

早稲田大学審査学位論文(博士)

非線形方程式の解の
精度保証付き数値計算に関する研究

1994年3月

柏木雅英

Masahide Kashiwagi

①

目次

非線形方程式の解の 精度保証付き数値計算に関する研究

1994年3月

早稲田大学大学院理工学研究科

柏木 雅英

Masahide Kashiwagi

目次

1 序論	7
1.1 背景	8
1.2 本論文の目的と概要	11
1.3 本論文の構成	15
2 区間解析	18
2.1 はじめに	19
2.2 区間演算	19
2.3 方程式の解の包み込み	23
2.4 むすび	26
3 計算機上における関数の表現法	28
3.1 はじめに	29
3.2 区間包囲による関数の表現	29
3.3 むすび	32
4 区間演算による近似解の包み込み	33
4.1 はじめに	34
4.2 区間演算による近似解の包み込み	34

4.3	アルゴリズムの妥当性	37
4.4	むすび	40
5	有理数演算による反復改良	41
5.1	はじめに	42
5.2	有理数演算による反復改良	42
5.3	むすび	48
6	精度保証システムの実現	49
6.1	はじめに	50
6.2	精度保証システム	50
6.3	精度保証システムの試作	51
6.4	数値例	53
6.4.1	数値例 1	53
6.4.2	数値例 2—区分線形系における“Lost Solution”の厳密解析—	55
6.5	むすび	59
7	構成的陰関数定理による非線形システムの数値解析	62
7.1	はじめに	63
7.2	占部の定理	63
7.3	構成的陰関数定理	69
7.4	ホモトピー法	77
7.5	予測子修正子法の改良	79
7.6	数値例	85
7.6.1	数値例 1	86
7.6.2	数値例 2	86

7.6.3 数値例 3	87
7.7 むすび	88
8 結論	93
謝辞	98
参考文献	100
関連業績	106

目 次

1 序	1
2 基礎	1
2.1 基礎理論	1
2.2 基礎理論の応用	1
2.3 基礎理論の発展	1
2.4 基礎理論の応用	1
2.5 基礎理論の発展	1
2.6 基礎理論の応用	1
2.7 基礎理論の発展	1
2.8 基礎理論の応用	1
2.9 基礎理論の発展	1
2.10 基礎理論の応用	1
2.11 基礎理論の発展	1
2.12 基礎理論の応用	1
2.13 基礎理論の発展	1
2.14 基礎理論の応用	1
2.15 基礎理論の発展	1
2.16 基礎理論の応用	1
2.17 基礎理論の発展	1
2.18 基礎理論の応用	1
2.19 基礎理論の発展	1
2.20 基礎理論の応用	1
2.21 基礎理論の発展	1
2.22 基礎理論の応用	1
2.23 基礎理論の発展	1
2.24 基礎理論の応用	1
2.25 基礎理論の発展	1
2.26 基礎理論の応用	1
2.27 基礎理論の発展	1
2.28 基礎理論の応用	1
2.29 基礎理論の発展	1
2.30 基礎理論の応用	1
2.31 基礎理論の発展	1
2.32 基礎理論の応用	1
2.33 基礎理論の発展	1
2.34 基礎理論の応用	1
2.35 基礎理論の発展	1
2.36 基礎理論の応用	1
2.37 基礎理論の発展	1
2.38 基礎理論の応用	1
2.39 基礎理論の発展	1
2.40 基礎理論の応用	1
2.41 基礎理論の発展	1
2.42 基礎理論の応用	1
2.43 基礎理論の発展	1
2.44 基礎理論の応用	1
2.45 基礎理論の発展	1
2.46 基礎理論の応用	1
2.47 基礎理論の発展	1
2.48 基礎理論の応用	1
2.49 基礎理論の発展	1
2.50 基礎理論の応用	1
2.51 基礎理論の発展	1
2.52 基礎理論の応用	1
2.53 基礎理論の発展	1
2.54 基礎理論の応用	1
2.55 基礎理論の発展	1
2.56 基礎理論の応用	1
2.57 基礎理論の発展	1
2.58 基礎理論の応用	1
2.59 基礎理論の発展	1
2.60 基礎理論の応用	1
2.61 基礎理論の発展	1
2.62 基礎理論の応用	1
2.63 基礎理論の発展	1
2.64 基礎理論の応用	1
2.65 基礎理論の発展	1
2.66 基礎理論の応用	1
2.67 基礎理論の発展	1
2.68 基礎理論の応用	1
2.69 基礎理論の発展	1
2.70 基礎理論の応用	1
2.71 基礎理論の発展	1
2.72 基礎理論の応用	1
2.73 基礎理論の発展	1
2.74 基礎理論の応用	1
2.75 基礎理論の発展	1
2.76 基礎理論の応用	1
2.77 基礎理論の発展	1
2.78 基礎理論の応用	1
2.79 基礎理論の発展	1
2.80 基礎理論の応用	1
2.81 基礎理論の発展	1
2.82 基礎理論の応用	1
2.83 基礎理論の発展	1
2.84 基礎理論の応用	1
2.85 基礎理論の発展	1
2.86 基礎理論の応用	1
2.87 基礎理論の発展	1
2.88 基礎理論の応用	1
2.89 基礎理論の発展	1
2.90 基礎理論の応用	1
2.91 基礎理論の発展	1
2.92 基礎理論の応用	1
2.93 基礎理論の発展	1
2.94 基礎理論の応用	1
2.95 基礎理論の発展	1
2.96 基礎理論の応用	1
2.97 基礎理論の発展	1
2.98 基礎理論の応用	1
2.99 基礎理論の発展	1
2.100 基礎理論の応用	1

目 次

2.1	Krawczyk の方法	26
3.1	連続な区間包囲	31
4.1	テスト区間 T の生成	35
5.1	区間反復 (I_k から I_{k+1} を生成)	44
6.1	Lost Solution	58
6.2	プログラム例	61
7.1	構成的陰関数定理	72
7.2	構成的陰関数定理の証明	77
7.3	解曲線追跡のアルゴリズム	82
7.4	数値例 1-1	86
7.5	数値例 1-2	87
7.6	数値例 1-3	88
7.7	数値例 2	89
7.8	解集合	90
7.9	数値例 3-1	90
7.10	数値例 3-2	91

7.11 数值例 3-3 91

7.12 数值例 3-4 92

表目次

4.1 数值例 3-1 (数值例 3-1)	91
4.2 数值例 3-2 (数值例 3-2)	92

表目次

第 1 章

6.1 区間反復 (小数表示)	56
6.2 区間反復 (有理数表示)	57

序論

本書は、コンピュータによる数値計算の基礎を解説している。この分野は、数値計算の基礎となる数値計算の分野であり、多くの分野で重要な役割を果たしている。その中でも、数値計算の基礎となる数値計算の分野は、多くの分野で重要な役割を果たしている。その中でも、数値計算の基礎となる数値計算の分野は、多くの分野で重要な役割を果たしている。その中でも、数値計算の基礎となる数値計算の分野は、多くの分野で重要な役割を果たしている。

第 1 章

序論

$$x^2 + y^2 = 1 \quad (1.1)$$

$$x^2 + y^2 = 1 \quad (1.2)$$

この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad (1.3)$$

この式は、単位円の方程式を示している。

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \quad (1.4)$$

この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。

この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。この式は、単位円の方程式を示している。

1.1 背景

現在の電子計算機による数値計算は、浮動小数点演算を基礎としている。この浮動小数点演算による計算には必ず誤差が伴い、電子計算機の誕生当初から問題点として認識されてきた。その結果、アルゴリズムの安定性等について多くの知見が得られ、多くの優れたアルゴリズムは誤差の影響がなるべく小さくなるように設計されている。しかし、実際の計算において丸め誤差の影響をすべて把握することは難しく、既存の数値計算法の大半は、計算結果の精度について確実な保証を与えることは出来ない。また、簡単な数値計算においてさえも、丸め誤差の影響によって有効数字がほとんど失われてしまうようなことは起こり得る。例えば、連立方程式

$$64919121x - 159018721y = 1 \quad (1.1)$$

$$41869520.5x - 102558961y = 0 \quad (1.2)$$

を倍精度浮動小数点数 (ANSI/IEEE Std. 754) を用いて Gauss の消去法で解くと、

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 237815437.895517 \\ 97087745.9667599 \end{pmatrix} \quad (1.3)$$

となるが、真の解は、

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 205117922 \\ 83739041 \end{pmatrix} \quad (1.4)$$

である。この例を見ても分かるように、整数演算でもない限り、広く行われている数値計算の結果は実は全く信用することが出来ないと言える。少々長くなるが、山本哲朗先生の言葉 [1] を引用しておく。

現在、我が国で数値計算を行っている人達は、計算機から出力された数値をいったい何桁まで正しい答としているのでしょうか。理論的に誤差 $O(h^2)$ であれば、 $h = 10^{-2}$ のとき、4桁あるいは安全をみて2桁正しいと信じるのでしょうか。知人の某教授によれば、具体的な問題では、数値をみれば、良い答えかどうか分かるものなのだそ

うです。もしそうならば、各種スプライン定規を用いてデータを外挿しあるいは内挿し、予測あるいは設計を行うのと同等の作業を、計算機を使って行っていることにならないでしょうか。各種計算センターで、適応型高性能解法と銘打たれたサブルーチンを使って、数字の列を得、それをもとに、何らかの結論が引き出されています。しかし、如何に高性能解法を用い、如何に多倍長演算を行っても、出力された数値が何処まで正しいかについては、整数計算でない限り、何の保証もないのです。考えてみれば誠に不思議なことではありませんか。我々は計算機という高価な道具を用い、キザミ幅を換え、苦勞した末に—しかも厳密な議論を展開した後に—、最後は神・仏と同様、出力された数値を正しいと信じることによってのみ救われているのです。数値計算は信仰なのでしょうか。

計算機自身に数値計算を行うと同時に結果の精度の保証を行わせるという発想とその研究は古くから行われてきた。精度保証付き数値計算と呼ばれるようになったこの分野は、近年急速な発展を見せ、今後の数値計算法のあるべき姿として世界的に注目を集めている [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

丸め誤差の追跡を行う上で現在主流になっているのは、区間演算と呼ばれる手法である [12, 13, 14, 15, 16, 17]. これは、すべての数値を [下限, 上限] というペアで表現し、区間と区間の演算で答えの区間を求めるもので、P. S. Dwyer によって提案され [12], R. E. Moore によって体系化された [13, 14]. 区間演算は一時注目を集めたものの、大規模な問題に対しては最終的に得られる区間の幅が広くなりすぎ、実用上は役に立たないと見られていた。ところがその欠点を補う手法が考案され、また不動点定理と組み合わせて方程式の解の存在保証を行うことも可能であることが分かり、再び注目を集めている。これらの研究では、ドイツ・Karlsruhe 大学の Kulisch を中心とする研究グループが中核となっている。

ところで、区間演算によって、数式で決定的に記述された量に対する数値計算の結果の精度保証、例えば

$$e^{\sqrt{163}\pi} - 262537412640768744 \quad (1.5)$$

を計算するといったことが行えることは明らかであろう(もちろん計算の過程に現れる関数に対する区間演算が行える場合)。ところが、方程式の解を求める場合はそうは言えない。線形方程式に対する Gauss の消去法などは解の計算プロセスは決定的であるが、非線形方程式、例えば

$$x_0^2 - x_1^2 - \sqrt{2} = 0 \quad (1.6)$$

$$\exp(x_0) - \frac{1}{x_1^3} = 0 \quad (1.7)$$

のような方程式に対しては一般に解を表現する数式は得られない。近年は数式処理技法が発達し、非線形方程式が厳密に解ける場合もないわけではないが、それは特殊な場合に限られる。そのため、非線形方程式に対しては一般に Newton 法などの反復法による近似解法が広く用いられ、この反復法を区間演算に置き換えても解の精度保証には何の役にも立たないことは明らかである(収束先の精度保証は出来ない)。すなわち、方程式の解の精度保証には単なる誤差の追跡とは異なった別のアプローチが必要である。ここで重要な役割を果たすのは、関数解析の分野で基本的なくつかの不動点定理であり、また Kantorovich をはじめとする Newton 法の収束定理 [18] である。方程式を同値な不動点問題に置き換え、不動点定理の成立を数値的に確かめることによって解の存在保証と誤差評価を行う。よってこの場合本質的な役割を果たすのは不動点定理であり、区間演算は存在条件の成立を厳密に確かめるために用いられる。

一方、これらの不動点定理の多くは Banach 空間などの空間上で議論されており、これらの定理を用いて有限次元の非線形方程式に留まらず無限次元の関数方程式の解の精度保証が行える可能性がある。微分方程式などの関数方程式を解くには、一般に差分近似などによって有限次元近似方程式を作り、その解として近似解を得る。計算機の丸め誤差を考えれば、有限次元方程式の場合でも結局近似解しか得ることは出来ないので、この点では無限次元も有限次元も事情は同じであるといえる。このような関数方程式に対する精度保証の試みは、Newton 法の収束定理の成立条件を数値的に確かめるものとして、占部 [19], Cesari [20], Kedem [21], Kaucher-Miranker [22, 23], 中尾 [3, 4], Plum [24], 大石 [25] などによる研究がある。また、Poincaré map の不動点の存在を

数値的に検証して周期解の存在を求めるものとして, Sinai[26], 山口-吉原-西田 [27, 6] らの研究がある. また, 高次の Taylor 展開と区間演算を用いて常微分方程式の解を包みこむ方法 [28, 29] もある.

また, 近年自動微分と呼ばれる手法が注目を集めている [30, 31, 8]. これは, 微分の chain-rule を用いて, プログラムの形で定義された関数の全ての偏導関数の値を関数値の計算と同時に計算するという手法である. 大規模な関数にも適用出来, 途中に分岐を含む関数にも適用出来る点が数式処理に対して有利な特徴である. 上で述べた精度保証技法は大規模な関数の厳密な微分を必要とするため, 精度保証技法の実用化には自動微分が不可欠である.

精度保証を行うためのソフトウェアの開発も行われている. 現在のところ, Karlsruhe 大学と IBM とで共同開発された ACRITH(High-Accuracy Arithmetic Subroutine Library, FORTRAN ライブラリ) [32, 9], FORTRAN-SC(FORTRAN for Scientific Computation, ACRITH を土台に FORTRAN を拡張) [33, 9], Karlsruhe 大学の研究グループによる PASCAL-SC(PASCAL for Scientific Computation, パソコン用精度保証ソフトウェア) [34, 10], PASCAL-XSC(A Pascal Extension for Scientific Computation), C-XSC(A C++ Class Library for Extended Scientific Computing) などが知られている. これらの Karlsruhe 大学によるソフトウェアは, 有効丸めを用いて両端に浮動小数点数を持つ区間演算を実現している. また, long accumulator を用いた高精度内積演算を実現することによって区間演算の無駄を軽減している点が特徴として挙げられる. また, operator-overloading 機構により, object 指向的なプログラミングが可能である. また, Aberth によるソフトウェア [35, 36, 37] もある. これは C++ 言語で記述され, 多倍長浮動小数点数による高精度演算を実現している.

1.2 本論文の目的と概要

本論文は, 前節で述べたような精度保証技法に関する現状を踏まえて, 特に非線形方程式の解の精度保証に関する研究成果をまとめたものである.

本論文の前半(第3-6章)では,有限次元の非線形方程式に対して,近似解を元にしたほぼ自動的な精度保証システムの構築に関する研究成果について述べる.

非線形方程式

$$f(x) = 0, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (1.8)$$

の解(零点)を求める場合,解析的に求まる場合は稀であり,計算機によって数値的に解を計算することが要求される.非線形方程式を数値的に解くためのアルゴリズムは数多く研究されているが,通常の浮動小数点演算は常に誤差を伴うため,その計算結果は誤差を含むものとなる.本論文では,従来の数値計算で求まった近似解を元にして,精度保証を行う方法を提案する.全ての計算を精度保証付き数値計算に置き換えるのではなく,従来法を補完して精度保証を行う方がより実用的であるという考えに基づいている.

非線形方程式の解を計算機上に表現する場合,無限精度の実数値を計算機上で得ることは不可能であるが,計算機上に実現されたプログラムによって,“任意に高精度の値が得られる”状態を実現することは可能である.例えば,計算機で円周率の値を無限桁求めることは言うまでもなく不可能である.しかし,任意に高精度の値を求められるようなアルゴリズムを我々は既に知っており,それを計算機上に実現できる.そのため我々は円周率の値を“知っている”と認識できると考えられ,方程式の解についても円周率と同様の状態になって初めて“精度保証”されたと考える.

ここでは,“ $a_i \leq x \leq b_i$, $b_i - a_i \rightarrow 0$ ”を満たす列 $\{a_i, b_i\}$ を計算するプログラムが計算機上に実現されたとき,実数 x は精度保証されたと考えることにする.すなわち,方程式の解が“精度保証された”ということは,区間 $[a_i, b_i]$ に必ず真の解が含まれていることの保証と,いくらでも精度の良い解を得ることが出来ることの保証という二重の意味を持っている.

