a^{-1}(x_i(t), t) \left[\frac{\varepsilon}{h_{i+1}h_i} \{(1+q_{i+1})U_{i-1} - 2U_i + (1-q_{i+1})U_{i+1}\} + f(x_i(t), t) \right], \quad \text{for } i = 1, \dots, N-1 \quad (7.6)$$

のような変数係数線型常微分方程式系となる．

解析の結果，数値スキームの収束条件はやや厳しいものであるが，

$$\frac{\Delta t}{\Delta x^2} = \text{const.}$$

である．つまり，十分な収束性を得るためには，空間方向の分割数を 2 倍にするとき，時間方向の分割数を 4 倍にする必要がある．安定性の条件は，

$$\Delta t < \frac{\mathcal{M}h^2}{4\varepsilon}.$$

であるが，本章の手法のように A 安定な離散変数法を適用する場合， \mathcal{M} による制限がなくなる．

7.3 適用する離散変数法

移流拡散方程式の特性曲線差分法によって差分化された離散化方程式は以下の 2 つの常微分方程式系の大連立である．

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{b}}{\mathbf{a}}(\mathbf{x}, t) \quad (7.7)$$

$$\frac{d\mathbf{U}}{dt} = (DA)\mathbf{U} + \mathbf{F} = \mathbf{B}(\mathbf{x}, t)\mathbf{U} + \mathbf{F}(\mathbf{x}, t) \quad (7.8)$$

ここで，安定な解を得るために陰的 Runge-Kutta 法を下側の数値解 \mathbf{U} の常微分方程式 (7.8) に適用する手順を考える．

式 (7.8) は，変数係数ではあるが線型の常微分方程式系であるので， s 段の一般化陰的 Runge-Kutta 法

$$\begin{aligned} \mathbf{U}_{n+1} &= \mathbf{U}_n + \Delta t \sum_{i=1}^s \beta_i \mathbf{k}_i, \\ \mathbf{k}_i &= \mathbf{f} \left(\mathbf{x}(t_n + \gamma_i \Delta t), \mathbf{U}_n + \Delta t \sum_{j=1}^s \alpha_{ij} \mathbf{k}_j, t_n + \gamma_i \Delta t \right), \quad \text{for } i = 1, \dots, s \end{aligned}$$

を適用するとき，下段の段値 (stage value) k の計算過程である非線型方程式は，

$$\begin{bmatrix} I - \Delta t \alpha_{11} B_1 & -\Delta t \alpha_{12} B_1 & \cdots & -\Delta t \alpha_{1s} B_1 \\ -\Delta t \alpha_{21} B_2 & I - \Delta t \alpha_{22} B_2 & \cdots & -\Delta t \alpha_{2s} B_2 \\ \vdots & & \ddots & \vdots \\ -\Delta t \alpha_{s1} B_s & -\Delta t \alpha_{s2} B_s & \cdots & I - \Delta t \alpha_{ss} B_s \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_s \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_s \end{bmatrix}$$

のように線型となる．

ただし，変数係数に対応する時刻 t 依存のパラメタは，

$$\begin{aligned} B_i &= B(\mathbf{x}(t_n + C_i \Delta t), t_n + C_i \Delta t), \\ f_i &= f(\mathbf{x}(t_n + \gamma_i \Delta t), \mathbf{U}, t_n + \gamma_i \Delta t) \end{aligned}$$

であり，さらに $\mathbf{x}(t_n + C_i \Delta t)$, $\mathbf{x}(t_n + \gamma_i \Delta t)$ は特性曲線の方程式 (7.7) を解くことで求めなければならない．以後表記の利便性のため， $\{c_i\} = \{C_1, C_2, \dots, C_s, \gamma_1, \gamma_2, \dots, \gamma_s\}$ と表記する．ここでは，特性曲線の方程式は不安定要因を含まないものとし，スケールつき Runge-Kutta 法 ([30], pp.86-90) を用いることで， $0 < C_i, \gamma_i < 1$ に対応する $t_n + c_i \Delta t$ における数値解 $\mathbf{X}_i \simeq \mathbf{x}(t_n + c_i \Delta t)$ を求め，これを以後の計算に利用する．

つまり，ここでの手順は，まず，特性曲線を解くのに7段4次のスケールつき Runge-Kutta 法 (Fehlberg 公式に基づく) を適用して必要な段値 \mathbf{X}_i ($i = 1, 2, \dots, 8$) を計算する．その後数値解 \mathbf{U} の常微分方程式には，条件付き NA 安定な並列化向き解法である (2, 2, 2) 型4次ブロック Rosenbrock 公式 (bR224) を適用する．また，各段で解くべき線型方程式系は3重対角行列向きの直接法 (Gauss 消去法, Thomas アルゴリズム [28]) を適用する．時間刻み幅 Δt は Fehlberg 公式から自動調節可能であるが，ここでは単に固定値を用いる．

7.4 実験結果

数値実験は，前節で述べたブロック Rosenbrock 法を適用するアルゴリズムと，その比較として，従来の古典的 Runge-Kutta 法，2段4次 Butcher-Kuntzmann 公式，2段3次 ROW 型解法についても同様の実験を行う．

まず，7.4.1 節においては，収束条件，安定条件を満たす数値解の収束性を確認する．次に，7.4.2 節においては，数値的安定性の確認および逐次実行にかかる計算コストを評価する．最後に，7.4.3 節においては，3重対角向きアルゴリズムを採用するときの並列計算時間および速度向上の見積りをする．

7.4.1 収束性の数値的確認

$\varepsilon = 10^{-3}$ と定めるときの $t = 1.0$ における刻み幅 $\Delta x, \Delta t$ に対する最大絶対誤差を表 7.1 および図 7.2 に示した．この図表は収束性条件

$$\frac{\Delta t}{\Delta x^2} = \text{const.}$$

と同時に安定性条件も満たすときの数値解の収束性を示している．この結果から以下のことがわかる．

- 時間方向の収束次数は bR224, BK24, RK4 については $O(\Delta t^4)$ であり, ROW23 については $O(\Delta t^3)$ であった．これらのことは適用する数値積分法の収束性と一致することを示している．
- 数値実験は倍精度実数にて行ったため, 10^{-12} 程度の高精度の実験において丸め誤差に依存する誤差が確認される．
- ただし, 特性曲線方程式を陽的な 4 次公式を用いて解いているため, より高次の解法を数値解の常微分方程式に適用するときは, 特性曲線方程式の解法についてもその次数に相当する解法に変更する必要がある．

表 7.1: $t = 1.0, \varepsilon = 10^{-3}$ における最大絶対誤差

Method	Δt	Δx	$\max \text{error} $
bR224	1/16	1/25	0.2065E-05
	1/64	1/50	0.4401E-07
	1/256	1/100	0.2620E-09
	1/1024	1/200	0.1311E-11
	1/4096	1/400	0.9148E-13
RK4	1/16	1/25	0.1166E-05
	1/64	1/50	0.3241E-08
	1/256	1/100	0.1215E-10
	1/1024	1/200	0.7061E-13
	1/4096	1/400	0.4530E-13
BK24	1/16	1/25	0.7001E-06
	1/64	1/50	0.2838E-08
	1/256	1/100	0.1168E-10
	1/1024	1/200	0.4796E-13
	1/4096	1/400	0.8882E-13
ROW23	1/16	1/25	0.9899E-04
	1/64	1/50	0.2000E-05
	1/256	1/100	0.4245E-07
	1/1024	1/200	0.7488E-09
	1/4096	1/400	0.1213E-10

以上より, 収束性および安定性の条件を満たす場合, 数値解は $\Delta t \rightarrow 0$ のとき収束し, その次数は適用する離散変数法に依存することが確認できた．

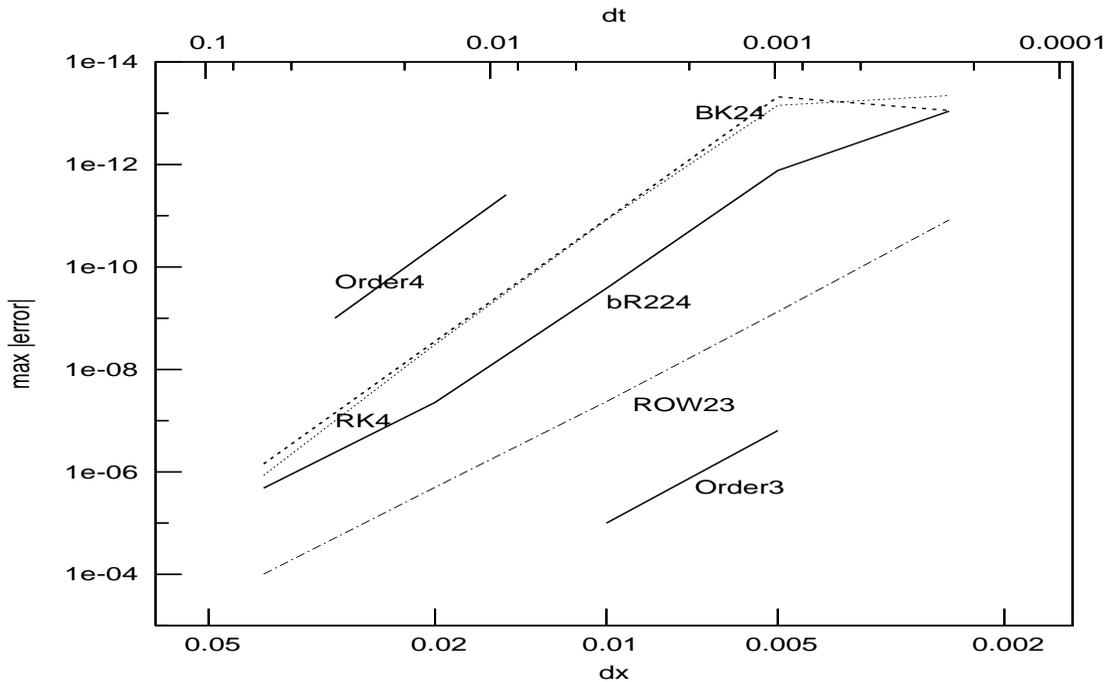


図 7.2: 数値的収束性の評価 ($t = 1.0$, $\varepsilon = 10^{-3}$)

7.4.2 安定性と逐次実行時間

一般に，十分な安定性を達成する数値解法の計算コストは増大する．本小節では，4 次の収束性を達成する bR224, BK24, RK4 の3つの解法による誤差の振る舞いを比較する．