注意 1.1 このようなプログラムを,その実数の計算機上の表現と考えることもできる.このようなプログラムが作れる実数は計算可能実数と呼ばれ,構成的実数論 [40] と深い関連がある. □

方程式の解についてこのようなプログラムが得られたとき, $[a_0, b_0]$ を計算すればそれが真の解を

含む区間解であり，区間幅が誤差評価となる．誤差が大きすぎる場合は， $[a_1, b_1], [a_2, b_2] \dots$ と順に計算していけば，いくらでも精度の高い解が得られる．すなわち，品質保証のみならずアフターサービスも万全という訳である．1変数の方程式の場合このようなプログラムはいわゆる二分法によって容易に実現できる（但し，後述する方程式の表現誤差の問題を考えるとそう簡単ではない）．2変数以上の場合，各成分について解を包含するような縮小区間列を計算することになるが，これは簡単ではなく，縮小写像原理などの不動点定理を用いる必要がある．

上の意味での精度保証を行う上で，現状の数値解析法を考えると次の二つの問題点が浮上してくる．

一つは，計算に用いる数値表現体系である．現状の計算機の多くは，実数の計算機上の表現として浮動小数点方式を採用している．この方法は，メモリ上に占める領域が一定で絶対値の大きい数値が扱えるという利点があり，ハードウェア化することにより極めて高速な演算が実現されている．ところが，表現出来る精度には限界があるため，任意に精度を改良するような目的には当然使用出来ない．本論文では，精度保証に適した数値表現体系として，分母分子を多倍長の整数として持つ有理数表現を採用する．この方法では，加減乗除において誤差が混入しないため誤差のない線形計算を仮定でき，精度保証のための理論が作りやすい．計算が進むにつれて桁の爆発が起こりやすいという欠点はあるが，アルゴリズムに適切な丸めを入れることによってそれを防ぐ工夫をする．実際，有理数に対して，連分数展開の手法を用いることにより誤差の大きさと方向を指定した丸めを効果的に行うことが出来る．更に，区間演算では計算途中の任意の時点で丸め（両端の数を外側へ丸める）が行えるため，有理数演算を行った場合の速度の低下を比較的抑え易い．

二つ目の問題点は，解析の対象となる非線形方程式の計算機上の表現方法，すなわち方程式を $f(x) = 0$ と記述したときの関数 $f: R^n \rightarrow R^n$ を計算機上にどう表現するかという問題である．関数の表現が誤差を含んでいれば当然精度保証は出来ない．有理数演算を導入すれば加減乗除だけで構成されている関数は厳密に表現できるが， $\sqrt{\quad}$ ， \log などの関数が含まれていれば厳密な表

現はできない。すなわち、このような場合は誤差評価付きで関数が表現されている必要があるが、これは区間演算の技法を用いることにより容易に実現出来る。更に、精度保証された解の反復改良の可能性を保証するためには、任意に高精度に関数値が計算できる必要がある。区間演算では、加減乗除や超越関数の計算において、区間の両端の数を外側に丸めながら計算を行うため、丸めの大きさに応じた幅の結果が得られる。本論文では、関数に入力された区間の幅をこの丸めの大きさの基準と考え、幅が狭いほど高精度な演算を行うような方法を提案し、これによって方程式を表現することにする。

また、それらとは別の問題点として、アルゴリズムの実現容易性がある。精度保証付き数値計算が実用になるためには、その利用が比較的容易である必要がある。また、その手法を利用するために事前に多くの検討を行う必要があれば、その段階での間違いの混入が防げなくなり、計算結果の信頼性の低下を招くことになる。すなわち、実用的な精度保証アルゴリズムは、方程式の記述が容易で、またなるべく自動的に解析を行うようなシステムが構築可能でなければならない。

本論文では、上で述べたような数値表現及び関数表現を仮定し、適当な方法によって得られた近似解に対して、精度保証を行うアルゴリズムを提案する。これは、区間演算と平均値の定理、不動点定理を巧みに組み合わせた Krawczyk の方法 [38, 39, 14] を基礎とし、またシステム化を十分に念頭においた方法である。解の包み込みを利用するものである。また、先に述べた目標のように、与えられた近似解を元にしてその誤差評価を行い、また任意に精度を高められるような反復改良の手段を与える。また、十分良い近似解に対して解の包み込みに成功する事、包み込みに成功した場合に任意に精度を高められる事を証明し、理論の妥当性を示す。また、それに有理数演算と自動微分を組み合わせた精度保証システムの構築法を述べる。更に、実際の試作と幾つかの例題に対する精度保証を行いその有用性を示す。

ここで示すような、任意精度までの反復改良を始めとした真に数値の精度について細部まで考慮して構築されたアルゴリズムは今まで知られておらず、また当然それを実現したソフトウェアも存在しない。

本論文の後半では，方程式に幾つかのパラメータが付加された場合など，定義域の次元が値域の次元よりも高いような方程式の解の包み込みについて論ずる．この次元の差が n ならば解の集合は一般に n 次元多様体となるが，これを包みこむ事を考える．これは，パラメータによる分岐現象等の解析や変動が加わった方程式の解析など，様々な非線形現象の解析の基礎となる．従来，通常方程式の解（0次元多様体）の包み込みの方法は盛んに研究されてきたが1次元多様体以上の解の包み込みの方法についてはあまり議論されていなかった．

ここでは，陰関数定理を拡張し，その成立範囲を見積もれるような構成的陰関数定理を導く．これは，簡易 Newton 法の収束定理である占部の定理 [19] を拡張する形で証明される．陰関数定理は，局所的な陰関数の存在を示すものであるが，その具体的な成立範囲を与えてはいない．陰関数定理の証明方法の一つとして縮小写像原理によるものが知られているが，Newton 法も同様に縮小写像原理によって支配されている．Newton 法の収束定理が縮小写像原理等の不動点定理によって証明され，その唯一解領域を特定出来る以上，陰関数定理においても同様に成立範囲を特定し，またその範囲での陰関数の計算方法を与えるようなものができるのは極めて自然なことである．

また，ここで導いた定理の応用例の一つとして，ホモトピー法 [41, 42, 43, 44] の解曲線の新しい追跡アルゴリズムを示す．また，その有効性をシミュレーションによって示す．ホモトピー法の解曲線の追跡法としては，安定性を高めるための様々な工夫は数多く知られているが，ここで示した方法は収束性を完全に保証しつつ追跡するような方法で，全く新しいものである．

1.3 本論文の構成

以下，本論文の構成を述べる．

第2章で，本論文の議論の基礎となる区間解析について説明する．まず，区間解析で基本となる区間演算をその用語とともに定義する．次に，区間演算を用いた非線形方程式の解の精度保証技法の基本的かつ代表的なものとして，Krawczyk の区間写像を用いた方法を説明する．

第3-6章では、有限次元の非線形方程式に対して、適当な方法で得られた近似解を元にして精度保証を行う方法について議論する。

まず第3章では、厳密な精度保証のために不可欠な、方程式の計算機上での表現方法について議論する。実数値連続関数を計算機上に厳密に情報落ちすることなく載せることを目的とする。そのために、従来の浮動小数点数に代わって、有理数演算を採用する。更に、計算誤差の影響を把握出来るように区間演算を採用する。また、こうして定義された区間写像から元の関数についての情報を完全に取り出せる(元の関数の定義となっている)ことを保証するための出来る限り緩くかつ確認しやすい条件として、連続な区間包囲の概念を提案する。

第4章では、Krawczykの方法を元として、適当な方法で得られた近似解を用いる精度保証アルゴリズムを提案する。Krawczykの方法は、与えられた領域に対してそこに解があるかどうかを確認するものである。よって、本論文で対象とするような与えられた近似解に対してその近くにあるであろう真の解を包みこむような目的の場合には、近似解を元にして真の解を含みそうな適当な領域を与えてやる必要があった。本章では、この領域の決定法を示す。そしてこの決定法を元にし、また第3章で提案したような関数表現を仮定して、与えられた近似解を用いた真解の包み込みアルゴリズムを示す。また、このアルゴリズムが十分良い近似解に対して有効である事を示す。

第5章では、任意精度までの反復改良を行うアルゴリズムを提案する。仮に無限精度の区間演算が行えたとすれば任意精度までの反復改良はKrawczykの方法により容易に行える。しかし、実際にはそれは不可能である。ここでは、第3章で提案した関数表現を仮定し、また有理数演算を用いた区間反復法を提案する。この方法では、適切に制御された有理数の丸めによって、有理数の桁数の爆発の防止と任意精度までの反復改良を同時に達成している。また、それが正しく動く事を証明する。

第6章では、精度保証システムの実際の実現について議論する。まず、本論文(第3-5章)で提案した方法を実現するために必要なソフトウェア、プログラミング技法について議論する。次に、

本論文で提案したアルゴリズムの実際の試作について説明する。自動微分の技法を用い、方程式と近似解の2つの入力だけで自動的に精度保証を行うシステムの構築が出来た。最後に、幾つかの例題に対する精度保証を行いその有用性を示す。

第7章では、方程式に幾つかのパラメータが付加された場合など、定義域の次元が値域の次元よりも高いような方程式の解の包み込みについて論ずる。

まず、これ以降の議論の基礎となる、簡易 Newton 法の収束定理である占部の定理について議論する。次に、陰関数定理を拡張し、その成立範囲を見積もれるような構成的陰関数定理を占部の定理を元に導く。更に、導関数の Lipschitz 連続性を仮定し、同様な定理を導く。

また、構成的陰関数定理の応用例の一つとして、ホモトピー法の解曲線追跡の新しい方法を提案する。まず、ホモトピー法について簡単に説明する。次に、解曲線追跡法の一つである予測子修正子法の改良として、決して失敗しないような新しい解曲線追跡法を構成的陰関数定理から導く。また、その解曲線追跡法について実際の適用に関する詳細な議論を展開する。最後に、シミュレーションによってその有効性を示す。

第8章は本論文の結論に充てられる。

2.1 はじめに

本論文の議論の基礎となる区間解析について説明する。まず、区間解析で基本となる区間演算をその用語とともに定義する。区間演算で用いられる用語は論文によってまちまちで決定版はまだなく、本論文でも独自のものをを用いている。次に、区間演算を用いた非線形方程式の解の精度保証技法の基本的かつ代表的なものとして、Krawczyk の区間写像を用いた方法 [38, 39, 14] を説明する。これは、区間演算と平均値の定理、不動点定理を巧みに組み合わせた方法で、第3章以降で議論される精度保証アルゴリズムの基礎となるものである。

2.2 区間演算

本節では、本論文で用いる区間演算の用語について説明する。区間演算は、計算機で実数が厳密に表現できないため、計算機内での数値をある幅を持った区間として扱い、真値の取りうる全ての値を区間内に包含しながら計算を進めようという考えから生まれた。

定義 2.1 (区間, 区間ベクトル, 区間行列) $x \leq y$ を満たす実数 x, y で定まる R の部分集合 $\{r \mid x \leq r \leq y\}$ を区間といい, $[x, y]$ と表す。また, 成分として区間を持つようなベクトル, 行列をそれぞれ区間ベクトル, 区間行列という。 n 次元区間ベクトル $([x_1, y_1], \dots, [x_n, y_n])^t$ は, ベクトル $x = (x_1, \dots, x_n)^t$, ベクトル $y = (y_1, \dots, y_n)^t$ を用いて $[x, y]$ と表す。 $n \times m$ 区間行列 $([a_{11}, b_{11}], \dots, [a_{nm}, b_{nm}])$ についても同様。また, これらの区間ベクトル, 区間行列は次のような部分集合を与える。

$$[x, y] = \{z = (z_1, \dots, z_n)^t \mid z_i \in [x_i, y_i]\} \quad (2.1)$$

$$[a, b] = \{c = (c_{11}, \dots, c_{nm}) \mid c_{ij} \in [a_{ij}, b_{ij}]\} \quad (2.2)$$

□

D を R, R^n または $\mathcal{L}(R^m; R^n)$ ($n \times m$ 行列の作る線形空間) の部分集合としたとき, D に含まれるような区間, 区間ベクトル, 区間行列の集合を $I(D)$ で表す.

区間 $I = [x, y] \in I(R)$ の中心, 半径, 絶対値を

$$\text{mid}(I) = \frac{x+y}{2} \quad (2.3)$$

$$\text{rad}(I) = \frac{y-x}{2} \quad (2.4)$$

$$|I| = \max(|x|, |y|) \quad (2.5)$$

で定義する. また, 区間ベクトル, 区間行列 I の中心, 半径, 絶対値も,

$\text{mid}(I)$ (各成分の中心によるベクトル, または行列)

$\text{rad}(I)$ (各成分の半径による $\prime\prime$)

$|I|$ (各成分の絶対値による $\prime\prime$)

で定義する. 例えば, 区間行列 I に対して $\text{rad}(I)$ は, I の各成分の半径を成分として持つような非区間行列である.

区間間の演算は, 通常演算 $* \in \{+, -, \times, /\}$ に対して

$$I_1 * I_2 = \{x_1 * x_2 | x_1 \in I_1, x_2 \in I_2\} \quad (2.6)$$

となるように定める. 具体的には, 区間 $X = [a, b], Y = [c, d]$ に対して,

$$X + Y = [a + c, b + d] \quad (2.7)$$

$$X - Y = [a - d, b - c] \quad (2.8)$$

$$X \times Y = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \quad (2.9)$$

$$X/Y = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)] \quad (\text{if } Y \not\equiv 0) \quad (2.10)$$

のようになる. また, 区間ベクトル, 区間行列に対する演算は通常演算順序で各成分に対して実数同士の区間演算を行うことにより定める.

注意 2.1 ここで定めた区間同士の演算方法は、演算結果が実際の像そのものになっている。これは、有理数演算などの丸めのない演算を用いれば可能であるが、当然有限精度の浮動小数点数では区間の両端の数を計算するときに丸め誤差が入ってしまうので、ここで述べたような演算は実現できない。このため、実際には丸め誤差の発生する方向(切り上げまたは切り捨て)を区間の外側向きに制御することによって演算を行う。このときの演算は後述する区間包囲(演算結果は像そのものでなく像を含む)となる。 □

注意 2.2 区間ベクトルや区間行列の演算では、有理数演算を使ったとしても、一般には演算結果として像そのものを得ることは出来ない。例えば演算結果がベクトルとなるような場合を考えると、区間ベクトルは空間 R^n の長方形の形をした集合しか表現出来ないが、一般に行列演算による区間の像はそうはならない。 □

また、ベクトル、行列の半順序を次のように定義する。

定義 2.2 (ベクトル、行列の半順序) ベクトル $x, y \in R^n$, $x = (x_1, \dots, x_n)^t$, $y = (y_1, \dots, y_n)^t$, 行列 $a, b \in \mathcal{L}(R^m; R^n)$, $a = (a_{11}, \dots, a_{nm})$, $b = (b_{11}, \dots, b_{nm})$ の半順序をそれぞれ $\omega \in \{\leq, <, \geq, >\}$ として

$$x \omega y \Leftrightarrow x_i \omega y_i \quad \text{for all } i \quad (2.11)$$

$$a \omega b \Leftrightarrow a_{ij} \omega b_{ij} \quad \text{for all } i, j \quad (2.12)$$

で定める。 □

区間ベクトル及び区間行列のノルムは、最大値ノルムを一般化した次のノルムで定める。

定義 2.3 (scaled maximum norm) 与えられた scaling vector $u = (u_1, \dots, u_n)^t \in R^n$, $u > 0$ に対して、区間ベクトル $I \in I(R^n)$ のノルムを

$$\|I\|_u = \max\{|I_i|/u_i \mid \text{for all } i\} \quad (2.13)$$

で、区間行列 $A \in I(\mathcal{L}(\mathbf{R}^n; \mathbf{R}^n))$ のノルムを

$$\begin{aligned}\|A\|_u &= \max\{\|AI\|_u \mid \|I\|_u = 1\} \\ &= \| |A|u \|_u\end{aligned}\tag{2.14}$$

で定める. □

$u = (1, 1, \dots, 1)^t$ のときは、通常 of 最大値ノルムになる.

通常 of $D \subset X = \mathbf{R}^n$ から $Y = \mathbf{R}^m$ への写像 f をもとに構成された $I(D)$ から $I(Y)$ への写像を一般に区間写像という.

定義 2.4 (区間包囲, 区間拡張) $X = \mathbf{R}^n$, $Y = \mathbf{R}^m$, $D \subset X$ とする. 区間写像 $F: I(D) \rightarrow I(Y)$ が $f: D \rightarrow Y$ の区間包囲 (interval enclosure) であるとは,

$$F(I) \supset f(I) \quad \text{for all } I \in I(D)\tag{2.15}$$

が成立することをいう. また, F が f の区間拡張 (interval extension) であるとは, F が f の区間包囲で,

$$F([x, x]) = f(x) \quad \text{for all } x \in D\tag{2.16}$$

が成立することをいう. □

区間包囲は, 弱区間拡張 (weak interval extension) とも呼ばれる. 区間演算の意義を考えると, 一般に区間写像は区間包囲であるように作られるべきである. 加減乗除などの二項演算, \sin , \log などの単項演算の区間包囲は計算機上で容易に実現できる. 従って, これらの基本演算の組み合わせで記述された関数の区間包囲も容易に実現できる. しかし, 有限桁の浮動小数点演算を用いる限り, 厳密な意味での区間拡張の実現は不可能である.

単純に全ての演算を区間演算に置き換えて区間包囲を構成するより効率のよい方法として, 次の方法が知られている.

定義 2.5 (mean value form) $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$, $D \subset X$, $m : I(D) \rightarrow Y$ を $m(I) \in I$ を満たす任意の関数とする. $f : D \rightarrow Y$ に対して,

$$c = m(I) \tag{2.17}$$

$$F(I) = f(c) + F'(I)(I - c) \tag{2.18}$$

で定義された区間写像 $F : I(D) \rightarrow I(Y)$ を f の generalized mean value form という. 特に, $m(I) = \text{mid}(I)$ である場合, mean value form という. ただし, F' は f' の区間包囲であるとする. □

これが f の区間包囲となることが平均値の定理により証明出来る. また, これは, 次節の方法で重要な役割を果たす.

2.3 方程式の解の包み込み

非線形方程式 $f(x) = 0$ の近似解に対して精度保証を行うことを考える. 写像 f が, 数学における写像の定義の通り点を与えると点を返すような単なる対応として計算機上に表現されていたならば, 精度保証はほとんど不可能である. f がそのような black box ならば, ある近似解における残差がどんなに小さかったとしてもその近くに真の解が存在する保証にはならない. ある点における情報がそのすぐ近くの点についての情報を何一つ与えない, まさに“一寸先は闇”といった状態である. よって, ただの一点でなくある領域についての情報が必要となるが, そこで写像 f による入力区間の像を容易に包み込める区間包囲を用いる. ここでの区間演算の役割は, もはや数値誤差の包み込みという消極的なものでなく, 区間の像を容易に包み込めるという性質を積極的に利用したものである. この場合, 扱われる区間は微小区間でなく, ある程度の幅を持った区間となる.

Krawczyk の区間写像を用いる方法 [38, 39, 14] は, このような区間包囲による精度保証の方法の代表的なものである. これは, 導関数 f' の区間包囲を利用し, 簡易 Newton 反復に対して縮小

写像原理の成立を確かめる方法である。導関数 f' の区間包囲を構成するためには f' のプログラムによる表現が必要になるが、 f がプログラムで書かれている場合には、自動微分の技法 [30, 31, 8] を用いることにより容易に f' の区間包囲を構成できる。

Krawczyk の区間写像を用いた解の包み込みを以下に示す。

定理 2.1 (Krawczyk の区間写像による解の包み込み) [38, 39, 14, 17] $U \subset \mathbb{R}^n$, $f : U \rightarrow \mathbb{R}^n$ を C^1 級とする。ある区間 $T \in I(U)$ について、区間行列 M を

$$M = E - L^{-1}F'(T) \quad (2.19)$$

で、区間写像 $K : I(T) \rightarrow I(\mathbb{R}^n)$ を

$$K(I) = c - L^{-1}f(c) + M(I - c) \quad (2.20)$$

$$c = \text{mid}(I) \quad (2.21)$$

で定義する。ただし、 E は単位行列、 L は (T における微分の近似である) 正則な非区間行列、区間写像 F' を導関数 f' の区間包囲とする。

ここで、

$$K(T) \subset T \quad (2.22)$$

$$\|M\| < 1 \quad (2.23)$$

が満たされるならば、以下が成立する。

(1) T に方程式 $f(x) = 0$ の解 x^* が唯一存在する。

(2) 区間反復を

$$I_0 = T \quad (2.24)$$

$$I_{k+1} = I_k \cap K(I_k) \quad (2.25)$$

で定義すると,

$$x^* \in I_k \quad \text{for all } k, \quad (2.26)$$

$$\|\text{rad}(I_{k+1})\| \leq \|M\| \cdot \|\text{rad}(I_k)\| \quad (2.27)$$

が成立する.

(3) $\forall x_0 \in T$ から出発する簡易 Newton 反復

$$x_{k+1} = x_k - L^{-1}f(x_k) \quad (2.28)$$

に対して,

$$x_k \in T \quad \text{for all } k, \quad (2.29)$$

$$x_k \rightarrow x^* \quad \text{as } k \rightarrow \infty, \quad (2.30)$$

$$\|x_k - x^*\| \leq \frac{\|L^{-1}f(x_k)\|}{1 - \|M\|} \quad (2.31)$$

$$\|x_k - x^*\| \leq \frac{\|M\|^k \|L^{-1}f(x_0)\|}{1 - \|M\|} \quad (2.32)$$

が成立する.

□

解が包み込まれたときの様子を図 2.1 に示す.