まず，図 7.3 において， $t = 1.0$ における相対誤差の分布を示す．ただし，同程度の誤差の大きさとするために，各解法の $\Delta t, \Delta x$ の組は異なるものとなっている．誤差の分散が小さいのは，もっとも安定な解法である BK24 の適用である．陽的な解法である RK4 は，安定条件を満たさない場合，急激に誤差が増大するので注意が必要である．我々の解法である bR224 は，空間間隔が小さい部分で誤差が小さくなるという特徴が現われている．

次に，時間の増加にともなう相対誤差の増大を評価する．4章の結果より，陽的解法である RK4 には時間刻み幅に対する安定条件が定められる． $t = 1.340$ のとき，安定条件は $\Delta t < 0.0624$ となり， $t \geq 1.340$ の計算において，固定した時間刻み幅 $\Delta t = 1/16$ は安定条件を満たさない．図 7.4 において，RK4 の結果は安定条件を満たさない $t > 1.340$ の場合に急激に誤差が増大している様子が見られる．bR224 の結果は誤差の値は大きいですが，時間的な増加傾向はもっとも安定な BK24 とほぼ同等である．bR224 および BK24 は $t = 2.0$ 付近で相対誤差が増大しているが，これは，解析解が 0 に近づくためである．この実験についても，各解法が同程度の精度となるように，時間および空間の刻み幅を調節している．

さらに，安定条件を満たす数値テストの逐次実行時間を比較する．数値実験は，4.4.2 節と同様のものを bR224 についても実施し，その結果は表 7.2 である．bR224 は前章までの結果のとおり，数値解は安定であり，並列計算しない場合でも 3 重対角系の場合劇的に計算量を

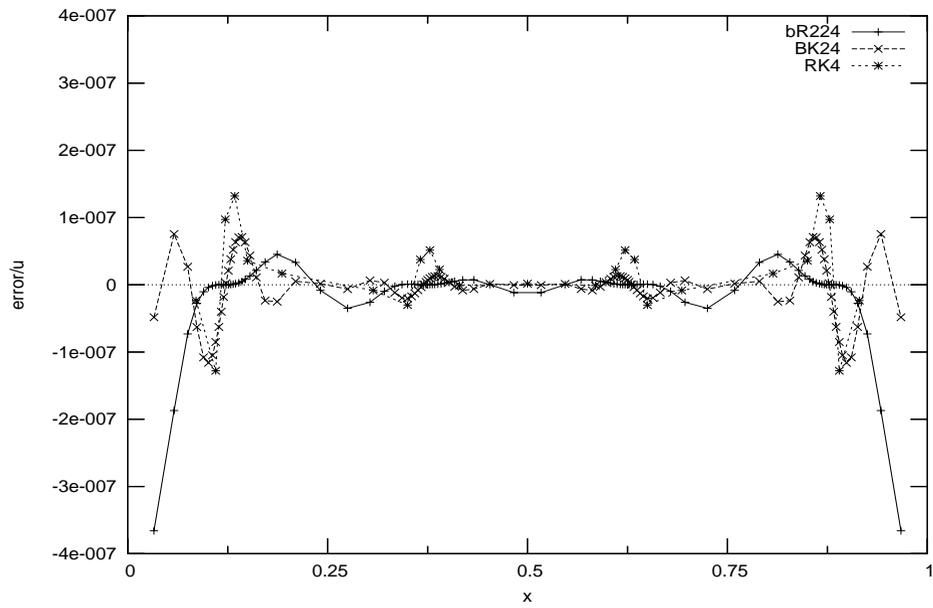


図 7.3: $t = 1.0$ における相対誤差の分布 (bR224: $\Delta t = 1/32, \Delta x = 1/100$, BK24: $\Delta t = 1/16, \Delta x = 1/100$, RK4: $\Delta t = 1/16, \Delta x = 1/25$)

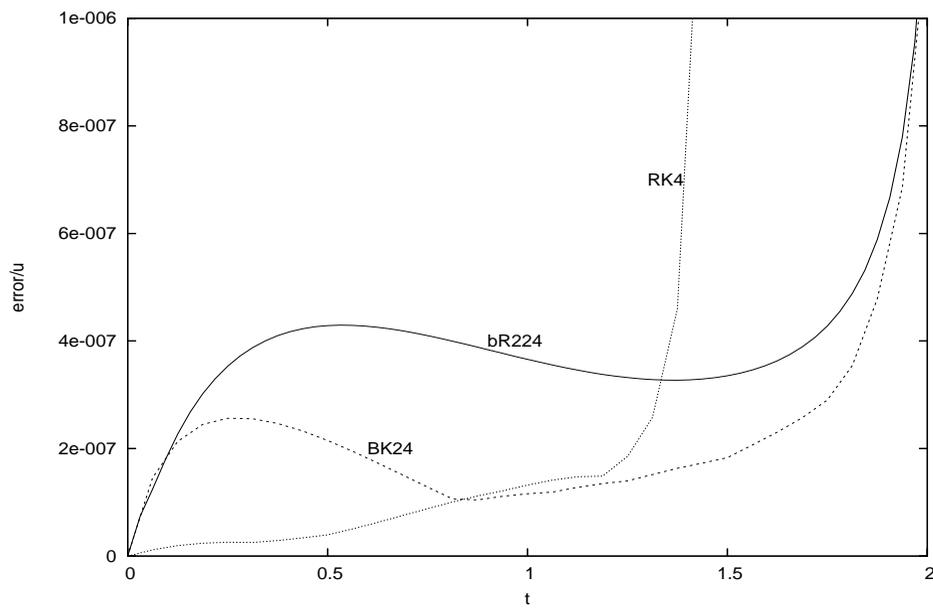


図 7.4: 数値的安定性の評価 ($t = 0 \sim 2.0$, $\varepsilon = 10^{-3}$, bR224: $\Delta t = 1/32, \Delta x = 1/100$, BK24: $\Delta t = 1/16, \Delta x = 1/100$, RK4: $\Delta t = 1/16, \Delta x = 1/25$)