この定理は, 簡易 Newton 反復を定義する写像を

$$g(x) = x - L^{-1}f(x) \quad (2.33)$$

とすると区間写像 K , 区間行列 M が,

$$K(I) \supset \{g(x) \mid x \in I\} \quad \text{if } I \subset T \quad (2.34)$$

$$M \supset \{g'(x) \mid x \in T\} \quad (2.35)$$

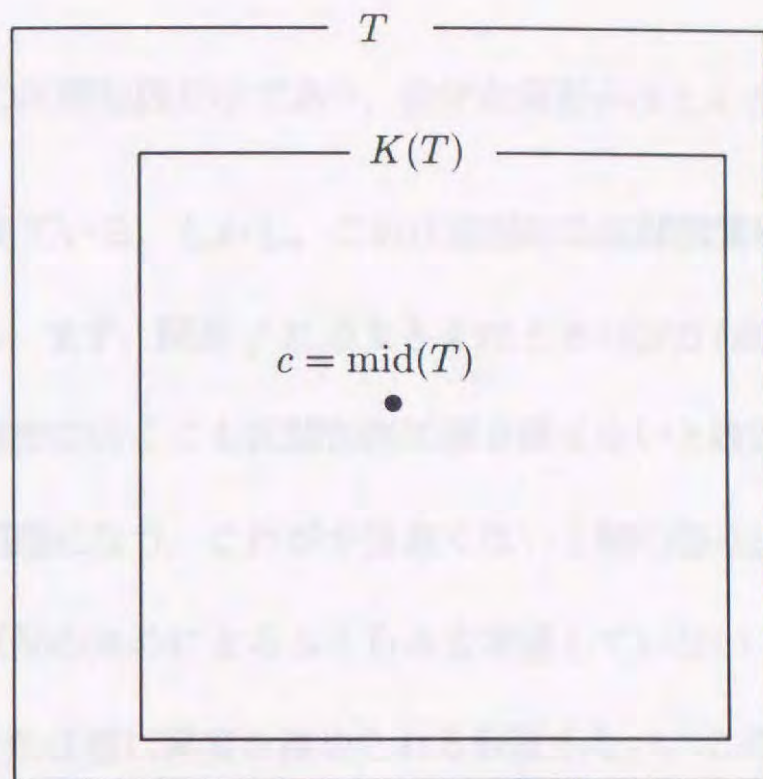


図 2.1: Krawczyk の方法

を満たすように定義されていることから，縮小写像原理により証明される。

K は， g の mean value form になっている．このことはこの方法にとって極めて重要である．
なぜなら， g の区間包囲を単純に

$$G(I) = I - L^{-1}F(I) \quad (2.36)$$

としてしまうと (F は f の区間包囲)，区間演算の減算の性質から $G(I) \subset I$ は決して成立しない。

2.4 むすび

本章では，本論文の議論の基礎となる区間解析について説明した．区間解析で基本となる区間演算をその用語とともに定義し，次に，区間演算を用いた非線形方程式の解の精度保証技法の基本的かつ代表的なものとして，Krawczyk の区間写像を用いた方法を説明した。

Krawczyk の方法は精度保証を行う上で，

- (1) 区間演算に基づいているため厳密な精度保証が行える。

(2) 自然でかつ効率の良い反復改良の方法を提供している.

(3) 必要なのは f と f' の区間包囲だけであり, 余分な仮定がほとんどないため自動化しやすい.

という理想的な性質を備えている. しかし, これは理想的な区間演算に基づいており, 現実には不可能な部分を含んでいる. まず, 関数 f に点を与えたときの出力 (点) を要求しており, これは実際には利用出来ない. 厳密にはここも区間包囲に置き換えないと厳密性が損なわれる. その時には, 区間包囲の精度が問題になり, これが十分良くないと解の包み込みが行えない. また, 反復改良の理論は途中での区間の丸めによるふくらみを考慮していない. 従って, 実際には理論通りに線形に収束はせず, また任意に精度を高められる保証もない. このような点の改善無くして, 本論文の目的とするような厳格な意味での精度保証は行えない.

3.1 はじめに

本章では，厳密な精度保証のために不可欠な，方程式の計算機上での表現方法について議論する．実数値連続関数を計算機上に厳密に情報落ちすることなく載せることを目的とする．そのために，従来の浮動小数点数に代わって，有理数演算を採用する．更に，計算誤差の影響を把握出来るように区間演算を採用する．また，こうして定義された区間写像から元の関数についての情報を完全に取り出せる (元の関数の定義となっている) ことを保証するための出来る限り緩くかつ確認しやすい条件として，連続な区間包囲の概念を提案する．

3.2 区間包囲による関数の表現

非線形方程式の近似解を精度保証する際，方程式自体をどう記述するかということは極めて重要な問題である．従来，精度保証の重要性が主張されるわりには，この方程式の記述の問題は余り真剣に議論されていないように思える．方程式を $f(x) = 0$ と書くとき，非線形写像 f は，一般的には計算機上で正確に表現できない．そこで，精度保証を行うために，

1. x に対して $f(x)$ を誤差評価付きで計算できること．
2. x に対して $f(x)$ を任意に高精度に計算できること．

が要請されると考えられる．特に，前者は必須で，誤差評価のない計算結果からは全く情報を得ることは出来ない．

ここでは， f の計算機上の表現 F として， f の区間包囲を用いる．この場合出力の区間幅がそのまま誤差評価を表す．非線形写像全体を考えると，区間包囲の構成は極めて難しい．しかし，通常我々が解析の対象とする非線形写像は，性質の良く分かっている基本関数の組み合わせであるものがほとんどなので，基本関数の区間包囲を作っておけばほとんどの非線形写像に対して自然に区間包囲が実現できることになる．

ところが、区間包囲が余りに大きな区間を出力するようでは、精度保証には役立たない。一定の条件の下で真解の包み込みに成功すること、また解の精度を任意に改良出来ることを保証するためには区間包囲がある程度以上の精度を持っていることが当然要求される。

本論文では、区間包囲の計算精度に関して、以下のように区間包囲の連続性を導入し、それを満たすものとして議論を進めることにする。

定義 3.1 (区間包囲の連続性) $f: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ を連続写像, $F: I(D) \rightarrow I(\mathbb{R}^m)$ を f の区間包囲とする。このとき、点 $c \in D$ において、

$$I_k \in I(D), \text{rad}(I_k) \neq 0 \text{ なる区間列 } \{I_k\} \text{ について } \|I_k - c\| \rightarrow 0 \text{ ならば } \|F(I_k) - f(c)\| \rightarrow 0,$$

すなわち、

全ての $\varepsilon > 0$ に対してある $\delta > 0$ が存在して

$$\text{rad}(I) \neq 0, \|I - c\| < \delta \implies \|F(I) - f(c)\| < \varepsilon \quad (3.1)$$

とできる

とき、 f の区間包囲 F は c で連続であるという。また、全ての $c \in D$ で連続であるとき、 D で連続であるという。□

これは、普通の連続性の定義との類似性からこう呼ぶことにした。要するに、“幅の狭い区間を入力するほど幅の狭い区間が出力される” という性質である。概念図を図 3.1 に示す。このとき、表現する関数 f は必然的に連続となるが、一様連続性は要求されない。通常の浮動小数点数による区間演算では、ある幅 (最良で浮動小数点数の隣同士の幅) 以下には狭くならず、この条件を満たすことはできない。

この仮定を満たすような区間包囲 F を計算機上に構成するためには、入力された区間ベクトルの幅を見て、それに応じて区間演算の精度 (外側への丸め幅) を変化させればよい。入力された区

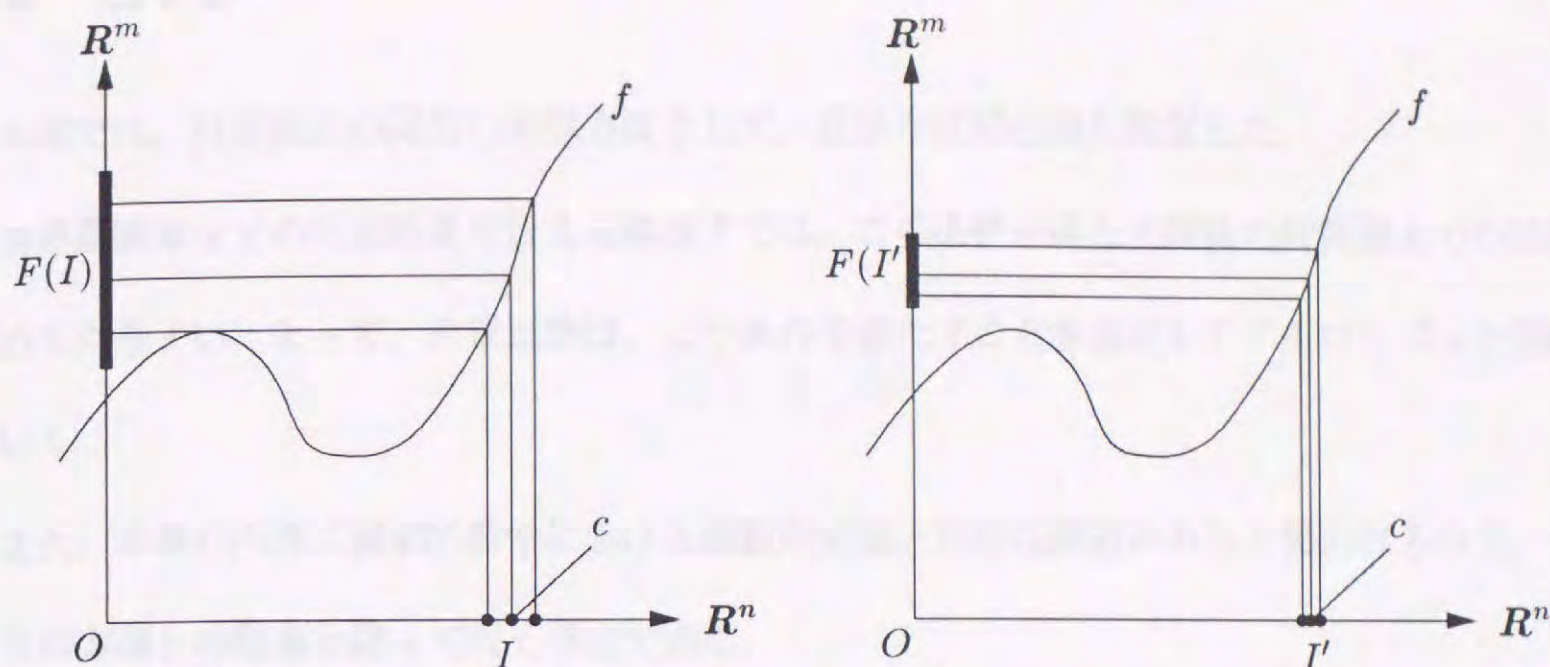


図 3.1: 連続な区間包囲

間の幅が狭いほど、高精度な演算が要求されていると解釈することになる。そのためには、有理数演算の導入などにより計算機内で任意精度実数の扱いを実現する必要がある。

ただし、 $\text{rad}(I_k) \neq 0$ が仮定されていることから分かるように、 F は幅 0 の区間の入力も出来なくてもよいことにする。この条件を入れないと、幅 0 の点を入力したとき幅 0 の区間を出力しなければならず、それは任意精度計算が可能だとしても一般には実現出来ない。これは結果が無理数となるような場合を考えればすぐに分かる。

基本的には、加減乗除や超越関数などの基本演算は、その演算に与えられた区間の幅を見て、それに応じた精度の区間演算を行えばよい。しかし、入力区間幅が 0 の場合 (関数全体に対する入力 I が $\text{rad}(I) \neq 0$ であったとしても、計算途中で幅 0 の区間が現れることもある) や、 π などの無理数が関数中に現れた場合を考慮して、次のようにする。最初に関数に対して与えられた入力 I に対して、 $\varepsilon = \|\text{rad}(I)\|$ を計算しておき、先のような場合には区間幅の代わりにこの ε を精度基準とみなして計算する。あるいは、この ε を全ての演算の精度基準とみなし、計算を行うという方法も考えられる。要するに、通常の区間演算と同じだが、入力区間幅が小さい場合は数値表現を 3, 4, ... 倍精度というように切り替えて行くと考えれば良い。

3.3 むすび

本章では、計算機上の関数の表現方法として、連続な区間包囲を提案した。

有理数演算などの任意精度を扱える環境下では、この条件を満たす関数の計算機上での実現は極めてたやすい。よって、次章以降は、この条件を満たすことを仮定してアルゴリズムを構成していく。

また、本章の内容は構成的数学における関数の定義と密接な関連があると思われるので、今後はその方面との関連を探っていく予定である。

区間演算による近似解の包み込み

4.1 はじめに

本章では、数値計算でよく用いられる区間演算について、数値演算を行う方法を解説する。区間演算は、数値計算の誤差を厳密に制御するために用いられる。数値計算を行う際に、数値の誤差を厳密に制御して数値計算を行う方法として区間演算がある。本章では、区間演算の基本的な概念について説明する。

第 4 章

区間演算による近似解の包み込み

本章では、数値計算でよく用いられる区間演算について、数値演算を行う方法を解説する。区間演算は、数値計算の誤差を厳密に制御するために用いられる。数値計算を行う際に、数値の誤差を厳密に制御して数値計算を行う方法として区間演算がある。本章では、区間演算の基本的な概念について説明する。

4.2 区間演算による近似解の包み込み

本章では、数値計算でよく用いられる区間演算について、数値演算を行う方法を解説する。区間演算は、数値計算の誤差を厳密に制御するために用いられる。数値計算を行う際に、数値の誤差を厳密に制御して数値計算を行う方法として区間演算がある。本章では、区間演算の基本的な概念について説明する。

4.1 はじめに

本章では，従来の数値計算で求めた近似解を元にして，精度保証を行う方法を提案する．全ての計算を精度保証付き数値計算に置き換えるのではなく，従来法を補完して精度保証を行う方がより実用的であるという考えに基づいている．

この方法は，区間演算と平均値の定理，不動点定理を巧みに組み合わせた Krawczyk の方法 [38, 39, 14] を基礎にして構成する．ところが Krawczyk の方法は，近似解が与えられたとき，その点を元にして真解の包み込みを行うには適していない．なぜなら，近似解に対して，真解を含むような適当な大きさの区間 T を決定する必要があり，その方法が不明であるためである．区間演算を近似解の精度保証に利用するための方法としては，Rump の提案する手法 [45] がある．近似解を初期点とするある種の区間反復を行うものであるが，十分良い近似解に対して必ず停止するという保証はない．

ここでは，第3章で提案したような関数表現を仮定して，Krawczyk の方法に対して，近似解が与えられたときそれに対する具体的な初期区間 T を与える方法を提案する．また，その方法で十分良い近似解に対して真解の包み込みが行えることを示す．

4.2 区間演算による近似解の包み込み

初期区間 T は次のように与える．まず，近似解 c で $\|f'(c)^{-1}f(c)\|$ (Newton 法の一回目の反復における修正量) を計算する．次に， c を中心とし，計算した量の $\rho > 1$ 倍を半径とする球 (区間ベクトル) を作り，この領域に対して Krawczyk の方法を適用する．図 4.1 にこの様子を示す．本論文では関数 f 自身を計算機で利用することはできず，また区間包囲に一点を入力することは出来ないという立場をとっているため，その代わりに近似解を含む微小区間を用いる．まとめると次のようになる．

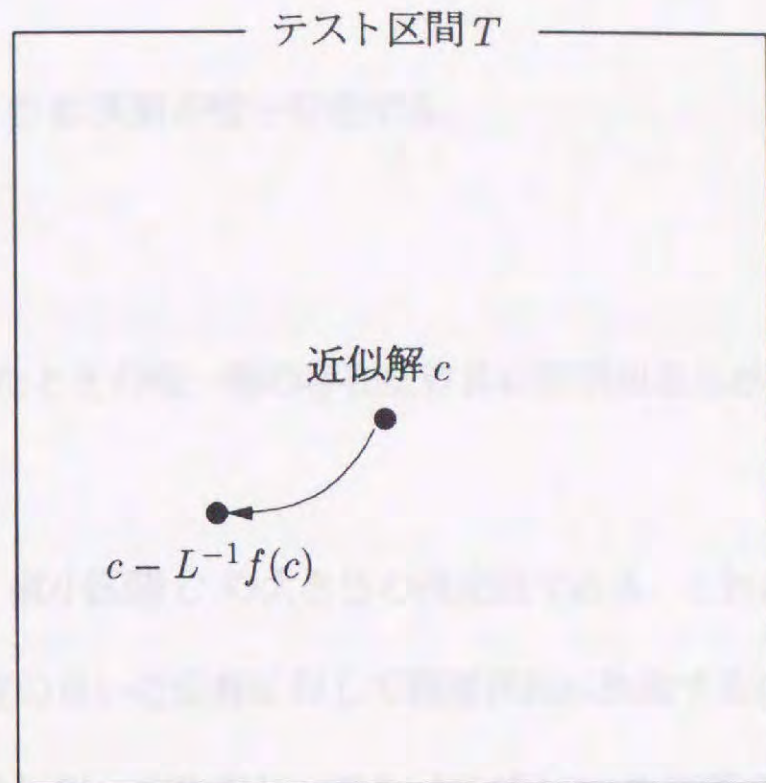


図 4.1: テスト区間 T の生成

アルゴリズム 4.1 (近似解を元にした真解の包み込み) $U \subset \mathbb{R}^n$, $f: U \rightarrow \mathbb{R}^n$ を C^1 級とし, \mathbb{R}^n のノルムは $u > 0$ を scaling vector とする scaled maximum norm とする. F, F' はそれぞれ f, f' の (定義 3.1 の意味で) 連続な区間包囲とする. $c \in U$ を近似解, $C \in I(U)$ を $\text{mid}(C) = c$ であるような微小区間とし, $\rho > 1$ (定数) とする.

- (1) $F(C), F'(C)$ を計算する.
- (2) $L \in F'(C)$ を任意に決定.
- (3) L^{-1} が存在しなければ失敗. 存在すれば, $L^{-1}F(C)$ を計算し,

$$\delta = \rho \|L^{-1}F(C)\|_u, \quad (4.1)$$

$$T = [-\delta u, \delta u] + c \quad (4.2)$$

とする.

- (4) $M = E - L^{-1}F'(T)$ とし, 条件

$$c - L^{-1}F(C) + M(T - c) \subset T \quad (4.3)$$

$$\|M\|_u < 1 \quad (4.4)$$

が成立すれば, 区間 T に真解が唯一存在する.

□

式 (4.3), (4.4) が成立したときの唯一解の存在は容易に証明出来るが, これは後の定理 5.1 で行うことにする.

ここで問題となるのは, 微小区間 C の大きさの決定法である. これが大きすぎると, 本来精度保証に成功する程度に精度の良い近似解に対して精度保証に失敗する恐れがある. ここでは, 微小区間の大きさを次の方法に従って決定し, アルゴリズム 4.1 を適用することにする.

アルゴリズム 4.2 (微小区間の大きさを考慮した方法) $U \subset \mathbb{R}^n$, $f: U \rightarrow \mathbb{R}^n$ を C^1 級とし, \mathbb{R}^n のノルムは $u > 0$ を scaling vector とする scaled maximum norm とする. F, F' はそれぞれ f, f' の (定義 3.1 の意味で) 連続な区間包囲とする. $c \in U$ を近似解とし, $0 < r < 1$ を微小区間の縮小率とする.

- (1) $k = 1$ とし, 微小区間の半径 d_1 を適当に定める.
- (2) $C_k = [-d_k u, d_k u] + c$ とする.
- (3) 微小区間 C_k にアルゴリズム 4.1 を適用する. 失敗したら次へ.
- (4) $F(C_k) \ni 0$ ならば, $0 < d_{k+1} \leq r d_k$ のように微小区間の半径を小さくし, $k = k + 1$ として (2) へ. そうでなければ近似解 c の精度が十分でなかったとする.

□

ここでは, 万一与えられた近似解が真解であった場合を考慮し, やや複雑になっている.

4.3 アルゴリズムの妥当性

前節で、与えられた近似解を元にした真解の包み込みアルゴリズムが与えられたが、本節では、そのアルゴリズムの有効性を示す。

アルゴリズム 4.2によって十分良い近似解に対して真解の包み込みが行えることを、次の定理 4.1に示す。

定理 4.1 (アルゴリズム 4.2による真解の包み込み) アルゴリズム 4.2の仮定の下で、 $x^* \in U$ が、 $f(x^*) = 0$ 、 $\exists f'(x^*)^{-1}$ を満たすとする。このとき、 $\|c - x^*\|_u$ が十分小さい近似解 c に対して、アルゴリズム 4.2は解 x^* を包みこむ。 \square

(証明) 以下、 $\|\cdot\| = \|\cdot\|_u$ とする。

$c = x^*$ の場合と $c \neq x^*$ の場合に分けて考える。

$c = x^*$ の場合 この場合、必ず $F(C_k) \ni 0$ となるので、 C_k に対するアルゴリズム 4.1の適用が永久に行われる。このとき、 $\|C_k - x^*\| \rightarrow 0$ が成立する事に注意する。

$c \neq x^*$ の場合 $c_j \rightarrow x^*$ かつ $c_j \neq x^*$ なる近似解の列 $\{c_j\}$ を考え、十分大きい j に対してアルゴリズム 4.2が成功する事を示せばよい。このとき、各 c_j について、必ず $F(C) \ni 0$ なる微小区間 C に対するアルゴリズム 4.1の適用が行われる。これらの微小区間を C_j と書く事にする。このとき、 $C_j \ni 0$ に注意すると、 $\|C_j - x^*\| \rightarrow 0$ が成立する事が分かる。

よって、どちらの場合を考えても、 $\|C_j - x^*\| \rightarrow 0$ なる微小区間の列 $\{C_j\}$ について、十分大きい j に対してアルゴリズム 4.1が解の包み込みに成功する事を示せばよい。

$C = C_j$ としてアルゴリズム 4.1を実行したときの c, L, δ, T, M をそれぞれ $c_j, L_j, \delta_j, T_j, M_j$ と書く。

仮定より x^* は孤立解であるから、十分大きい j に対して式 (4.3), (4.4) が成立した場合、 T_j が包みこむ解は x^* である。よって十分大きい j に対して式 (4.3), (4.4) が成立することを言えばよい。

式(4.3)について,

$$1 + \|M_j\|\rho \leq \rho \quad (4.5)$$

↓

$$\|L_j^{-1}F(C_j)\| + \|M_j\|\delta_j \leq \delta_j \quad (4.6)$$

↓

$$|L_j^{-1}F(C_j)| + |M_j|\delta_j u \leq \delta_j u \quad (4.7)$$

↓

式(4.3)

となるので, 十分大きい j に対して式(4.5)をいえばよい. ここで, 式(4.4)と $\rho > 1$ を考えると,

$$\|M_j\| \rightarrow 0 \quad (j \rightarrow \infty) \quad (4.8)$$

を示せば題意が示されることが分かる.

まず, F' の定義 3.1 の意味での連続性と $\|C_j - x^*\| \rightarrow 0$, $\text{rad}(C_j) \neq 0$ により,

$$\|L_j - f'(x^*)\| \leq \|F'(C_j) - f'(x^*)\| \rightarrow 0 \quad (4.9)$$

となる. よって, $\beta \stackrel{\text{def}}{=} \|f'(x^*)^{-1}\|$ とすると, $\beta\|L_j - f'(x^*)\| < 1$ となるような十分大きい j に対して, ある定数 $\gamma > \beta$ が存在し,

$$\|L_j^{-1}\| \leq \frac{\beta}{1 - \|L_j - f'(x^*)\|\beta} \leq \gamma \quad (4.10)$$

が成立する.

また, 同様に F の連続性を用いて,

$$\begin{aligned} \|T_j - x^*\| &\leq \|T_j - c_j\| + \|c_j - x^*\| \\ &\leq \delta_j + \|C_j - x^*\| \end{aligned}$$

$$\begin{aligned}
&= \rho \|L_j^{-1} F(C_j)\| + \|C_j - x^*\| \\
&\leq \rho \gamma \|F(C_j)\| + \|C_j - x^*\| \\
&= \rho \gamma \|F(C_j) - f(x^*)\| + \|C_j - x^*\| \\
&\rightarrow 0
\end{aligned} \tag{4.11}$$

が成立する.

更に, 仮に $\text{rad}(T_j) = 0$ であるとする,

$$\begin{aligned}
\text{rad}(T_j) &= |T_j - c_j| \\
&= \delta_j u
\end{aligned} \tag{4.12}$$

$$= \rho \|L_j^{-1} F(C_j)\| u \tag{4.13}$$

より $F(C_j) = 0$ となるが, これは L_j^{-1} が存在する事と矛盾するので, $\text{rad}(T_j) \neq 0$ が言える事になる.

よって, F' の連続性により,

$$\|F'(T_j) - f'(x^*)\| \rightarrow 0 \tag{4.14}$$

が言える.

以上により, 式 (4.8) を示すことが出来る. 式 (4.9), (4.10), (4.14) を用いて

$$\begin{aligned}
\|M_j\| &= \|E - L_j^{-1} F'(T_j)\| \\
&\leq \gamma \|L_j - F'(T_j)\| \\
&\leq \gamma (\|L_j - f'(x^*)\| + \|f'(x^*) - F'(T_j)\|) \\
&\rightarrow 0
\end{aligned} \tag{4.15}$$

となり, 題意は示された. □

ρ が 1 よりも大きい実数であればこの定理は成立するが、幾つかの理由により、 $\rho = 2$ が最適であると考えている。

4.4 むすび

本節では、第 3 章で提案したような関数表現を仮定して、Krawczyk の方法に対して、近似解が与えられたときそれに対する具体的な初期区間 T を与える方法を提案した。また、その方法で十分良い近似解に対して真解の包み込みが行えることを示した。

第 3 章で提案した関数表現 (区間包囲の連続性) を利用するためには、区間包囲を実現するために、関数の演算構造が必要になる。本章の方法では $f(x) = 0$ の解の包み込みを行うのに f のみならず f' の区間包囲も必要となるので、 f' の演算構造も必要となる。すなわち、数値微分等による微分では役に立たず、数式处理的な微分が要求されてしまう。これは、自動微分の技法を用いることにより回避できるが、これについては第 6 章で詳しく述べることにする。

5.1 はじめに

本書により、有理数演算による反復改良の重要性が示される。本書では、有理数演算による反復改良の重要性を示す。

第 5 章

有理数演算による反復改良

有理数演算による反復改良の重要性を示す。有理数演算による反復改良の重要性を示す。有理数演算による反復改良の重要性を示す。

本書では、以上のように入力値を修正した反復改良の重要性を示す。この場合、反復改良の重要性を示す。反復改良の重要性を示す。

本書の目的は、文法規則の理解と実行である。

5.2 有理数演算による反復改良

本書では、有理数演算による反復改良の重要性を示す。有理数演算による反復改良の重要性を示す。有理数演算による反復改良の重要性を示す。

$$x + |x| < 1$$

5.1 はじめに

前章により，近似解を用いた真解の包み込みが可能になった．本章では任意精度までの反復改良の方法を提案する．

定理 2.1 の (2) によれば，区間反復によって解を含む任意に小さい区間を得ることが出来る．しかし，本章では方程式を記述する写像 f は正確に表現できないという立場を取っており，区間包囲 F の返す区間は幅を持つので，当然定理どおりの反復改良は行えない．また，仮に区間包囲 F が幅 0 の区間 (つまり点) を返したとしても，理論通り全く丸めのない区間反復を有理数を用いて行えば，分母分子が“桁の爆発”を起こし，ほとんど反復を進めることはできない．

区間反復を効率良く行う方法として良く知られているのは，機械区間演算 (計算機が通常持つ浮動小数点数を両端に持つ区間演算) を用い，内積演算に限って十分長いアキュムレータを用いた完全精度演算を導入するもので，多くの場合最大精度の (浮動小数点の最小単位の誤差の) 区間解が得られることが知られている．しかし，この方法によって必ず最大精度が達成されるという保証はなく，方程式を記述する写像の誤差に対する考慮もなされていない．

本章では，以上のような問題点を解決した区間反復法を提案する．これは，有理数演算を用いつつ，反復を行う毎に丸めを行い桁の爆発を防ぐというものである．また，その際に同時に F の返す区間の幅も考慮に入れ，毎回の丸めの幅を適切に制御することによって定理 2.1 の (2) のような区間幅の線形収束性を損なわないようにすることが出来る．

本章の内容は，文献 [46] の発展と言える．

5.2 有理数演算による反復改良

アルゴリズム 4.1 で区間 T に真解 x^* が包み込まれたとする．以下， $\|\cdot\| = \|\cdot\|_u$ とする．

まず最初に，縮小定数の余裕分 $\kappa \in Q^{n \times n} > 0$ (行列) を決定する．これは，

$$\|\kappa + |M|\| < 1 \quad (5.1)$$

を満たす行列とする。これは、丸めによって縮小率を $\|M\|$ から $\|\kappa + \|M\|\|$ に悪化させることを意味する。この κ を用いて丸めを制御する。

反復改良は、 $I_0 = T$ を初期区間とする。以下に、区間 I_k から区間 I_{k+1} を生成するアルゴリズムをまとめる。

- (1) 現在の区間幅 $\text{rad}(I_k)$ (ベクトル) と κ から、丸めの許容誤差 (関数値の計算誤差, 区間の丸め) を決定する。具体的には,

$$p_k + r_k \leq \kappa \text{rad}(I_k) \quad (5.2)$$

を満たすベクトル $p_k > 0, r_k > 0$ を決定する。

- (2) $c_k = \text{mid}(I_k)$ とし, $\text{rad}(L^{-1}F(C_k)) \leq p_k$ となるような $C_k \ni c_k$ を決定する。

- (3)

$$I_{k+1}^* = c_k - L^{-1}F(C_k) + M(I_k - c_k) \quad (5.3)$$

によって I_{k+1}^* を誤差無しで計算する。

- (4) I_{k+1}^* を誤差 r_k 以内で外側に丸め, I_{k+1}^{**} とする。

- (5)

$$I_{k+1} = I_k \cap I_{k+1}^{**} \quad (5.4)$$

によって I_{k+1} を決定する。

この様子を図 5.1 に示す。斜線部が I_{k+1} である。

$\text{rad}(I_k)$ は次第に小さくなっていくので、丸めの許容誤差も次第に小さくなっていく。すなわち、反復の初期段階は粗い精度で計算し、反復が進行するにつれ高精度になっていくようなアルゴリズムになっている。

このアルゴリズムについて、次の定理 5.1 が成立する。

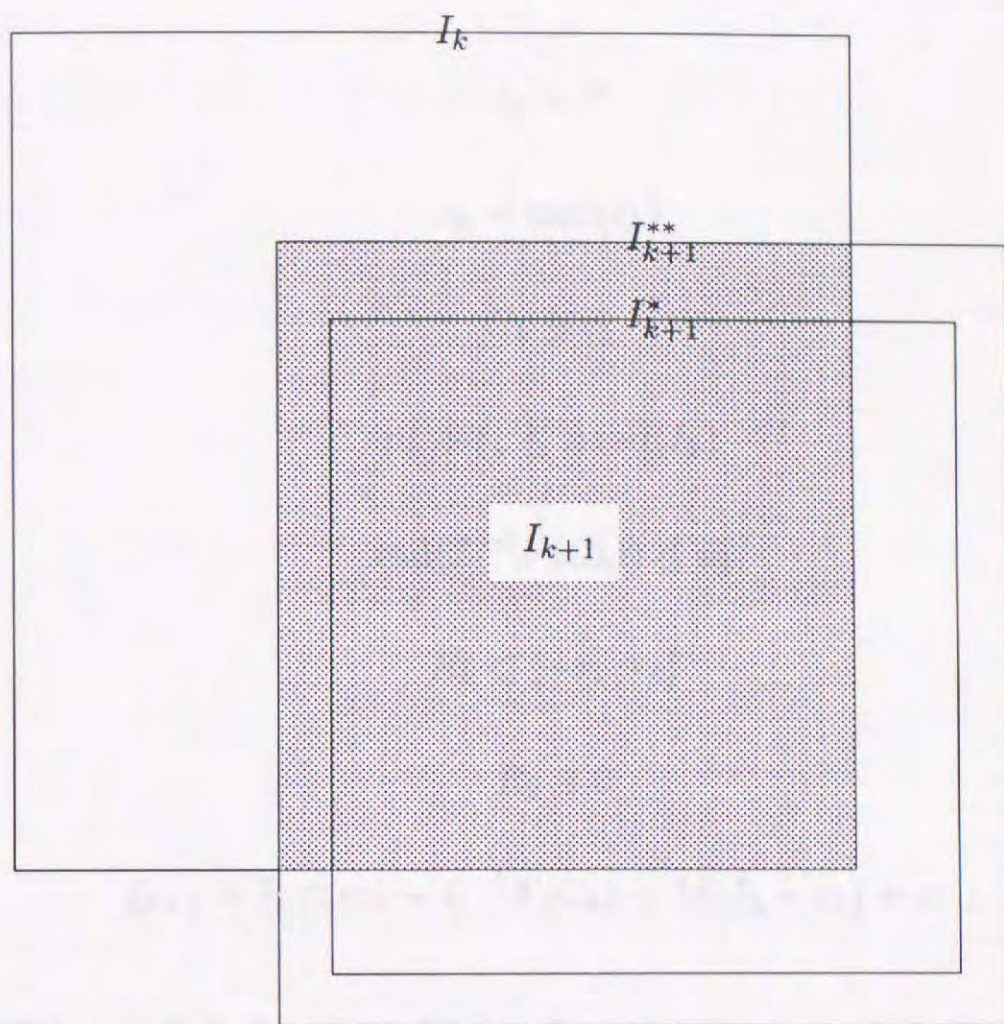


図 5.1: 区間反復 (I_k から I_{k+1} を生成)

定理 5.1 (有理数演算を用いた区間反復法) $U \subset \mathbb{R}^n$, $f: U \rightarrow \mathbb{R}^n$ を C^1 級とし, F は f の区間包囲であるとする. ある区間 $T \in I(U \cap \mathbb{Q}^n)$, 非区間行列 $L \in \mathbb{Q}^{n \times n}$, 区間行列 $M \supset \{E - L^{-1}f'(x) \mid x \in T\}$, 微小区間 $C \ni c = \text{mid}(T)$ について,

$$T \supset c - L^{-1}F(C) + M(T - c) \quad (5.5)$$

$$\|M\| < 1 \quad (5.6)$$

が成立するとする. このとき, 以下が成立する:

- (1) T に方程式 $f(x) = 0$ の解 x^* が唯一存在する.
- (2) 行列 $\kappa \in \mathbb{Q}^{n \times n} > 0$, 点列 $\{p_k\}$ ($p_k \in \mathbb{Q}^n > 0$), $\{r_k\}$ ($r_k \in \mathbb{Q}^n > 0$), 微小区間列 $\{C_k\}$ ($C_k \in I(\mathbb{Q}^n) > 0$), 区間列 $\{I_k\}$ ($I_k \in I(U \cap \mathbb{Q}^n)$), $\{R_k\}$ ($R_k \in I(\mathbb{Q}^n)$) につ

いて,

$$I_0 = T \quad (5.7)$$

$$c_k = \text{mid}(I_k) \quad (5.8)$$

$$C_k \ni c_k \quad (5.9)$$

$$p_k + r_k \leq \kappa \text{rad}(I_k) \quad (5.10)$$

$$\text{rad}(L^{-1}F(C_k)) \leq p_k \quad (5.11)$$

$$R_k \subset [-r_k, r_k] \quad (5.12)$$

$$R_k \ni 0 \quad (5.13)$$

$$I_{k+1} = I_k \cap (c_k - L^{-1}F(C_k) + M(I_k - c_k) + R_k) \quad (5.14)$$

が成立するとする. このとき, $\{I_k\}$ について,

$$x^* \in I_k \quad \text{for all } k, \quad (5.15)$$

$$\|\text{rad}(I_{k+1})\| \leq \|\kappa + |M|\| \cdot \|\text{rad}(I_k)\| \quad (5.16)$$

が成立する.

□

式 (5.16) が, 反復の収束速度を表している. これを見ると, この反復が収束するために式 (5.1) ($\|\kappa + |M|\| < 1$) が必要であることが分かる.

証明に必要な補題を文献 [17] から挙げておく.

補題 5.1 A を $m \times n$ 区間行列, B を $n \times p$ 区間行列とすると, $\text{mid}(B) = 0$ ならば,

$$AB = |A|B = [-|A||B|, |A||B|] = [-1, 1]|A||B| \quad (5.17)$$

が成立する.

□

補題 5.2 A, B を区間ベクトルとするとき,

$$\text{rad}(A + B) = \text{rad}(A) + \text{rad}(B) \quad (5.18)$$

が成立する. □

定理 5.1 の証明を示す.

(証明)

作用素 $g: U \rightarrow R^n$ を

$$g(x) = x - L^{-1}f(x) \quad (5.19)$$

で定義する.

ここで, $x \in T$ に対して, $g'(x) = E - L^{-1}f'(x) \subset M$ なので, 式 (5.6) より

$$\|g'(x)\| < 1 \quad (5.20)$$

が成立する. また, 同じく $x \in T$ に対して,

$$\begin{aligned} g(x) &= g(c) + \int_0^1 g'(tx + (1-t)c)(x-c)dt \\ &= c - L^{-1}f(c) + \int_0^1 (E - L^{-1}f'(tx + (1-t)c))(x-c)dt \\ &\subset c - L^{-1}F(C) + \int_0^1 M(x-c)dt \\ &= c - L^{-1}F(C) + M(x-c) \\ &\subset c - L^{-1}F(C) + M(T-c) \\ &\subset T \quad (\text{for 式 (5.5)}) \end{aligned} \quad (5.21)$$

が成立する. 式 (5.20) を合わせると, g は T から T への縮小写像であることが分かる.

$$g(x) = x \iff f(x) = 0 \quad (5.22)$$

であるから, 縮小写像原理により (1) の成立が従う.

次に, (5.15) の成立を示す. (5.7), (5.14) により, $I_k \subset T$ である. また, $x \in I_k$ に対して,

$$\begin{aligned}
 g(x) &= g(c_k) + \int_0^1 g'(tx + (1-t)c_k)(x - c_k) dt \\
 &= c_k - L^{-1}f(c_k) + \int_0^1 \left(E - L^{-1}f'(tx + (1-t)c_k) \right) (x - c_k) dt \\
 &\subset c_k - L^{-1}F(C_k) + \int_0^1 M(x - c_k) dt \quad (\text{for } I_k \subset T) \\
 &= c_k - L^{-1}F(C_k) + M(x - c_k) \\
 &\subset c_k - L^{-1}F(C_k) + M(I_k - c_k) + R_k \quad (\text{for 式 (5.13)})
 \end{aligned} \tag{5.23}$$

となるので, 式 (5.14) より

$$x^* \in I_k \implies x^* \in I_{k+1} \tag{5.24}$$

が分かる. $x^* \in I_0 = T$ であるから, 帰納法により (5.15) の成立が分かる.