減少させていることがわかる．特に no4 のような次元数が高いことによる不安定要因の場合，BK24 では計算量が増大するのに対して，bR224 はその影響を線形時間しか受けなくてすむ．

表 7.2: bR224 適用時の最大誤差と実行時間 (MM:SS.)

	no1	no2	no3	no4
ε	10^{-1}	10^{-3}	10^{-3}	10^{-3}
t_M	1.0	1.0	2.0	1.0
Δx	1/25	1/25	1/25	1/400
stiffness ratio	1.916E+3	1.916E+3	2.285E+4	7.834E+5
RK4-scalable				
Δt_0	2^{-4}	2^{-4}	2^{-4}	2^{-4}
Δt_M	8.634E-4	6.250E-2	1.897E-3	3.252E-4
total steps	427	16	68	1074
max error	0.3871E-09	0.1166E-05	0.2001E-06	0.5587E-12
CPU time	00:00.34	00:00.02	00:00.06	00:15.09
BK24				
Δt	2^{-4}	2^{-4}	2^{-4}	2^{-4}
total steps	16	16	16	16
max error	0.5745E-04	0.7002E-06	0.4607E-06	0.7757E-06
CPU time	00:00.12	00:00.12	00:00.23	05:51.65
bR224				
Δt	2^{-4}	2^{-4}	2^{-4}	2^{-4}
total steps	16	16	16	16
max error	0.5394E-02	0.2065E-05	0.2705E-05	0.1273E-04
CPU time	00:00.03	00:00.02	00:00.05	00:00.39

以上より，bR224 は応用問題である移流拡散方程式の特性曲線差分法に適用する場合でも，BK24 と同程度の安定性を保ち，さらに 3 重対角アルゴリズムを適用することで，計算量を軽減できるといえる．

7.4.3 並列実行時間と効果

最後に，bR224 適用時の並列化効果を評価する．数値実験は，それぞれの解法を適用し，同程度の誤差となる時間刻み幅による逐次および並列実行の経過時間を記録する．ただし，*RK4 は，安定な数値解を得るためには， $1.0E-6$ よりも十分に小さい誤差に抑えられている．

6.6 節における見積りで，3 重対角行列向きの解法を特性曲線差分法に適用する場合は，十分な並列化効果が得られないことがわかっている．これは，実験結果の表 7.3 に現われている．

る．問題のサイズを大きくすることによって，それぞれの解法の計算時間が増大するが，同等の精度を達成するのにもっとも高速なものは，bR224の1CPUである．その次がbR224の2CPUである．bR224は，1CPUに対する2CPUの速度向上度が0.8～0.94程度であり，並列化による速度向上は我々が利用するアーキテクチャでは達成できなかった．

表 7.3: 疎行列に対するアルゴリズム実行の経過時間 (秒)

d	max error	*RK4	BK24	ROW23	ROW23	bR224	bR224
		1-CPU	1-CPU	1-CPU	2-CPU	1-CPU	2-CPU
200	1.0E-6	0.14	1.29	0.08	0.11	0.04	0.05
400	1.0E-6	1.12	9.79	0.17	0.23	0.06	0.09
800	1.0E-6	8.96	107.56	0.32	0.31	0.13	0.14
1600	1.0E-6	72.16	1235.07	0.71	0.68	0.26	0.28
3200	1.0E-6	582.52	9890.31	1.46	1.35	0.53	0.57
6400	1.0E-6	4648.46	78824.23	2.96	2.69	1.07	1.14

7.5 本章のまとめ

本章では，ブロック Rosenbrock 法を移流拡散方程式の特性曲線差分法に適用する場合の収束性・安定性および並列計算による速度向上度を評価した．

第6章において構成した変数係数線型常微分方程式系に適用するブロック Rosenbrock 法の(2, 2, 2)型4次公式は，応用問題である移流拡散方程式の特性曲線差分法に適用する場合においても，従来の有効な解法である2段4次 Butcher-Kuntzmann 公式と同等の数値的収束性，安定性を示した．さらに，従来の解法の課題であった1ステップにおける計算量の増大についても，計算領域を分割すること，および，3重対角行列向きのアルゴリズムを適用することで大きく抑えることができた．期待していた並列計算については，逐次実行よりも実行時間がかかる結果となった．これは，並列化可能な計算部分の計算量のオーダが，逐次部分や通信部分と同等となってしまったためで，言い換えると，並列部分の逐次的なアルゴリズムの効率化が並列化の効果を上回ったということである．したがって，並列化可能な部分の計算量のオーダが他の部分よりも大きい場合は，依然として並列化が有効であることを示している．