最後に, (5.16) の成立を示す. 不等式

$$\begin{aligned}
 \text{rad}(I_{k+1}) &= \text{rad}(I_k \cap (c_k - L^{-1}F(C_k) + M(I_k - c_k) + R_k)) \quad (\text{for 式 (5.14)}) \\
 &\leq \text{rad}(L^{-1}F(C_k)) + \text{rad}(M(I_k - c_k)) + \text{rad}(R_k) \quad (\text{for 式 (5.14), 補題 5.2}) \\
 &\leq p_k + r_k + \text{rad}(M(I_k - c_k)) \quad (\text{for 式 (5.11), (5.12)}) \\
 &= p_k + r_k + \text{rad}([-1, 1] |M| |I_k - c_k|) \quad (\text{for 補題 5.1}) \\
 &= p_k + r_k + |M| |I_k - c_k| \\
 &= p_k + r_k + |M| \text{rad}(I_k)
 \end{aligned} \tag{5.25}$$

が成立し, 条件 (5.10) を考えると,

$$\begin{aligned}
 \text{rad}(I_{k+1}) &\leq \kappa \text{rad}(I_k) + |M| \text{rad}(I_k) \\
 &= (\kappa + |M|) \text{rad}(I_k)
 \end{aligned} \tag{5.26}$$

が成立する. 両辺のノルムを考えると, (5.16) の成立が分かる. \square

5.3 むすび

本章では、第3章で提案した関数表現を仮定し、また有理数演算を用いた区間反復法を提案した。この方法では、適切に制御された有理数の丸めによって、有理数の桁数の爆発の防止と任意精度までの反復改良を同時に達成している。また、それが正しく動く事を証明した。

精度保証システムの実現

6.1 はじめに

本章では、精度保証システムの構築の背景について紹介する。まず、本論文で扱っている精度保証システムの構築の背景について紹介する。次に、本論文で扱ったアルゴリズムの構築の背景について紹介する。最後に、本論文で扱ったアルゴリズムの構築の背景について紹介する。

第 6 章

6.2 精度保証

精度保証システムの実現

本章では、本章で扱ったような精度保証システムを実現するためのソフトウェアの構築の背景について紹介する。まず、精度保証システムの構築の背景について紹介する。次に、本章で扱ったアルゴリズムの構築の背景について紹介する。最後に、本章で扱ったアルゴリズムの構築の背景について紹介する。

このような構築の背景の下では、精度保証システムの構築の背景について紹介する。まず、精度保証システムの構築の背景について紹介する。次に、本章で扱ったアルゴリズムの構築の背景について紹介する。最後に、本章で扱ったアルゴリズムの構築の背景について紹介する。

本章で扱ったアルゴリズムを実現するために、本章で扱ったアルゴリズムの構築の背景について紹介する。まず、精度保証システムの構築の背景について紹介する。次に、本章で扱ったアルゴリズムの構築の背景について紹介する。最後に、本章で扱ったアルゴリズムの構築の背景について紹介する。

本章で扱ったアルゴリズムの構築の背景について紹介する。まず、精度保証システムの構築の背景について紹介する。次に、本章で扱ったアルゴリズムの構築の背景について紹介する。最後に、本章で扱ったアルゴリズムの構築の背景について紹介する。

6.1 はじめに

本章では、精度保証システムの実際の実現について議論する。まず、本論文(第3-5章)で提案した方法を実現するために必要なソフトウェア、プログラミング技法について議論する。次に、本論文で提案したアルゴリズムの実際の実作について説明する。自動微分の技法を用い、方程式と近似解の2つの入力だけで自動的に精度保証を行うシステムの構築を行った。最後に、幾つかの例題に対する精度保証を行いその有用性を示す。

6.2 精度保証システム

本節では、本論文で提案したようなアルゴリズムを実現するためのソフトウェアについて考える。現在の電子計算機では、数値計算のための機構として浮動小数点演算を採用している。単精度(32bit)及び倍精度(64bit)による演算機構を備え、有効桁数は10進数でそれぞれ6,15桁程度である場合が多い。また、浮動小数点演算はハードウェアによる専用の演算機構によって極めて高速化されている。

このような演算方式とその高速化は、数値計算の実用化に大きな役割を果たしてきた。しかし、精度保証付き数値計算に使うには、誤差の発生の予測が難しく、精度も有限でしかない現状の演算機構では役に立たず、新しい演算方式が必要となる。

本論文のアルゴリズムを実現するためには、任意に高精度な数値が扱える演算体系が必要である。例えば分母分子を長整数として持つ有理数演算や、任意に仮数部、指数部を増大できる浮動小数点演算などである。既存のプログラミング言語では、LISP系の言語や数式処理システムの一部にこれが組み込まれているが、一般的ではない。また、これらはハードウェアの支援がないため全てソフトウェアで実装しており、演算速度は概して遅い。

任意高精度な演算体系が利用できたとしても、実数そのものを表現することはできないので、結局誤差評価は必要である。そのため、区間演算を利用する必要がある。また、精度保証アルゴリ

ズムでは、不動点定理の利用や区間評価の高精度化などに関して、プログラムによって定義された関数の入力変数に関する微分が必要になる場合が多い。これらは必然的に大規模になる場合が多く、自動微分 [30, 31, 8] を利用する必要がある。

区間演算や自動微分の実装は、原理的には大抵の言語で可能であるが、精度保証付き数値計算が実用になるためには、それが比較的容易である必要がある。その手法を利用するために事前に多くの検討を行う必要があれば、その段階での間違いの混入が防げなくなり、計算結果の信頼性の低下を招くことになる。更に重要なこととして、これらの技法は混在して用いられることに注意しなくてはならない。すなわち、自動微分を行うときに用いられる数値は区間であり、その区間の両端の数は有理数で表現される。このようなプログラミングは既存のCなどの言語ではかなり困難である。すなわち、自動微分や区間演算は universal な数値に対して行うようにプログラムし、数値同士の演算は runtime にその時に与えられた数値の型によって決定されるようなプログラミングが要求される。

このためには、演算子や関数の多重定義 (polymorphism) が可能で、更に動的結合 (dynamic binding) の可能な object 指向的な言語が望ましい。

6.3 精度保証システムの試作

前節で述べたように、精度保証付き数値計算の実現には object 指向的な言語がほとんど必然的に要求される。C++言語は現状でこのような目的に適した言語の一つと考えられる。しかし、有理数演算を行うための機構がなく、これを実現する実用的なクラスライブラリも存在しなかった。自力で実用的な有理数演算を実装するのはかなりの技術と手間を要するので、C++言語は採用しなかった。

現在試作中のシステムでは、David I. Bell 氏による

CALC - C-style arbitrary precision calculator : version 1.26.4

を改良したものを採用している。このフリーソフトウェアは全てC言語で記述されており、任意精度計算の可能な対話的な電卓プログラムである。数値演算は全て有理数で行われ、任意精度計算が可能な様々な数学関数を備えている。文法はC言語に極めて類似しており、新しい関数を定義することによってプログラミングが行える。インタプリタであるため変数の型に関して柔軟であり、ベクトルや行列は要素として任意の型を持つことが出来る。更に、ユーザーが新しい型を定義することも出来、それに対する演算子の働きを多重定義出来るため、極めて容易に言語の拡張が行える。この言語で精度保証アルゴリズムを実現するには、特に object を扱う機構に不十分な点があったので、

- 複数 object 間の優先順位の設定機構の実現
- 関数の overload 機構の実現
- object と matrix の演算に関する優先順位の変更
- 幾つかの数学関数の精度を改善
- 演算子の優先順位の変更

などの独自の拡張を施している。また、多くの Bug を fix した。

この CALC に新しい型を追加し、また幾つかの関数を定義することによって、本章で述べたアルゴリズムをほぼ完全に実現することが出来た。以下、その概要を説明する。

区間演算の実現 区間演算を実現するために、上限と下限からなる区間型を定義し、演算子の多重定義によってその加減乗除を定義した。これは、非区間数との混在も可能とした。また、 mid , rad , $|\cdot|$, $\|\cdot\|_u$ などの関数を定義した。これらの関数は、区間だけでなくベクトルや行列に対しても適用可能とした。

自動微分の実現 自動微分を、Bottom Up 型のアルゴリズム [30] により実現した。関数値と入力変数に関する偏微分値を持つような自動微分型を定義し、その加減乗除を演算子多重定義に

よって定義した。これにより、関数への入力ベクトルをこの型で初期化しておけば、その後の演算で得られる全ての数値について自動的に入力変数に関する偏微分値が得られる。また、先に定義した区間型を同時に用いると、自然に関数のヤコビアンの区間包囲が実現できる。

数学関数の実現 上で定義した2つの型を考慮して、数学関数を定義した。すなわち、区間の入力に対して区間包囲として働くように、自動微分型の入力に対して自動微分を行うようにした。また、この場合の区間包囲の精度は、入力区間の幅またはある大域変数に従うようにした。これにより、連続な区間包囲が極めて容易に記述可能となった。

精度保証システムの実現 以上で定義した機能を用いて、与えられた近似解に対して精度保証を行うことが出来るようになった。具体的には、近似解、関数名、精度要求を与えて、それに対して精度保証を試みる関数を作成した。近似解が十分良い近似ならば、真の解を包含する区間を求め、それを精度要求まで反復改良して精度要求以下の幅を持つ区間を返す。近似解が悪ければその旨表示する。

自動微分により導関数の記述を必要としないので、本システムに対する入力は、方程式のプログラムによる記述と近似解だけである。プログラムに誤りが無いことを証明することは出来ないが、仮に誤りが無いとすると、このシステムによって精度保証された解は数学的に厳密なものである。

6.4 数値例

6.4.1 数値例 1

例題として、方程式

$$x_0^2 - x_1^2 - \sqrt{2} = 0 \quad (6.1)$$

$$\exp(x_0) - \frac{1}{x_1^3} = 0 \quad (6.2)$$

を考える. この方程式を記述する関数は, 例えば,

```
define func(x)
{
  local y;
  mat y[2];

  y[0] = x[0]^2 - x[1]^2 - Sqrt(2);
  y[1] = Exp(x[0]) - 1/x[1]^3;

  return y;
}
```

と書ける. これは, 2次元のベクトル型 x を受け取って, 2次元のベクトル型 y を返す 'func' という名前の関数の定義である. 但し, x 及び y の成分の型に関する定義はなく, この関数の動作は実行時に $x[0]$ 及び $x[1]$ に入っていた型によって決まる. 従って, 区間演算や自動微分を行う型の入力を与えてやるだけで, 区間演算や自動微分を行える.

近似解 (1.3, 0.6) を与えると, 精度保証に失敗した.

近似解 (1.35, 0.64) を与えると, 精度保証に成功した. このときのアルゴリズム中の諸量は, 次の通りとなった. ただし, 精度要求 (区間の半径がこれ以下になるまで反復) を 10^{-15} とした.

$$c = (1.35, 0.64)^t \quad (6.3)$$

$$F(c) = \begin{pmatrix} [-.001313562469695, -.001313562269695] \\ [.042728264972582, .042728265172582] \end{pmatrix} \quad (6.4)$$

$$F'(c) = \begin{pmatrix} 2.7 & -1.28 \\ [3.857425530597582, 3.857425530797582] & [17.881393432617187, 17.881393432617187] \end{pmatrix} \quad (6.5)$$

$$L = \begin{pmatrix} 2.7 & -1.28 \\ 3.857425530777046 & 17.881393432617187 \end{pmatrix} \quad (6.6)$$

$$T = \begin{pmatrix} [1.345454545454545, 1.354545454545454] \\ [.635454545454545, .644545454545454] \end{pmatrix} \quad (6.7)$$

$$M = \begin{pmatrix} [-.003477300141371, .003475383197188] & [-.028245443160889, .026769637247082] \\ [-.001550556761233, .001546513207098] & [-.053795858443029, .050682830343593] \end{pmatrix} \quad (6.8)$$

$$\kappa = \begin{pmatrix} .037037037037037 & .037037037037 \\ .025 & .025 \end{pmatrix} \quad (6.9)$$

これらの数値は、小数点以下 15 桁まで表示したが、内部では全て有理数で表現されている。反復改良の様子は、表 6.1, 6.2 に示す。分数の分母分子の長さを見ると、反復の初期段階は粗い精度で計算し、反復が進行するにつれ高精度になっていく様子が分かる。

κ, p_k, r_k の決め方や丸めの方法など、本章の方法にはまだ自由度が残されているが、実際のシステムではこれらも固定されている。ここで採用した方法については、別の機会に改めて述べることにする。

6.4.2 数値例 2—区分解形系における“Lost Solution”の厳密解析—

文献 [47] で、ある区分解形 2 次元自立系において Lost Solution と呼ばれる、パラメータの微小変化が発振振幅の大きな変化を引き起こす現象が報告されている。パラメータを微小変化させたときの軌道を図 6.1 に示す。また、同じ文献で、その系の周期解の厳密解を求めるための方程式が示されている。その方程式の解に対して精度保証を行った例を示す。パラメータの微小変化が解に大きな変化をもたらすため、精度保証がかなり難しい例題であると思われる。

その系は次のようなものである。

$$\varepsilon \dot{x} = y - f(x) \quad (6.10)$$

$$\dot{y} = -(x - a) \quad (6.11)$$

表 6.1: 区間反復 (小数点以下 15 桁表示)

k	I_k	p_k, r_k
0	$([1.345454545454545, 1.354545454545454])$ $([.635454545454545, .644545454545454])$	$(.000168350168350)$ $(.000113636363636)$
1	$([1.349206349206349, 1.349593495934959])$ $([.637426900584795, .638095238095238])$	$(.000009773002213)$ $(.000006596776494)$
2	$([1.349407114624505, 1.349436392914653])$ $([.637729549248747, .637770897832817])$	$(.000000653952539)$ $(.000000441417963)$
3	$([1.349420849420849, 1.349422875131164])$ $([.637749737118822, .637752318603382])$	$(.000000042659211)$ $(.000000028794967)$
4	$([1.349421999761649, 1.349422121085094])$ $([.637750909731189, .637751094153293])$	$(.000000002830977)$ $(.000000001910909)$
5	$([1.349422057451857, 1.349422066549912])$ $([.637750993910724, .637751006043836])$	$(.000000000196584)$ $(.000000000132694)$
6	$([1.349422061375974, 1.349422061869446])$ $([.637750999622577, .637751000448591])$	$(.000000000012217)$ $(.000000000008246)$
7	$([1.349422061612589, 1.349422061662260])$ $([.637751000017424, .637751000072420])$	$(.000000000000969)$ $(.000000000000654)$
8	$([1.349422061635065, 1.349422061637829])$ $([.637751000042774, .637751000046756])$	$(.000000000000062)$ $(.000000000000042)$
9	$([1.349422061636302, 1.349422061636556])$ $([.637751000044560, .637751000044834])$	$(.000000000000004)$ $(.000000000000003)$
10	$([1.349422061636419, 1.349422061636440])$ $([.637751000044690, .637751000044711])$	$(.000000000000000)$ $(.000000000000000)$
11	$([1.349422061636428, 1.349422061636430])$ $([.637751000044700, .637751000044701])$	$(.000000000000000)$ $(.000000000000000)$
12	$([1.349422061636429, 1.349422061636429])$ $([.637751000044701, .637751000044701])$	—

但し,

$$f(x) = x + 0.5(1+k)(|x-1| - |x+1|) \quad (6.12)$$

パラメータを, $\varepsilon = 0.2$, $k = 0.885$, $a = 0.9999999999$ としたとき, 周期解が $x = 1$ を横切るときの y の値 y_p, y_2 を未知数として, 次のような方程式が導ける.

$$G_1(y_p, y_2) = 0 \quad (6.13)$$

$$G_2(y_p, y_2) = 0 \quad (6.14)$$

表 6.2: 区間反復 (有理数表示)

k	I_k
0	$\left(\begin{array}{l} [74/55, 149/110] \\ [699/1100, 709/1100] \end{array} \right)$
1	$\left(\begin{array}{l} [85/63, 166/123] \\ [109/171, 67/105] \end{array} \right)$
2	$\left(\begin{array}{l} [1707/1265, 838/621] \\ [382/599, 206/323] \end{array} \right)$
3	$\left(\begin{array}{l} [699/518, 1286/953] \\ [1213/1902, 1169/1833] \end{array} \right)$
4	$\left(\begin{array}{l} [11323/8391, 22534/16699] \\ [5433/8519, 5683/8911] \end{array} \right)$
5	$\left(\begin{array}{l} [114292/84697, 19263/14275] \\ [50691/79484, 252617/396106] \end{array} \right)$
6	$\left(\begin{array}{l} [36541/27079, 100984/74835] \\ [116593/182819, 143589/225149] \end{array} \right)$
7	$\left(\begin{array}{l} [210607/156072, 174066/128993] \\ [256212/401743, 519570/814691] \end{array} \right)$
8	$\left(\begin{array}{l} [384673/285065, 1712758/1269253] \\ [2191697/3436603, 1677685/2630627] \end{array} \right)$
9	$\left(\begin{array}{l} [7656919/5674221, 15872577/11762500] \\ [10705449/16786252, 5675570/8899351] \end{array} \right)$
10	$\left(\begin{array}{l} [38458661/28500098, 6887573/5104091] \\ [23346383/36607364, 57011912/89395253] \end{array} \right)$
11	$\left(\begin{array}{l} [66008953/48916462, 79784099/59124644] \\ [60236397/94451278, 56624815/88788281] \end{array} \right)$
12	$\left(\begin{array}{l} [1062261475/787197353, 3360127769/2490049529] \\ [180064294/282342629, 138917801/217824513] \end{array} \right)$

但し, $p_1 = a - (1 - k)$, $p_2 = -ka$, $\lambda_1, \lambda_2 = \frac{1 \pm \sqrt{1-4\epsilon}}{2\epsilon}$, $\delta = \frac{k}{2\epsilon}$, $\omega = \frac{\sqrt{4\epsilon-k^2}}{2\epsilon}$ として,

$$G_1(y_1, y_2) = \left(\frac{\lambda_1(y_2 - p_1) - (1 - a)}{\lambda_1(y_1 - p_1) - (1 - a)} \right)^{\lambda_1} - \left(\frac{\lambda_2(y_2 - p_1) - (1 - a)}{\lambda_2(y_1 - p_1) - (1 - a)} \right)^{\lambda_2} \quad (6.15)$$

$$G_2(y_3, y_2) = \sqrt{A_1^2 + B_1^2} \exp\left(\frac{\delta}{\omega} \left(\tan^{-1} \frac{B_1}{A_1} + \frac{\pi}{2}\right)\right) - \sqrt{A_2^2 + B_2^2} \exp\left(\frac{\delta}{\omega} \left(\tan^{-1} \frac{B_2}{A_2} - \frac{\pi}{2}\right)\right) \quad (6.16)$$

$$A_1 = -(y_2 - p_2), \quad \omega B_1 = \delta(y_2 - p_2) + (1 - a) \quad (6.17)$$

$$A_2 = y_3 - p_2, \quad \omega B_2 = -\delta(y_3 - p_2) - (1 - a)$$

である. この方程式は, CALC のプログラムで例えば次の図 6.2 のように書ける.