第8章 結論

8.1 本論文のまとめ

本研究のまとめは、以下のとおりである。

- 1-1. 第3章で述べた“移流拡散方程式の特性曲線差分法”は、本研究の問題提起として位置づけられる。移流拡散方程式の特性曲線差分法とは、移流項優位の移流拡散方程式系について、まず、移流項に対応する特性曲線を表わす節点の常微分方程式を仮定し、数値解を計算する節点が特性曲線上を移動すると考える。そして、偏微分方程式系を特性曲線に沿う方向の常微分方程式系に変換し、さらに拡散項について節点上の数値解を用いて不等間隔中心差分で近似すると、解くべき方程式系は変数係数線型常微分方程式に置き換えられる。このようにして得られた、2つの常微分方程式系に対して、適当な離散変数法を適用するとした。
- 1-2. この解法は、流れに沿う方向に数値解を追跡するので、一般的な偏微分方程式系の数値解法適用時に見られる不自然な振動現象を軽減することが期待できるが、詳しい解析の結果、数値的収束性・安定性のためにやや厳しい制限があり、実際の数値実験のためには新たな工夫が必要となった。
- 1-3. 研究の当初では、2つに分けた常微分方程式系の相互依存性が小さいため、共有メモリ型の並列計算が可能で、適用する離散変数法は陽的な解法でよいと考えられた。しかし、先に挙げた安定条件のため、陽的な離散変数法はこの解法には適当でないことがわかった。
- 1-4. 2つの常微分方程式系のうち不安定要因が大きいのは、移流拡散方程式の数値解を求めるため差分された系とわかった。この系は、変数係数線型系であり、Butcher-Kuntzmann公式等のA安定な解法を適用するとき、大規模疎行列の線型方程式を解くことに帰着する。ゆえに、以後の研究は変数係数線型常微分方程式系の数値解法に焦点を当てて進めた。
- 2-1. まず考察した“陰的ROW型解法”は、条件付きNA安定であり、2CPUによる並列計算が可能である。並列計算による速度向上度は、次元数が200～400の常微分方程式系に対して2CPUを用いて1.91～1.96を示した。しかし、最高到達次数が3と不十分であったため、引き続き並列性を保ちつつ、収束性を向上する解法の構成が望まれた。

- 2-2. “ブロック Rosenbrock 法” は、陰的 ROW 型解法の一つの拡張版であり、条件付き NA 安定を達成する 4 次のスキームを構成できた。しかし、逐次計算部分が追加されたため、並列性は陰的 ROW 型解法に劣っている。ここで構成した $(2, 2, 2)$ 型スキームによる数値実験において、2CPU を用いる並列計算の速度向上度は、次元数が 200 ~ 400 の常微分方程式系に対して 1.72 ~ 1.81 を示した。
- 2-3. 陰的 ROW 型解法およびブロック Rosenbrock 法は、解くべき線型常微分方程式系が 3 重対角であるとき、各ブロックにおいて解くことになる線型方程式も 3 重対角となり、線型次数で計算可能な Thomas アルゴリズムが適用できる。これによって、計算時間は大いに軽減できる。
- 3-1. 以上を踏まえて、移流拡散方程式に対して特性曲線差分法を適用し、得られた変数係数線型常微分方程式系にブロック Rosenbrock 法を適用する並列化数値実験を行い、従来の解法と比較した。特性曲線差分法で出現する線型系の係数行列は 3 重対角であるため、空間・時間とも離散化された線型方程式系の解法として Thomas アルゴリズムが適用できた。
- 3-2. この数値実験において、数値解の誤差から収束性および安定性を評価し、逐次および並列の実経過時間から、従来の解法との比較と速度向上度の見積りをした。ここで従来の解法のうち、 AN 安定な解法として 2 段 4 次 Butcher-Kuntzmann 公式を、安定性よりも計算速度を意識した陽的解法として古典的 Runge-Kutta 法を採用した。
- 3-3. ブロック Rosenbrock 法を数値積分に用いる方法について、その収束性および安定性は、従来の解法と同程度であった。また、同程度の精度を得るための計算速度については、我々の方法が比較したすべての解法のうちで最速であった。ただし、並列計算の速度向上度については、逐次実行部分および CPU 間通信の増大のため、2CPU のほうが 1CPU の実行時間を上回る、速度向上が得られない結果となった。

以上より、特に変数係数線型常微分方程式系に適用することを目指した専用解法として、ブロック Rosenbrock 法の構成は成功したといえよう。そしてこの解法によって、十分な数値的収束性、安定性を保ち、かつ、領域分割や並列計算によって計算時間の軽減が可能となった。また、目標である移流拡散方程式の特性曲線差分法に適用し、応用問題においても有効であることを示した。

8.2 今後の課題

本研究でえられた成果を基礎に、さらに実用的な問題に応用することを意識すると、以下の事項が課題として挙げられる。

8.2.1 空間 2 次元の移流拡散問題

上記結論では、係数行列の疎性（3重対角性）を最大限活用し、劇的に計算量を軽減したため、かえって並列化効果が下がったが、空間次元を拡張すると、事情は変わる。

移流拡散問題を 2 次元

$$a(x, y, t)u_t + b(x, y, t)u_x + c(x, y, t)u_y - \varepsilon(u_{xx} + u_{yy}) = f(x, y, t), \quad (x, y, t) \in Q, \quad (8.1)$$

$$\text{for } t \in [0, T], \quad (x, y) \in \Omega \subset R^2, \quad Q = \Omega \times (0, T)$$

に拡張するとき、特性曲線方程式も同様に 2 次元に拡張され、

$$\frac{dx_{ij}(t)}{dt} = b(x_{ij}, y_{ij}, t), \quad (8.2)$$

$$\frac{dy_{ij}(t)}{dt} = c(x_{ij}, y_{ij}, t), \quad \text{for } i, j = 1, \dots, N-1 \quad (8.3)$$

と仮定できる。これを用いると、微分の連鎖側より、特性曲線に沿う方向の解は、

$$\frac{du_{ij}(t)}{dt} = a^{-1}(x_{ij}(t), y_{ij}(t), t)[\varepsilon\{u_{xx}(x_{ij}(t), y_{ij}(t), t) + u_{yy}(x_{ij}(t), y_{ij}(t), t)\} + f(x_{ij}(t), y_{ij}(t), t)], \quad (8.4)$$

$$u_{ij}(0) = u_0(x_{ij}, y_{ij}), \quad \text{for } i, j = 1, \dots, N-1 \quad (8.5)$$

となり、1次元のときと同様に、2階偏導函数についての中心差分近似

$$u_{xx}(x_{ij}(t), y_{ij}(t), t) \approx \frac{1}{h_{i+1,j}h_{ij}} [(1 + q_{i+1,j})U_{i-1,j} - 2U_{ij} + (1 - q_{i+1,j})U_{i+1,j}], \quad (8.6)$$

$$u_{yy}(x_{ij}(t), y_{ij}(t), t) \approx \frac{1}{k_{i,j+1}k_{ij}} [(1 + r_{i,j+1})U_{i,j-1} - 2U_{ij} + (1 - r_{i,j+1})U_{i,j+1}] \quad (8.7)$$

を適用することができる。ただし、

$$h_{ij} = \sqrt{\{x_{ij}(t) - x_{i-1,j}(t)\}^2 + \{y_{ij}(t) - y_{i-1,j}(t)\}^2},$$

$$k_{ij} = \sqrt{\{x_{ij}(t) - x_{i,j-1}(t)\}^2 + \{y_{ij}(t) - y_{i,j-1}(t)\}^2},$$

$$q_{i+1,j} = \frac{h_{i+1,j} - h_{ij}}{h_{i+1,j} + h_{ij}}, \quad r_{i,j+1} = \frac{k_{i,j+1} - k_{ij}}{k_{i,j+1} + k_{ij}}$$

である。以上の離散化を用いると、2次元の場合も、連立線型変数係数常微分方程式系が得られる。この系は、 $(N-1)^2$ 次元であり、さらの時間方向にブロック Rosenbrock 法を適用すると、 N に依存する帯行列の線型方程式系を解くことに帰着され、Thomas アルゴリズムはもはや適用できない。この系に対する Gauss 消去法の計算量のオーダは $O(N^6)$ となるため、ブロック Rosenbrock 法の並列化効果が再び期待できるようになると考えられる。

8.2.2 特性曲線の再構成

移流拡散方程式の特性曲線差分法の欠点は、特性曲線に沿った節点 (node point) の移動によって、節点間の距離の不均等が生じ、それが安定条件に影響することであった。数値解が十分滑らかであれば、不等間隔の数値解を補間して、等間隔の新たな節点上に置き換えることができる。中島 [34] は空間離散化のために多項式補間を用いることを提案し、我々は、特性曲線の再構成のために3次自然スプライン補間を用いることを試みた [9]。これは、本研究における安定な数値解法を用いる手法と並んで、安定条件の改善策となり得、これによって、離散変数法に陽的解法を用いる可能性と並列化可能性を検討することが考えられる。安定性の観点からは、この特性曲線の再構成は、格子点を追加するよりも、元の格子点数のまま、間隔を均等にするほうが有効である。また、特性曲線に関する条件

$$\Delta_h(t) : 0 = x_0(t) < x_1(t) < \dots < x_{N-1}(t) < x_N(t) = 1$$

を満たさない場合に対する解決策としても期待できる。

このスプライン補間を用いての特性曲線の再構成は、問題を2次元に拡張するときにも有効である。まず、1次元問題の場合と同様に、格子点が移動することによる安定条件の悪化を考慮しなければならない。また、我々の解法は、有限差分法のひとつであるので、その数値計算は構造格子上について行われる。つまり、連結する格子点が現象として自然でない場合、数値計算が有効でなくなる。そして、その可能性は、1次元の場合よりも大きいと考えられる。この場合、2次元スプラインを適用することになり、実用のためには、計算量についての見積もりが必要である。

8.2.3 近似 NA 安定性の妥当性

本研究において、変数係数線型常微分方程式系に特化して適用可能な離散変数法を構成し、その安定性解析のために新たに近似 NA 安定性を定義した。この近似 NA 安定性解析の妥当性については、安定性検証のためのテスト解法、および、この安定性解析に基づいたテスト方程式を構成し、その数値解の振る舞いをさらに検証する必要がある。最近の報告 [12] において、我々は、AN 安定とそうでない1段2次の一般化陰的 Runge-Kutta 法2個 (AN 安定版: GIRK12-1, 非 AN 安定版: GIRK12-2) を構成し、固有値を自在に変化させるテスト方程式に対する数値解を比較したところ、硬い系であることに起因すると考えられる誤差の増大が確認できた。表 8.1 に結果の概要を示す。この問題に対する詳細な研究も必要である。