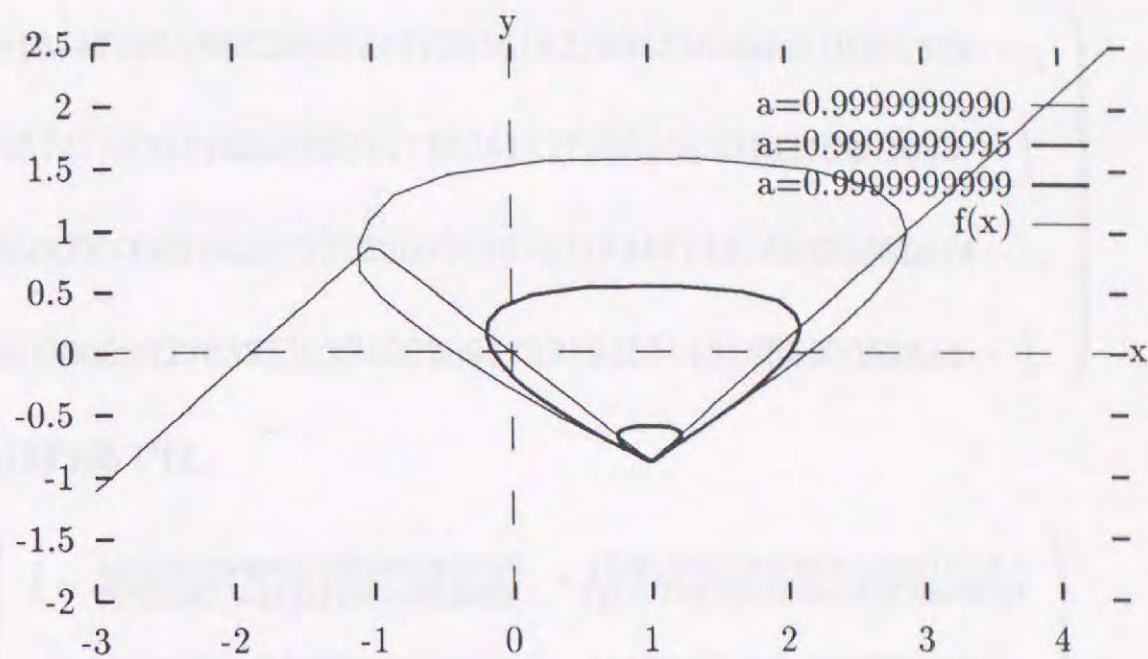


図 6.1: Lost Solution

この方程式に対して、近似解

$$\begin{pmatrix} -.6045747895188622869341136361827905556304 \\ -.8850000000276393202250020967531945311817 \end{pmatrix} \quad (6.18)$$

を与えると、区間

$$\begin{pmatrix} [-.60457478951886228693411363618279055563049359892311 \dots, \\ -.60457478951886228693411363618279055563030640107688 \dots] \\ [-.88500000002763932022500209675319453118179359892311 \dots, \\ -.88500000002763932022500209675319453118160640107688 \dots] \end{pmatrix} \quad (6.19)$$

に真の解が唯一存在することが保証できた。更に、区間の半径が $1e - 50$ 以下になるように区間

反復を行うと、保証区間

$$\left(\begin{array}{l} [-.60457478951886228693411363618279055563040210401689\dots, \\ -.60457478951886228693411363618279055563040210401689\dots] \\ [-.88500000002763932022500209675319453118165320053844\dots, \\ -.88500000002763932022500209675319453118165320053844\dots] \end{array} \right) \quad (6.20)$$

が得られた。これは内部では、

$$\left(\begin{array}{l} \left[-\frac{438866869404817939048569295}{725909973444444488933036089}, -\frac{772573485794623044466813216}{1277879096495999891973000933} \right] \\ \left[-\frac{631759001435395993088480108}{713851979000752101296753501}, -\frac{118821492123645216705963747}{134261573016874935701059706} \right] \end{array} \right) \quad (6.21)$$

という有理数で表現されている。

なお、上で用いた近似解は、軌道を描かせて求めた値を初期値として、CALCで記述した高精度 Newton 法により求めた。この近似解は 40 桁の精度を持っているが、後で調べたところ精度保証を行うためには最低 36 桁の有効数字が必要なことが分かった。

6.5 むすび

本章では、精度保証を行うためのソフトウェアについて議論し、第 3-5 章で提案された精度保証アルゴリズムが実際に実現できることを、システムの試作によって示した。

本論文の手法を任意精度計算が出来ないような数値体系の下で使用した場合を考える。本論文で保証した事柄は、

- (1) 解が包み込まれたときそこに数学的に厳密な意味で唯一解が存在することの保証
- (2) 解が包み込まれたとき任意にその解の精度を高められることの保証
- (3) 十分良い近似解に対して真解の包み込みが可能であることの保証

の 3 つである。任意精度計算が出来ない場合は、(2)、(3) の保証は失われるが、(1) は保たれる。

すなわち、その様な場合でも、必ず出来るとは言えないがその数値体系下で可能な範囲で解の存在保証は出来、またその数値体系下で可能な範囲で反復改良も出来る。

また、現在試作中のシステムは基本的にインタプリタであるため、速度に限界がある。より高速な実用システムを実現する場合に発生する問題点についても考えていきたい。

```
1000 0.0, 0.0, 0.0
1001 0.0, 0.0, 0.0
1002 0.0, 0.0, 0.0
1003 0.0, 0.0, 0.0
1004 0.0, 0.0, 0.0
1005 0.0, 0.0, 0.0
1006 0.0, 0.0, 0.0
1007 0.0, 0.0, 0.0
1008 0.0, 0.0, 0.0
1009 0.0, 0.0, 0.0
1010 0.0, 0.0, 0.0
1011 0.0, 0.0, 0.0
1012 0.0, 0.0, 0.0
1013 0.0, 0.0, 0.0
1014 0.0, 0.0, 0.0
1015 0.0, 0.0, 0.0
1016 0.0, 0.0, 0.0
1017 0.0, 0.0, 0.0
1018 0.0, 0.0, 0.0
1019 0.0, 0.0, 0.0
1020 0.0, 0.0, 0.0
1021 0.0, 0.0, 0.0
1022 0.0, 0.0, 0.0
1023 0.0, 0.0, 0.0
1024 0.0, 0.0, 0.0
1025 0.0, 0.0, 0.0
1026 0.0, 0.0, 0.0
1027 0.0, 0.0, 0.0
1028 0.0, 0.0, 0.0
1029 0.0, 0.0, 0.0
1030 0.0, 0.0, 0.0
1031 0.0, 0.0, 0.0
1032 0.0, 0.0, 0.0
1033 0.0, 0.0, 0.0
1034 0.0, 0.0, 0.0
1035 0.0, 0.0, 0.0
1036 0.0, 0.0, 0.0
1037 0.0, 0.0, 0.0
1038 0.0, 0.0, 0.0
1039 0.0, 0.0, 0.0
1040 0.0, 0.0, 0.0
1041 0.0, 0.0, 0.0
1042 0.0, 0.0, 0.0
1043 0.0, 0.0, 0.0
1044 0.0, 0.0, 0.0
1045 0.0, 0.0, 0.0
1046 0.0, 0.0, 0.0
1047 0.0, 0.0, 0.0
1048 0.0, 0.0, 0.0
1049 0.0, 0.0, 0.0
1050 0.0, 0.0, 0.0
1051 0.0, 0.0, 0.0
1052 0.0, 0.0, 0.0
1053 0.0, 0.0, 0.0
1054 0.0, 0.0, 0.0
1055 0.0, 0.0, 0.0
1056 0.0, 0.0, 0.0
1057 0.0, 0.0, 0.0
1058 0.0, 0.0, 0.0
1059 0.0, 0.0, 0.0
1060 0.0, 0.0, 0.0
1061 0.0, 0.0, 0.0
1062 0.0, 0.0, 0.0
1063 0.0, 0.0, 0.0
1064 0.0, 0.0, 0.0
1065 0.0, 0.0, 0.0
1066 0.0, 0.0, 0.0
1067 0.0, 0.0, 0.0
1068 0.0, 0.0, 0.0
1069 0.0, 0.0, 0.0
1070 0.0, 0.0, 0.0
1071 0.0, 0.0, 0.0
1072 0.0, 0.0, 0.0
1073 0.0, 0.0, 0.0
1074 0.0, 0.0, 0.0
1075 0.0, 0.0, 0.0
1076 0.0, 0.0, 0.0
1077 0.0, 0.0, 0.0
1078 0.0, 0.0, 0.0
1079 0.0, 0.0, 0.0
1080 0.0, 0.0, 0.0
1081 0.0, 0.0, 0.0
1082 0.0, 0.0, 0.0
1083 0.0, 0.0, 0.0
1084 0.0, 0.0, 0.0
1085 0.0, 0.0, 0.0
1086 0.0, 0.0, 0.0
1087 0.0, 0.0, 0.0
1088 0.0, 0.0, 0.0
1089 0.0, 0.0, 0.0
1090 0.0, 0.0, 0.0
1091 0.0, 0.0, 0.0
1092 0.0, 0.0, 0.0
1093 0.0, 0.0, 0.0
1094 0.0, 0.0, 0.0
1095 0.0, 0.0, 0.0
1096 0.0, 0.0, 0.0
1097 0.0, 0.0, 0.0
1098 0.0, 0.0, 0.0
1099 0.0, 0.0, 0.0
1100 0.0, 0.0, 0.0
1101 0.0, 0.0, 0.0
1102 0.0, 0.0, 0.0
1103 0.0, 0.0, 0.0
1104 0.0, 0.0, 0.0
1105 0.0, 0.0, 0.0
1106 0.0, 0.0, 0.0
1107 0.0, 0.0, 0.0
1108 0.0, 0.0, 0.0
1109 0.0, 0.0, 0.0
1110 0.0, 0.0, 0.0
1111 0.0, 0.0, 0.0
1112 0.0, 0.0, 0.0
1113 0.0, 0.0, 0.0
1114 0.0, 0.0, 0.0
1115 0.0, 0.0, 0.0
1116 0.0, 0.0, 0.0
1117 0.0, 0.0, 0.0
1118 0.0, 0.0, 0.0
1119 0.0, 0.0, 0.0
1120 0.0, 0.0, 0.0
1121 0.0, 0.0, 0.0
1122 0.0, 0.0, 0.0
1123 0.0, 0.0, 0.0
1124 0.0, 0.0, 0.0
1125 0.0, 0.0, 0.0
1126 0.0, 0.0, 0.0
1127 0.0, 0.0, 0.0
1128 0.0, 0.0, 0.0
1129 0.0, 0.0, 0.0
1130 0.0, 0.0, 0.0
1131 0.0, 0.0, 0.0
1132 0.0, 0.0, 0.0
1133 0.0, 0.0, 0.0
1134 0.0, 0.0, 0.0
1135 0.0, 0.0, 0.0
1136 0.0, 0.0, 0.0
1137 0.0, 0.0, 0.0
1138 0.0, 0.0, 0.0
1139 0.0, 0.0, 0.0
1140 0.0, 0.0, 0.0
1141 0.0, 0.0, 0.0
1142 0.0, 0.0, 0.0
1143 0.0, 0.0, 0.0
1144 0.0, 0.0, 0.0
1145 0.0, 0.0, 0.0
1146 0.0, 0.0, 0.0
1147 0.0, 0.0, 0.0
1148 0.0, 0.0, 0.0
1149 0.0, 0.0, 0.0
1150 0.0, 0.0, 0.0
1151 0.0, 0.0, 0.0
1152 0.0, 0.0, 0.0
1153 0.0, 0.0, 0.0
1154 0.0, 0.0, 0.0
1155 0.0, 0.0, 0.0
1156 0.0, 0.0, 0.0
1157 0.0, 0.0, 0.0
1158 0.0, 0.0, 0.0
1159 0.0, 0.0, 0.0
1160 0.0, 0.0, 0.0
1161 0.0, 0.0, 0.0
1162 0.0, 0.0, 0.0
1163 0.0, 0.0, 0.0
1164 0.0, 0.0, 0.0
1165 0.0, 0.0, 0.0
1166 0.0, 0.0, 0.0
1167 0.0, 0.0, 0.0
1168 0.0, 0.0, 0.0
1169 0.0, 0.0, 0.0
1170 0.0, 0.0, 0.0
1171 0.0, 0.0, 0.0
1172 0.0, 0.0, 0.0
1173 0.0, 0.0, 0.0
1174 0.0, 0.0, 0.0
1175 0.0, 0.0, 0.0
1176 0.0, 0.0, 0.0
1177 0.0, 0.0, 0.0
1178 0.0, 0.0, 0.0
1179 0.0, 0.0, 0.0
1180 0.0, 0.0, 0.0
1181 0.0, 0.0, 0.0
1182 0.0, 0.0, 0.0
1183 0.0, 0.0, 0.0
1184 0.0, 0.0, 0.0
1185 0.0, 0.0, 0.0
1186 0.0, 0.0, 0.0
1187 0.0, 0.0, 0.0
1188 0.0, 0.0, 0.0
1189 0.0, 0.0, 0.0
1190 0.0, 0.0, 0.0
1191 0.0, 0.0, 0.0
1192 0.0, 0.0, 0.0
1193 0.0, 0.0, 0.0
1194 0.0, 0.0, 0.0
1195 0.0, 0.0, 0.0
1196 0.0, 0.0, 0.0
1197 0.0, 0.0, 0.0
1198 0.0, 0.0, 0.0
1199 0.0, 0.0, 0.0
1200 0.0, 0.0, 0.0
```



```

define lost(x)
{
    local k, ep, a;
    local y1, y2, y3;
    local delta, omega;
    local l1, l2;
    local p1, p2;
    local A1, B1, A2, B2;
    local G1, G2;
    local y, tmp;

    k = 0.885; ep = 0.2; a = 0.9999999999;

    y1 = x[0]; y2 = x[1]; y3 = y1;

    p1 = a - (1+k);
    p2 = -k * a;

    delta = k / (2 * ep);
    omega = Sqrt(4 * ep - k^2) / (2 * ep);

    tmp = Sqrt(1 - 4 * ep);
    l1 = - (-1 - tmp) / (2 * ep);
    l2 = - (-1 + tmp) / (2 * ep);

    A1 = -(y2 - p2);
    A2 = y3 - p2;
    B1 = (delta * (y2 - p2) + (1-a)) / omega;
    B2 = (-delta * (y3 - p2) - (1-a)) / omega;

    tmp = Power((l1*(y2-p1)-(1-a)) / (l1*(y1-p1)-(1-a)), l1);
    G1 = tmp - Power((l2*(y2-p1)-(1-a)) / (l2*(y1-p1)-(1-a)), l2);

    tmp = Sqrt(A1^2 + B1^2) * Exp(delta/omega*(Atan(B1/A1)+Pi()/2));
    G2 = tmp - Sqrt(A2^2 + B2^2) * Exp(delta/omega*(Atan(B2/A2)-Pi()/2));

    mat y[2];
    y[0] = G1; y[1] = G2;
    return y;
}

```

図 6.2: プログラム例

7.1 はじめに

本章では、方程式に幾つかのパラメータが付加された場合など、定義域の次元が値域の次元よりも高いような方程式の解の包み込みについて論ずる。この次元の差が n ならば解の集合は一般に n 次元多様体となるが、これを包みこむ事を考える。これは、パラメータによる分岐現象等の解析や変動が加わった方程式の解析など、様々な非線形現象の解析の基礎となる。従来、通常の方程式の解 (0 次元多様体) の包み込みの方法は盛んに研究されてきたが 1 次元多様体以上の解の包み込みの方法についてはあまり議論されていなかった。

まず、これ以降の議論の基礎となる、簡易 Newton 法の収束定理である占部の定理について議論する。次に、陰関数定理を拡張し、その成立範囲を見積もれるような構成的陰関数定理を占部の定理を元に導く。更に、導関数の Lipschitz 連続性を仮定し、同様な定理を導く。

また、構成的陰関数定理の応用例の一つとして、ホモトピー法の解曲線追跡の新しい方法を提案する。まず、ホモトピー法について簡単に説明する。次に、解曲線追跡法の一つである予測子修正子法の改良として、決して失敗しないような新しい解曲線追跡法を構成的陰関数定理から導く。また、その解曲線追跡法について実際の適用に関する詳細な議論を展開する。最後に、シミュレーションによってその有効性を示す。

7.2 占部の定理

本節では、簡易 Newton 法の収束条件を与える占部の定理と、その誤差評価に関する拡張について述べる。更に、占部の定理の実際の適用に関するいくつかの性質について述べる。

Newton 法の反復は

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k) \quad (7.1)$$

で与えられる。 $f'(x_k)^{-1}$ の計算に手間がかかる場合は、ほかの作用素 $A(x_k)$ で置き換えることがあり、この方法は広義 Newton 法と呼ばれる。特に $A(x_k) = f'(x_0)$ としたものは、簡易 Newton

法と呼ばれる。

Newton 法の収束条件については, Kantorovich 以来多くの人によって研究がなされている [18]. これらの定理は, 反復の収束の保証のみならず, 解の存在の保証, 一意性領域の決定などを行える. また, 無限次元の Banach 空間においても成立するという特徴も極めて重要である.

占部が広義 Newton 法の収束について導いた占部の定理 [19] もそのような定理の一つである. これは, 簡易 Newton 法を含むある種の広義 Newton 法の収束を保証するものである.

定理 7.1 (占部の定理) [19] X, Y を Banach 空間, $U \subset X$ を非空な開集合, $f: U \rightarrow Y$ を C^1 級とする. $f(x) = 0$ の近似解 $x_0 \in U$ と, 線形作用素 $L: X \rightarrow Y$ (x_0 における f の線形成分の近似) が与えられ, ある $\delta > 0$ と非負の数 $\varepsilon < 1$ が存在して次の仮定 (A)-(C) が満たされているとする:

仮定 (A) L^{-1} が存在して有界, すなわち,

$$\|L^{-1}\| \leq M \quad (M > 0). \quad (7.2)$$

仮定 (B) $B(x_0; \delta) \subset U$ かつ, そこで

$$\|f'(x) - L\| \leq \frac{\varepsilon}{M}. \quad (7.3)$$

仮定 (C) $\|f(x_0)\| \leq r$ とするとき,

$$\frac{Mr}{1-\varepsilon} \leq \delta. \quad (7.4)$$

このとき, 広義 Newton 法

$$x_{k+1} = x_k - L^{-1}f(x_k) \quad (k \geq 0) \quad (7.5)$$

につき, 次のことが成り立つ:

$$(1) \quad x_k \in B(x_0, \frac{Mr}{1-\varepsilon}) \quad (k \geq 0).$$

(2) $\{x_k\}$ は $f(x) = 0$ の解 $x^* \in X$ に収束する.

(3) x^* は $B(x_0; \delta)$ 内における一意解である.

□

この占部の定理は、ほぼオリジナル通りであるが、いくらか無駄がある上、縮小写像原理で当然導かれるはずの誤差評価がない。書き直し、誤差評価を加えたものを次に示す。前提条件はオリジナルの占部の定理とかなり異なるが、オリジナルを含むより緩い条件となっている。

定理 7.2 (占部の定理と誤差評価) X, Y を Banach 空間, $U \subset X$ を非空な開集合, $f: U \rightarrow Y$ を C^1 級とする. $f(x) = 0$ の近似解 $x_0 \in U$ と, 線形作用素 $L: X \rightarrow Y$ (x_0 における f の線形成分の近似) が与えられ, ある $\delta > 0$ が存在して次の仮定 (A)-(D) が満たされているとする:

仮定 (A) $B(x_0; \delta) \subset U$

仮定 (B) $B(x_0; \delta)$ で $\|f'(x) - L\|$ が有界, すなわちある $K_0 > 0$ が存在して

$$\|f'(x) - L\| \leq K_0 \quad (\forall x \in B(x_0; \delta)) \quad (7.6)$$

仮定 (C) $L^{-1}: Y \rightarrow X$ が存在し有界線形作用素となり, 次の条件を満たす:

$$\|L^{-1}f(x_0)\| + \|L^{-1}\|K_0\delta \leq \delta \quad (7.7)$$

仮定 (D) $\|L^{-1}\|K_0 < 1$

このとき, 広義 Newton 法

$$x_{k+1} = x_k - L^{-1}f(x_k) \quad (k \geq 0) \quad (7.8)$$

につき, 次のことが成り立つ:

(1) $f(x) = 0$ の解 x^* は $B(x_0; \delta)$ にただ一つ存在.

$$(2) x_k \in B(x_0; \delta) \quad (k \geq 0).$$

$$(3) x_k \rightarrow x^* \quad (k \rightarrow \infty).$$

(4) (事後誤差評価)

$$\|x_k - x^*\| \leq \frac{\|L^{-1}f(x_k)\|}{1 - \|L^{-1}\|K_0} \quad (k \geq 0). \quad (7.9)$$

(5) (事前誤差評価)

$$\|x_k - x^*\| \leq \frac{(\|L^{-1}\|K_0)^k}{1 - \|L^{-1}\|K_0} \|L^{-1}f(x_0)\| \quad (k \geq 0). \quad (7.10)$$

□

占部は点列 $\{x_k\}$ の有界性を導くために数学的帰納法を用いた複雑な証明を行っていたが、縮小写像の原理に持ち込めばはるかに見通しが良くなる。

証明は、作用素 $g: B(x_0; \delta) \rightarrow X$ を

$$g(x) = x - L^{-1}f(x) \quad (7.11)$$

で定義したとき、

(1) g は縮小定数を $\|L^{-1}\|K_0$ とする縮小写像

(2) $g(B(x_0; \delta)) \subset B(x_0; \delta)$

が成立することにより行われる。上の2つが成立するとき、(1)–(5) が成立することは容易に分かる。以下、仮定 (C), (D) の導出を行う。

Lipschitz 定数は、次の補題を利用すると容易に分かる。

補題 7.1 (微分の有界性と Lipschitz 連続性) X, Y を Banach 空間, $D \subset X$ を凸集合, $f: D \rightarrow Y$ を C^1 級とする。このとき、

$$\|f'(x)\| \leq \alpha \quad (\forall x \in D) \quad (7.12)$$

ならば,

$$\|f(x) - f(y)\| \leq \alpha \|x - y\| \quad (\forall x, y \in D) \quad (7.13)$$

が成立する. □

この補題により, g の Lipschitz 定数を見積もってみる.

$$g'(x) = I - L^{-1}f'(x) = L^{-1}L - L^{-1}f'(x) \quad (7.14)$$

$$= L^{-1}(L - f'(x)) \quad (7.15)$$

であるから, 仮定 (B) より

$$\|g'(x)\| \leq \|L^{-1}\| \|L - f'(x)\| \quad (7.16)$$

$$\leq \|L^{-1}\| K_0 \quad (7.17)$$

となる. ここで補題 7.1 を用いると,

$$\|g(z_1) - g(z_2)\| \leq \|L^{-1}\| K_0 \|z_1 - z_2\| \quad (7.18)$$

となる. よって, 仮定 (D) を設けると, $g : B(x_0; \delta) \rightarrow X$ は縮小写像となることがわかる.