表 8.1: NA 安定性妥当性テストの $t = 1.0$ における最大絶対誤差 ($h = 1/32$)

	GIRK12-1	GIRK12-2
安定テスト (固有値の実部 -10^3)	3.76E-05	1.51E-05
不安定テスト 1 (固有値の実部 $-10^3 \sim 0$)	6.95E-05	2.30E-03
不安定テスト 2 (固有値の実部 $-10^5 \sim -10^3$)	1.17E-04	2.11E-02

謝辞

本博士論文に関する研究の全過程を通じて、多大なるご指導とご助言を頂きました名古屋大学大学院情報科学研究科教授・三井斌友先生に深く感謝の意を表します。学部生時代から数えて本当に長きに渡るご指導の下、研究に対する取り組みは言うまでもなく、人生観にいたるまでを教えていただいたと心得ます。また、本論文をまとめるにあたり、有益なご助言を頂きました同じく助教授・小藤俊幸先生他数理系グループの諸先生方に厚く御礼申し上げます。

本研究の問題提起となる移流拡散方程式の特性曲線差分法を提案していただいた Prof. F.A. OLIVEIRA と Prof. F. PATRÍCIO に感謝申し上げます。また、研究の主要部分、変数係数線型常微分方程式系に適用する並列化向き数値解法については、Prof. O.A. KARAKASHIAN, Prof. W. RUST, 後藤彰先生の先行研究である定数係数線型常微分方程式系に適用する並列化向き陰的 Runge-Kutta 法に倣って進めることができました。深く感謝致します。

学生時代から変わらずご指導くださいました鳥居達生先生、南山大学教授・杉浦洋先生、名古屋大学教授・張紹良先生、岐阜聖徳学園大学助教授・齊藤善弘先生、愛知工科大学助教授・大迫尚行先生、九州工業大学助手・小守良雄先生、岐阜聖徳学園大学助教授・阿部邦美先生、神戸大学講師・影山康夫先生に感謝致します。今後ともご指導、ご鞭撻を頂きますよう宜しくお願いします。さらに、学生時代の研究生活を共に過ごし、互いに励ましあってきた名古屋大学工学部情報工学科鳥居研究室および名古屋大学大学院人間情報学研究科数理系グループの方々に感謝の意を表します。

豊田工業高等専門学校教職員の方々には、研究を進める上で格別のご配慮、ご指導を頂き、感謝申し上げます。赴任当時の校長・鬼頭幸生先生には、高専においては研究・教育だけでなく、校務雑用に至るまでが業務であることをご教示賜りました。また、学生気分の抜けぬ著者を叱咤激励し、卒業研究等で学生指導を切り盛りするのにご援助くださいました同校情報工学科教授・竹下鉄夫先生、同元教授・井口健先生には、重ねて深く感謝の意を表します。

参考文献

- [1] Bernstein, I.B., Brookshaw, L. and Fox, P.A.: A numerical method for solving systems of linear ordinary differential equations with rapidly oscillating solutions, *J. Comput. Phys.*, Vol. 98, pp.269–284 (1992).
- [2] Blanes, S., Casas, F. and Ros J.: Improved high order integrators based on the Magnus expansion, *BIT*, Vol. 40, pp.434–450 (2000).
- [3] Burrage, K.: *Parallel and Sequential Methods for Ordinary Differential Equations*, Oxford University Press, Oxford (1995).
- [4] Butcher, J.C.: *Numerical Methods for Ordinary Differential Equations*, John Wiley and Sons, Hoboken, NJ (2003).
- [5] Casas, F.: Solution of linear partial differential equations by Lie algebraic methods, *J. Comput. Appl. Math.*, Vol. 76, pp.159–170 (1996).
- [6] Casas, F.: Numerical integration methods for the double-bracket flow, *J. Comput. Appl. Math.*, Vol. 166, pp.477–495 (2004).
- [7] Dekker, K. and Verwer, J.G.: *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, Amsterdam (1984).
- [8] Douglas, J. and Russel, T. F.: Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite differences procedures. *SIAM J. Numer. Anal.*, Vol. 19, pp.871–885 (1982).
- [9] 江崎信行, 三井斌友: 移流拡散方程式の特性曲線差分法における特性曲線の再構成と並列化, 日本応用数理学会 1999 年度年会講演予稿集, pp.52–53 (1999) .
- [10] Esaki, N. and Mitsui, T.: Parallelizable ROW methods for linear variable-coefficient system of ODEs, *IPSJ Trans. High Perform. Computing Systems*, Vol. 42, No. SIG 9(HPS 3), pp.10–18 (2001).
- [11] Esaki, N. and Mitsui, T.: Parallelizable block Rosenbrock methods for linear variable-coefficient system of ODEs, *IPSJ Trans. Advanced Computing Systems*, Vol. 45, No. SIG 11(ACS 7), pp.290–302 (2004).

- [12] 江崎信行, 三井斌友: 変数係数線型系に適用する数値解法の NA 安定領域解析, 日本応用数学会 2005 年度年会講演予稿集, pp.144–145 (2005) .
- [13] Esaki, N., Mitsui, T., Oliveira, F.A. and Patrício, F.: Semi-discretization method along the characteristics for linear advection-diffusion equations, *the Southeast Asian Bull. of Math.*, Vol. 27, pp.813–827 (2004).
- [14] 福井義成, 野寺隆志, 久保田光一, 戸川隼人: 新 数値計算, 共立出版, (1999).
- [15] 藤井孝蔵: 流体力学の数値計算法, 東京大学出版会, (1994).
- [16] 藤間昌一, 田端正久: 特性曲線有限要素法に用いる数値積分について, 日本応用数学会 2004 年度年会講演予稿集, pp.288–289 (2004) .
- [17] Gear, C.W.: *The potential of parallelism in ordinary differential equations*, Tech. Rep. UIUCDCS-R-86-1246, Comp. Sci. Dept., Univ. of Illinois at Urbana-Champaign (1986).
- [18] 後藤彰, 三井斌友: 並列計算機向きの 3 段 4 次陰的 RK 法日本応用数学会論文誌, Vol. 1, No. 4, pp.291–304 (1991) .
- [19] Goto, A., Mitsui, T. and Sugiura, H: PIRK solver of ODES by parallel algorithm. In Natori, M. and Nodera, T., editors, *Parallel Processing for Scientific Computing*, Keio University (1990).
- [20] Hairer, E., Nørsett, S.P. and Wanner, G.: *Solving Ordinary Differential Equations I, Non-Stiff Problems*, 2nd edition, Springer-Verlag (1993).
- [21] Hairer, E. and Wanner, G.: *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Systems*, 2nd revised ed., Springer-Verlag (1996).
- [22] Harrison, G. W.: Numerical solution of the Fokker-Planck equation using moving finite elements. *Numer. Meth. for PDEs*, 4:219–232 (1988).
- [23] Hansbo, P.: A Crank-Nicolson type space-time finite element method for computing on moving meshes, *J. Comput. Phys.*, Vol. 159, pp.274–289 (2000).
- [24] 飯塚肇, 緑川博子 (訳): 並列プログラミング入門—ネットワーク結合 UNIX マシンによる並列処理—, 丸善 (2000).
- [25] Iserles, A.: Solving linear ordinary differential equations by exponentials of iterated commutators, *Numer. Math.*, Vol. 45, pp.183–199 (1984).
- [26] Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P. and Zanna, A.: Lie-group methods, *Acta Numerica*, Vol. 9, pp.215–365 (2000).

- [27] Karakashian, O.A. and Rust, W.: On the parallel implementation of implicit Runge-Kutta methods, *SIAM J. Sci. Stat. Comput.*, Vol. 9, pp.1085–1090 (1988).
- [28] 河村哲也: 流体解析 I, 朝倉書店 (1996).
- [29] Magnus, W.: On the exponential solution of differential equations for a linear operator, *Comm. Pure Appl. Math.*, Vol. 7, pp.649–673 (1954).
- [30] 三井斌友: 数値解析入門 —常微分方程式を中心に, 朝倉書店 (1985).
- [31] 三井斌友: 微分方程式の数値解法 I, 岩波書店 (1993).
- [32] 三井斌友: 保存系の計算数理論の総合的研究, 平成 11 年度 ~ 平成 13 年度日本学術振興会科学研究費補助金 (基盤研究 (A)(1)) 研究成果報告書 (2002).
- [33] 三井斌友, 小藤俊幸, 齊藤善弘: 微分方程式による計算科学入門, 共立出版 (2004).
- [34] 中島学: 移流拡散方程式の特性曲線に沿った差分法とその並列化の研究, 名古屋大学大学院工学研究科修士論文 (1994).
- [35] 名取亮: 線形計算, 朝倉書店 (1993).
- [36] 野津裕史, 田端正久: Navier-Stokes 方程式に対する特性有限要素法—時間刻み 1 次と 2 次精度の比較—, 第 19 回 数値流体力学シンポジウム講演プログラム, A3-4 (2005) .
- [37] Ohlberger, M. and Rohde, C.: Adaptive finite volume approximations for weakly coupled convection dominated parabolic systems, *IMA J. of Numer. Anal.*, Vol. 22, pp.253–280 (2002).
- [38] Pironneau, O.: *Finite Element Methods for Fluids*, Chapt. 3. John Wiley & Sons, Chichester, USA (1989).
- [39] Prothero, A. and Rosinson, A.: On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, *Math. Comp.*, Vol. 28, pp.145–162 (1974).
- [40] Rees, M. D. and Morton, K. W.: Moving point, particle and free-Lagrange methods for convection diffusion equations. Technical Report 89/1, Oxford University Computing Laboratory, Oxford (1989).
- [41] Rui, H. and Tabata, M.: A second order characteristic finite element scheme for convection-diffusion problems, *Numerische Mathematik*, Vol. 92, pp. 161–177 (2002).
- [42] 棚橋隆彦: はじめての CFD —移流拡散方程式—, コロナ社 (1996).

- [43] Tourigny, Y. and Süli, E.: The finite element method with nodes moving along the characteristics for convection-diffusion equations. *Numer. Math.*, 59:399–412 (1991).
- [44] Vreugdenhil, C. B. and Koren, B. (ed.): *Numerical Methods for Advection-Diffusion Problems*. Vieweg, Braunschweig, Germany (1993).
- [45] Wilkinson, B. and Allen, M.: *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, Prentice-Hall (1998).
- [46] 山口昌哉, 井上昭 (訳): *線型代数とその応用*, 産業図書 (1978).