次に, $g(B(x_0; \delta)) \subset B(x_0; \delta)$ を示す. $x \in B(x_0; \delta)$ ならば $g(x) \in B(x_0; \delta)$ となることを示せばよい.

$$\|x_0 - g(x)\| = \|x_0 - g(x_0) + g(x_0) - g(x)\| \quad (7.19)$$

$$\leq \|x_0 - g(x_0)\| + \|g(x_0) - g(x)\| \quad (7.20)$$

$$\leq \|L^{-1}f(x_0)\| + \|L^{-1}\| K_0 \delta \quad (7.21)$$

となるから, 仮定 (C) を設ければよいことが分かる.

占部の定理を, 非線形方程式 $f(x) = 0$ の数値解法に応用するとき, 問題となるのはある領域についての条件となっている仮定 (B) である. すなわちこれが満たされることを保証するために

は, f に何らかの大域的な制約を加える必要がある. f の導関数が Lipschitz 連続という制約を加えれば, 明らかに仮定 (B) は満たされ, K_0 を評価することができる. これを次の補題 7.2 に示す (証明は明らか).

定義 7.1 (α -Lipschitz 連続) X, Y を Banach 空間, $D \subset X$ とする. このとき, $f: D \rightarrow Y$ が α -Lipschitz 連続であるとは, $\forall x, y \in D$ について

$$\|f(x) - f(y)\| \leq \alpha \|x - y\| \quad (7.22)$$

が成立することをいう. □

補題 7.2 (Lipschitz 連続 \Rightarrow 定理 7.2 の仮定 (B)) 定理 7.2 において, f のフレッシュェ導関数 f' が $B(x_0; \delta)$ で α -Lipschitz 連続, すなわち

$$\|f'(x_1) - f'(x_2)\| \leq \alpha \|x_1 - x_2\| \quad (\forall x_1, x_2 \in B(x_0; \delta)) \quad (7.23)$$

であれば, $L = f'(x_0)$ とすることにより仮定 (B) は満たされ, $K_0 = \alpha\delta$ と評価できる. □

導関数 f' の Lipschitz 連続性は, 補題 7.1 により, 定義域 D が凸で 2 次導関数 f'' が有界であれば成立する. すなわち,

$$\sup_{x \in D} \|f''(x)\| \leq M \quad (7.24)$$

であれば, $\alpha = M$ とできる. 定義域における導関数の Lipschitz 定数 α が計算できた場合, 仮定 (C), (D) は,

$$\|f'(x_0)^{-1} f(x_0)\| + \|f'(x_0)^{-1}\| \alpha \delta^2 \leq \delta \quad (7.25)$$

$$\|f'(x_0)^{-1}\| \alpha \delta < 1 \quad (7.26)$$

となる. これを満たす δ が存在すれば, 近似解 x_0 の δ 近傍に真解が唯一存在することが保証され, Newton 法による反復改良の成功が保証される.

7.3 構成的陰関数定理

ここでは、陰関数定理を拡張し、その成立範囲を見積もれるような構成的陰関数定理を導く。陰関数定理は、局所的な陰関数の存在を示すものであるが、その具体的な成立範囲を与えてはいない。陰関数定理の証明方法の一つとして縮小写像原理によるものが知られているが、Newton法も同様に縮小写像原理によって支配されている。Newton法の収束定理が縮小写像原理等の不動点定理によって証明され、その唯一解領域を特定出来る以上、陰関数定理においても同様に成立範囲を特定し、またその範囲での陰関数の計算方法を与えるようなものができるのとは極めて自然なことである。

本節では、簡易Newton法の収束定理である占部の定理を用いて、構成的な陰関数定理を導く。これによれば、導関数のLipschitz連続性を仮定した上で、陰関数の存在を保証できる近傍の大きさを見積もることが可能であり、その近傍内における陰関数はNewton法によって計算が可能である。

通常陰関数定理は次のような形に書ける。

定理 7.3 (陰関数定理) X, Y, Z を Banach 空間, $U \subset X \times Y$ を開集合, $g: U \rightarrow Z$ を C^1 級とする。ここで、ある元 $(p_x, p_y) \in X \times Y$ に対して

(1) $g(p_x, p_y) = 0$

(2) g の (p_x, p_y) における Y に関する偏微分 $D_Y g(p_x, p_y): Y \rightarrow Z$ が同相写像

が成立するとする。

このとき、十分小さい $\varepsilon > 0, \delta > 0$ に対して、以下が成立する:

全ての固定された $x \in B_X(p_x; \varepsilon)$ (X 内における半径 ε , 中心 p_x の閉球) に対して、

(y を変数とする) 方程式

$$g(x, y) = 0, y \in B_Y(p_y; \delta) \tag{7.27}$$

は唯一解を持つ.

□

ここで, x に対して y を対応させる写像は g の陰関数と呼ばれ, 通常陰関数定理には陰関数の連続性や微分可能性についての記述も含まれる. しかし, 陰関数定理が成立する近傍 $B_X(p_x; \varepsilon) \times B_Y(p_y; \delta)$ の大きさを見積もることは出来ない.

ここで, 占部の定理 (定理 7.2) を元にして, 成立する範囲を見積もれるような陰関数定理を導く. $X \times Y$ のノルムは, 次の仮定を満たすとする.

仮定 7.1 (直積空間のノルム) Banach 空間 $X \times Y$ のノルム $\|\cdot\|_{X \times Y}$ は,

$$\|(x, 0)\|_{X \times Y} = \|x\|_X \quad (\forall x \in X) \quad (7.28)$$

$$\|(0, y)\|_{X \times Y} = \|y\|_Y \quad (\forall y \in Y) \quad (7.29)$$

を満たす.

□

定理 7.4 (構成的陰関数定理 I) X, Y, Z を Banach 空間, $U \subset X \times Y$ を開集合, $g: U \rightarrow Z$ を C^1 級とし, 仮定 7.1 が満たされているとする. また, ある元 $(p_x, p_y) \in U$ と, 線形作用素 $A: X \times Y \rightarrow Z$ に対し, ある実数 R, K_0, d が存在して $\varepsilon > 0, \delta > 0$ が

仮定 (A) $\forall x \in B_X(p_x; \varepsilon)$ について,

$$\|g(x, p_y)\| \leq R \quad (7.30)$$

仮定 (B) $\forall (x, y) \in B_X(p_x; \varepsilon) \times B_Y(p_y; \delta)$ について,

$$\|D_Y g(x, y) - A_Y\| \leq K_0 \quad (7.31)$$

仮定 (C) A_Y^{-1} が存在し,

$$\|A_Y^{-1}\| \leq d \quad (7.32)$$

仮定 (D)

$$d\left(\frac{R}{\delta} + K_0\right) \leq 1 \quad (7.33)$$

仮定 (E)

$$dK_0 < 1 \quad (7.34)$$

仮定 (F)

$$B_X(p_x; \varepsilon) \times B_Y(p_y; \delta) \subset U \quad (7.35)$$

を満たすとする. ただし, $A_X(x) = A(x, 0)$, $A_Y(y) = A(0, y)$ とする.

このとき, 全ての固定された $x \in B_X(p_x; \varepsilon)$ に対して, 以下が成立する:

(1) y を変数とする方程式 $g(x, y) = 0$ の解 y^* は $B_Y(p_y; \delta)$ にただ一つ存在.

(2) $y_k \in B_Y(p_y; \delta)$ ($k \geq 0$).

(3) $y_k \rightarrow y^*$ ($k \rightarrow \infty$).

(4) (事後誤差評価)

$$\|y_k - y^*\| \leq \frac{\|A_Y^{-1}g(x, y_k)\|}{1 - dK_0} \quad (k \geq 0) \quad (7.36)$$

(5) (事前誤差評価)

$$\|y_k - y^*\| \leq \frac{(dK_0)^k}{1 - dK_0} \|A_Y^{-1}g(x, y_0)\| \quad (k \geq 0) \quad (7.37)$$

但し, $\{y_k\}$ は

$$y_0 = p_y, \quad (7.38)$$

$$y_{k+1} = y_k - A_Y^{-1}g(x, y_k) \quad (7.39)$$

で生成される点列とする. □

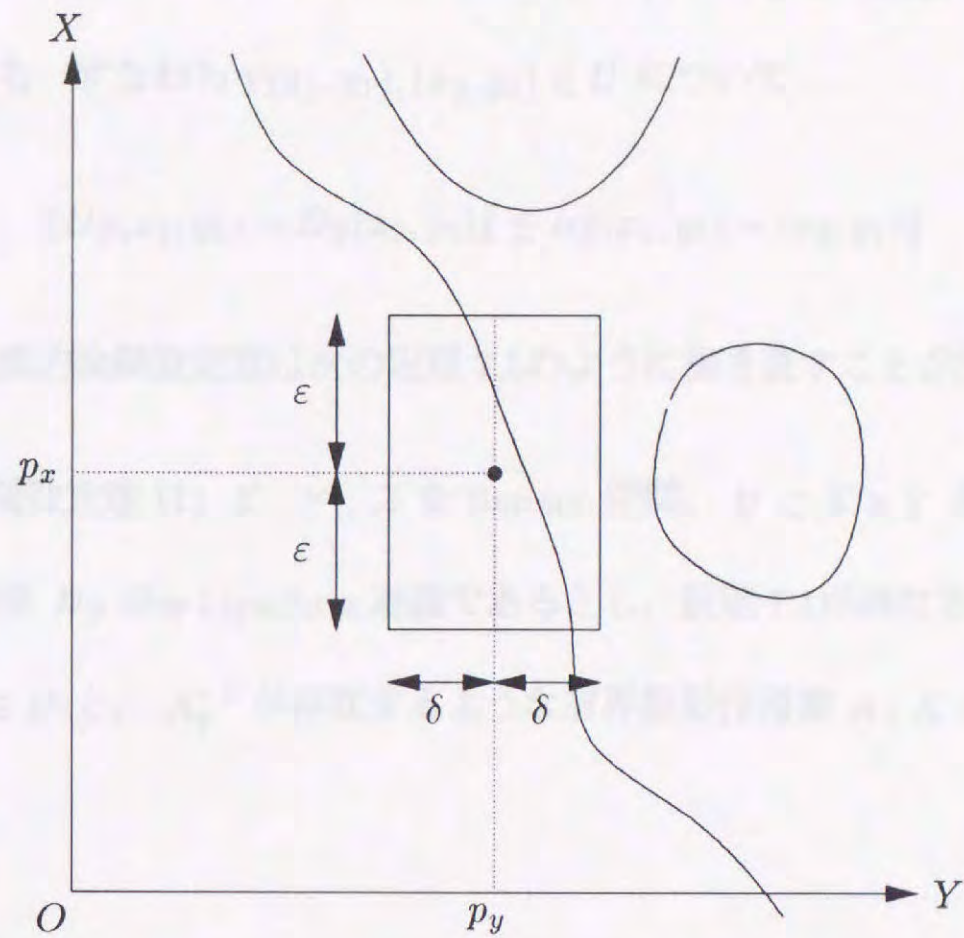


図 7.1: 構成的陰関数定理

図 7.1 にこの定理の様子を示す.

定理 7.4 の証明を示す.

(証明) $\forall x \in B_X(p_x; \varepsilon)$ に対して, $L = A_Y$, $x_0 = p_y$, $f = g(x, \cdot)$, $\delta = \delta$, $K_0 = K_0$ としたとき
占部の定理 (定理 7.2) の条件を満たすことを示す.

仮定 (A) は, 式 (7.35) により成立. 仮定 (B) は, 式 (7.31) により成立. 仮定 (C) は, 式 (7.32)
より L^{-1} の存在が分かり, 式 (7.30), 式 (7.33) より

$$\|L^{-1}f(x_0)\| + \|L^{-1}\|K_0\delta \leq dr + dK_0\delta \tag{7.40}$$

$$\leq \delta \tag{7.41}$$

となり, 式 (7.7) が満たされた. 仮定 (D) は, 式 (7.34) により成立.

よって占部の定理が成立し, (1)-(5) が示される. □

ε と δ を見積もりやすくするため, Newton 法の場合に従って g の導関数 Dg が α -Lipschitz 連続という制約を加える. すなわち $\forall (x_1, y_1), (x_2, y_2) \in U$ について

$$\|Dg(x_1, y_1) - Dg(x_2, y_2)\| \leq \alpha \|(x_1, y_1) - (x_2, y_2)\| \quad (7.42)$$

が成立するとき, 構成的陰関数定理は次の定理 7.5 のように書き直すことが出来る.

定理 7.5 (構成的陰関数定理 II) X, Y, Z を Banach 空間, $U \subset X \times Y$ を開集合, $g: U \rightarrow Z$ を C^1 級, g の導関数 Dg が α -Lipschitz 連続であるとし, 仮定 7.1 が満たされているとする. また, ある元 $(p_x, p_y) \in U$ と, A_Y^{-1} が存在するような有界線形作用素 $A: X \times Y \rightarrow Z$ に対し, r, K, d_1, d_2 を

$$(1) \|g(p_x, p_y)\| \leq r,$$

$$(2) \|Dg(p_x, p_y) - A\| \leq K,$$

$$(3) \|A_X\| \leq d_1,$$

$$(4) \|A_Y^{-1}\| \leq d_2$$

を満たす実数とする. ただし, $A_X(x) = A(x, 0)$, $A_Y(y) = A(0, y)$ とする.

このとき, $\varepsilon > 0$, $\delta > 0$ が

$$d_2 \left(\frac{\frac{1}{2}\alpha\varepsilon^2 + (K + d_1)\varepsilon + r}{\delta} + \alpha(\varepsilon + \delta) + K \right) \leq 1 \quad (7.43)$$

$$d_2(\alpha(\varepsilon + \delta) + K) < 1 \quad (7.44)$$

$$B_X(p_x; \varepsilon) \times B_Y(p_y; \delta) \subset U \quad (7.45)$$

を満たすならば, 全ての固定された $x \in B_X(p_x; \varepsilon)$ に対して, 以下が成立する:

(1) y を変数とする方程式 $g(x, y) = 0$ の解 y^* は $B_Y(p_y; \delta)$ にただ一つ存在.

(2) $y_k \in B_Y(p_y; \delta)$ ($k \geq 0$).

(3) $y_k \rightarrow y^*$ ($k \rightarrow \infty$).

(4) (事後誤差評価)

$$\|y_k - y^*\| \leq \frac{\|A_Y^{-1}g(x, y_k)\|}{1 - \theta} \quad (k \geq 0) \quad (7.46)$$

(5) (事前誤差評価)

$$\|y_k - y^*\| \leq \frac{\theta^k}{1 - \theta} \|A_Y^{-1}g(x, y_0)\| \quad (k \geq 0) \quad (7.47)$$

但し, $\theta = d_2(\alpha(\varepsilon + \delta) + K)$, $\{y_k\}$ は

$$y_0 = p_y, \quad (7.48)$$

$$y_{k+1} = y_k - A_Y^{-1}g(x, y_k) \quad (7.49)$$

で生成される点列とする. □

注意 7.1 定理 7.5 で $r = K = 0$ とすると, 定理 7.3 と同じ状況になる. □

定理 7.5 の証明の前に次の補題 [42] を示す.

補題 7.3 (線形化写像の精度) X, Y を Banach 空間, $U \subset X$ を凸集合, $f: U \rightarrow Y$ は C^1 級で導関数 f' が α -Lipschitz 連続であるとする. このとき, f の $x_0 \in U$ における線形化写像 $F: X \rightarrow Y$ を

$$F(x) = f(x_0) + Df(x_0)(x - x_0) \quad (7.50)$$

と定義すると, その精度は,

$$\|F(x) - f(x)\| \leq \frac{1}{2}\alpha\|x - x_0\|^2 \quad (\forall x \in U) \quad (7.51)$$

で与えられる. □

定理 7.5 の証明を示す.

(証明)

$$d = d_2 \quad (7.52)$$

$$R = \frac{1}{2}\alpha\varepsilon^2 + (K + d_1)\varepsilon + r \quad (7.53)$$

$$K_0 = K + \alpha(\varepsilon + \delta) \quad (7.54)$$

としたとき, 定理 7.4 の仮定 (A)–(F) を満たすことを示せばよい. 仮定 (C)–(F) は自明. 仮定 (A), (B) を示す.

まず, 仮定 (A) の成立を示す. $G : B_X(p_x; \varepsilon) \rightarrow Z$ を,

$$G(x) = g(x, p_y) \quad (7.55)$$

で定義する. このとき,

$$\|DG(x_1) - DG(x_2)\| = \|D_Y g(x_1, p_y) - D_Y g(x_2, p_y)\| \quad (7.56)$$

$$\leq \|Dg(x_1, p_y) - Dg(x_2, p_y)\| \quad (7.57)$$

$$\leq \alpha\|x_1 - x_2\| \quad (7.58)$$

より, DG は α -Lipschitz 連続となる. ここで,

$$\|g(x, p_y)\| = \|G(x)\| \quad (7.59)$$

$$\leq \|G(x) - G(p_x)\| + \|G(p_x)\| \quad (7.60)$$

$$\leq \|G(p_x) + DG(p_x)(x - p_x) - G(x)\| + \|DG(p_x)(x - p_x)\| + r \quad (7.61)$$

であるが, 補題 7.3 により上式の右辺第 1 項は,

$$\|G(p_x) + DG(p_x)(x - p_x) - G(x)\| \leq \frac{1}{2}\alpha\|x - p_x\|^2 \quad (7.62)$$

$$\leq \frac{1}{2}\alpha\varepsilon^2 \quad (7.63)$$

となる. 右辺第2項は,

$$\|DG(p_x)(x - p_x)\| = \|Dg(p_x, p_y)(x - p_x, 0)\| \quad (7.64)$$

$$= \|(Dg(p_x, p_y) - A + A)(x - p_x, 0)\| \quad (7.65)$$

$$\leq \|(Dg(p_x, p_y) - A)(x - p_x, 0)\| + \|A(x - p_x, 0)\| \quad (7.66)$$

$$\leq (K + d_1)\varepsilon \quad (7.67)$$

となる. よって $\|g(x, p_y)\|$ は

$$\|g(x, p_y)\| \leq \frac{1}{2}\alpha\varepsilon^2 + (K + d_1)\varepsilon + r \quad (7.68)$$

$$= R \quad (7.69)$$

と評価でき, 仮定 (A) が成立する.

仮定 (B) は,

$$\|D_Y g(x, y) - A_Y\| \leq \|Dg(x, y) - A\| \quad (7.70)$$

$$\leq \|Dg(x, y) - Dg(p_x, p_y)\| + \|Dg(p_x, p_y) - A\| \quad (7.71)$$

$$\leq \alpha\|(x, y) - (p_x, p_y)\| + K \quad (7.72)$$

$$\leq \alpha(\|(x, y) - (p_x, y)\| + \|(p_x, y) - (p_x, p_y)\|) + K \quad (7.73)$$

$$\leq \alpha(\varepsilon + \delta) + K \quad (7.74)$$

$$= K_0 \quad (7.75)$$

により示される.

以上で題意は示された. □

図 7.2 にこの定理の証明の概念図を示す.

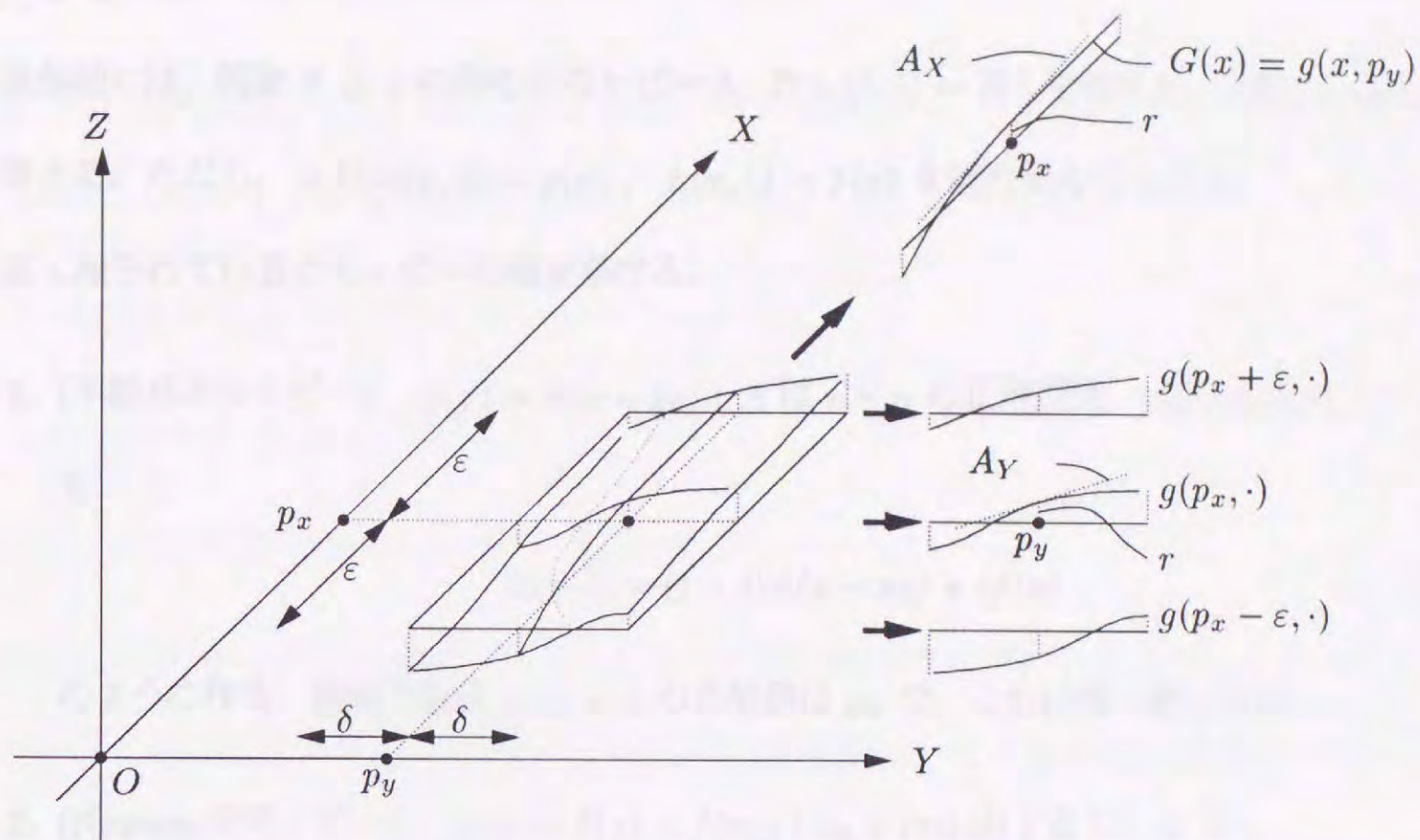


図 7.2: 構成的陰関数定理の証明

7.4 ホモトピー法

ホモトピー法は、有限次元の非線形方程式の数値的解法であり、関数がある条件を満たせば必ず解が求められる。この性質は Newton 法にはないものである。また、必ず解が求められる条件を確認できない場合でも、Newton 法に比べれば多くの初期点で解が求まることが経験的に知られている (大域的収束性があるという)。

今、 $D \subset \mathbb{R}^n$ を内点を持つ有界閉集合、 $f: D \rightarrow \mathbb{R}^n$ を連続写像とし、非線形方程式 $f(x) = 0$ を考える。この方程式の解を求めるために、ホモトピー法では補助方程式 $g(x) = 0$ を導入する。ただし、 $g: D \rightarrow \mathbb{R}^n$ は連続で性質のよく分かった関数 (少なくとも一つの自明解を持つ) であるとする。

ホモトピー法とは、簡単な関数 g から、パラメーター t を用いて徐々に複雑な関数 f に変形させていき、そのときの解の変化を追跡して最終的に $f(x) = 0$ の解を求めようというアイデアに

基づいている。

具体的には、関数 f と g の間のホモトピー $h: D \times [0, 1] \rightarrow \mathbb{R}^n$ を導入し、方程式 $h(x, t) = 0$ を考える。ただし、 h は $h(x, 0) = g(x)$ 、 $h(x, 1) = f(x)$ を満たすものとする。

良く知られているホモトピーの例を挙げる。

1. (不動点ホモトピー) $g(x) = A(x - x_0)$ (A は $n \times n$ の正則行列, $x_0 \in \text{int}(D)$) とし、 h を

$$h(x, t) = (1 - t)A(x - x_0) + tf(x) \quad (7.76)$$

のように作る。補助方程式 $g(x) = 0$ の自明解は x_0 で、これは唯一解となる。

2. (Newton ホモトピー) $g(x) = f(x) - f(x_0)$ ($x_0 \in \text{int}(D)$) とし、 h を

$$h(x, t) = f(x) - (1 - t)f(x_0) \quad (7.77)$$

のように作る。補助方程式 $g(x) = 0$ の自明解は x_0 (ただし唯一解とは限らない)。

3. (奇ホモトピー) $0 \in \text{int}(D)$ 、 D は原点对称な集合とする。 $g(x) = \frac{1}{2}(f(x) - f(-x))$ とし、 h を

$$h(x, t) = \frac{1+t}{2}f(x) - \frac{1-t}{2}f(-x) \quad (7.78)$$

のように作る。補助方程式 $g(x) = 0$ の自明解は 0 (唯一とは限らない)。

但し、 $\text{int}(D)$ は D の内点全体の集合を表わす。

$h(x, t) = 0$ の解集合を $D \times [0, 1]$ の中で見れば、曲線になっている。この曲線は $g(x) = 0$ の自明解 x_0 を通るから、そこから解曲線を追跡していき、 $t = 1$ 超平面に到達したならばその点が $f(x) = 0$ の解を与える。

$h(x, t) = 0$ の解曲線は、 0 が h の正則値ならば 1 次元多様体となり、また h が C^2 級ならば Sard の補題 [48] により h の正則値は \mathbb{R}^n で稠密となる。よって、一般に解曲線は途中で途切れたり枝分かれしたりすることがない。

ホモトピー法の特徴的な性質を示しておく。

定理 7.6 (解の存在) $D \subset \mathbb{R}^n$ は内点を持つ有界閉集合, $h : D \times [0, 1] \rightarrow \mathbb{R}^n$ は C^2 級で $h(x, 0) = g(x)$, $h(x, 1) = f(x)$, 0 は g の正則値とする. このとき,

1. $g(x) = 0$ が奇数個の解を持つ.
2. $h(x, t) \neq 0$ on $\partial D \times [0, 1]$.

ならば $f(x) = 0$ の解が少なくとも一つ存在する. □

これは, $g(x) = 0$ の解に連結する $h(x, t) = 0$ の解曲線のうち少なくとも一つは $t = 1$ 超平面に到達することから示される. すなわち, すべての解曲線が $t = 0$ 超平面に両端を持つとすると, $g(x) = 0$ の解は偶数個でなければならない. 0 が g の正則値であることは, 解曲線が $t = 0$ 超平面に接しないことを保証するために要求されている.

また, よく知られている Brouwer の不動点定理:

定理 7.7 (Brouwer の不動点定理) [49] \mathbb{R}^n の有界凸閉集合 D から自分自身への連続写像 f は, D の中に不動点を持つ. □

は, 背理法によって証明され, その不動点を求める方法を与えるものではないが, ホモトピー法を用いると不動点を求める方法を同時に与えるような証明, いわゆる構成的証明が行える. D が内点を持ち, f が C^2 級であると仮定した上で, $x_0 \in \text{int}(D)$ として, ホモトピー h を

$$h(x, t) = (1 - t)x_0 + tf(x) - x \tag{7.79}$$

と定めればよい.

7.5 予測子修正子法の改良

ホモトピー法の解曲線の追跡法は, 微分を用いる方法 (予測子修正子法など) と, ホモトピー方程式を区分的線形近似してその解曲線を追跡する方法 (単体法など) に大別できる. 前者は, 解曲

線の接線を求めてそれに沿って解を進め、実際の解曲線とずれた分を Newton 法などを用いて修正しながら解曲線を追跡するという手法である。しかし、これら微分を用いる方法は、解曲線の追跡の成功を完全に保証することができないという欠点を有している。すなわち、Newton 法が発散したり、別の解曲線に移ってしまうといった追跡の失敗を避けることができない。理論上は、1 ステップにおける進み幅を十分小さくすれば追跡の失敗を避けることができるが、そのような進み幅を求めることは従来不可能であるとされていた。

本節では、構成的陰関数定理を用いて、従来不可能であるとされていた微分を用いた解曲線追跡を必ず成功させる手法を導く。この方法によって、予測子修正子法を、区分的線形近似による方法に劣らない確実なアルゴリズムとすることができたと考える。更に、実際の解曲線追跡の際の事情を考慮して解曲線追跡に関するより詳細な議論を行なう。

前節までは一般の Banach 空間を扱っていたが、本節以降は有限次元の通常の Euclid 空間を扱う。

新しい解曲線追跡法の概要は以下の通りである。

方程式 $h(x) = 0$, $h: \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$ の解曲線を追跡するとき、定理 7.5 における $X \times Y, Z, g$ を、各々 $\mathbf{R}^{n+1}, \mathbf{R}^n$, ホモトピー h に対応させる。そして、予測子修正子法の各ステップごとに、 X, Y をそれぞれ解曲線の接線方向、接線の直交補空間を表わす部分空間として再定義し、定理 7.5 を用いて予測子の長さを決定する。

このアルゴリズムをまとめると、次のようになる。

アルゴリズム 7.1 (解曲線追跡) $P = \mathbf{R}^{n+1}$, $Z = \mathbf{R}^n$, $h: P \rightarrow Z$, Dh が α -Lipschitz 連続とし、方程式 $h(x) = 0$ の解曲線を追跡することを考える。 $h(x) = 0$ の解曲線上の点の近似を x_0 , 追跡する方向を表わす単位ベクトルを q_{-1} とする。

(Step 1) $i = 0$ とする。

(Step 2) $h(x_i)$, $r = \|h(x_i)\|$ を計算する。

(Step 3) $Dh(x_i) : P \rightarrow Z$ の近似 $A_i : P \rightarrow Z$ を計算する. 但し, $\|Dh(x_i) - A_i\|$ の大きさを見積ることとし, その上限を K とする.

(Step 4)

$$A_i x = 0, \quad \|x\| = 1, \quad q_{i-1} \cdot x > 0 \quad (7.80)$$

を解き, 解を q_i に代入.

(Step 5) P の部分空間 X, Y を $X = [q_i]$ (q_i によって生成される部分空間), $Y = (X)^T$ (X の直交補空間) と定める.

(Step 6) 式 (7.43), (7.44) を満たすような ε, δ を求める. ただし $d_1 = 0, d_2 = \|A_{Y_i}^{-1}\|$ とする.

(Step 7) 反復

$$y_0 = x_i + \varepsilon q_i, \quad (7.81)$$

$$y_{k+1} = y_k - A_{Y_i}^{-1} h(y_k) \quad (7.82)$$

を繰り返し, 精度を高める.

(Step 8) 十分精度を高めたら, $x_{i+1} = y_k, i = i + 1$ として (Step 2) へ.

□

図 7.3 にこのアルゴリズムの様子を示す. なお, q_{-1} は初期点から追跡したい方向を表わすベクトルで, 例えば t が正の方向に追跡したい場合は, $(x, t) = (0, 1)$ のように定める.

アルゴリズム 7.1 を実際の数値計算に適用する際には, 次のような点が問題となる.

(1) $A_{Y_i}^{-1}$ 及び $\|A_{Y_i}^{-1}\|$ の計算方法.

(2) 式 (7.43), (7.44) を満たす ε, δ の算出. 解曲線の追跡という立場からは, そのような ε, δ の内 ε (予測子の大きさ) を最大にするような ε, δ の組を求める必要がある.

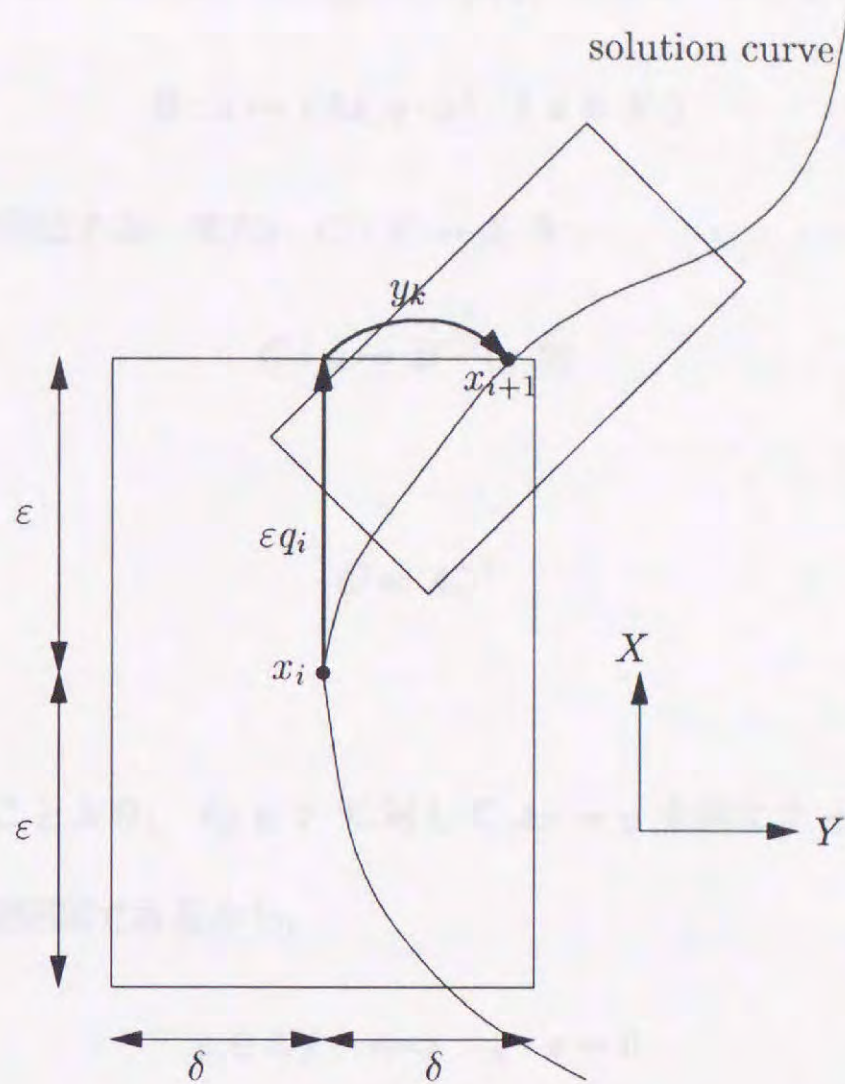


図 7.3: 解曲線追跡のアルゴリズム

(3) Newton 法の反復の停止条件. 十分反復して解の精度を上げないと, 次の ϵ , δ の算出が出来ない (構成的陰関数定理が成立しない).

(4) $t = 1$ 超平面での停止アルゴリズム.

以上の問題点の内, 本節では (1),(2) の解決法を示す.

アルゴリズム 7.1(Step 5) のように部分空間 X , Y を定めたとき, A_i がフルランクであれば A_Y^{-1} は存在する. このとき, A_Y^{-1} 及び $\|A_Y^{-1}\|$ は以下のようにして計算できる.

補題 7.4 (A_Y^{-1} の計算) X を Hilbert 空間, Y を Banach 空間とし, X_1 を $q \in X (q \neq 0)$ によって生成される 1 次元部分空間, X_2 を X_1 の直交補空間, $A: X \rightarrow Y$ を有界線形作用素と

し, A_Y^{-1} が存在するとする. ただし, $A_Y(y) = A(0, y)$ である. このとき, $B: X \rightarrow Y \times R$ を

$$B: x \mapsto (Ax, q \cdot x) \quad (x \in X) \quad (7.83)$$

と定義すると, B^{-1} が存在する. また, $C: Y \rightarrow X$ を

$$C: y \mapsto B^{-1}(y, 0) \quad (7.84)$$

と定義すると,

$$C = A_Y^{-1} \quad (7.85)$$

が成立する. □

(証明) A_Y^{-1} が存在することより, $\forall y \in Y$ に対して $Ax = y$ を満たす $x \in X_2$ は一意に定まる.

X_2 は $X_1 = [q]$ の直交補空間であるから,

$$x \in X_2 \iff q \cdot x = 0 \quad (7.86)$$

である. よって, $Bx = (Ax, q \cdot x) = (y, 0)$ を満たす $x \in X$ は一意に定まり, B^{-1} は存在することが分かる. 更に, $Cy = B^{-1}(y, 0)$ は $Ax = y$, $q \cdot x = 0$ を満たすから, 式 (7.85) の成立が分かる. □

注意 7.2 補題 7.4において, A, q に対して式 (7.84) で定められた C を $C(A; q)$ と書くことにする. $q \in N(A)$ (A の零空間), $q \neq 0$ のとき, $C(A; q)$ は q の選び方によらず同じ作用素となる. これは, A の Moore-Penrose 逆行列 [43] と呼ばれるものと一致する. □

次に, ε, δ の存在条件と, ε が最大となる ε, δ の組を与える計算式として, 次の補題を示す.

補題 7.5 (構成的陰関数定理の成立条件) $\alpha > 0, r \geq 0, K \geq 0, d_1 \geq 0, d_2 > 0$ とし,

$D_1 \equiv d_1 + K, D_2 \equiv \frac{1}{d_2} - K$ と定義する. このとき,

$$d_2 \left(\frac{\frac{1}{2}\alpha\varepsilon^2 + (K + d_1)\varepsilon + r}{\delta} + \alpha(\varepsilon + \delta) + K \right) \leq 1 \quad (7.87)$$

$$d_2(\alpha(\varepsilon + \delta) + K) < 1 \quad (7.88)$$

を満たす $\varepsilon > 0$, $\delta > 0$ が存在するための必要十分条件は,

$$D_2 > 2\sqrt{\alpha r} \quad (7.89)$$

が成立することである. また, そのような (ε, δ) の組のうち ε が最大となるものは,

$$\varepsilon = \frac{1}{\alpha} \left(-2D_1 - D_2 + \sqrt{(2D_1 + D_2)^2 + D_2^2 - 4\alpha r} \right) \quad (7.90)$$

$$\delta = \frac{1}{2\alpha} \left(2D_1 + 2D_2 - \sqrt{(2D_1 + D_2)^2 + D_2^2 - 4\alpha r} \right) \quad (7.91)$$

$$(7.92)$$

で与えられる. □

(証明)

$\alpha > 0$, $\varepsilon > 0$ より式 (7.87) が成立するならば式 (7.88) も自動的に成立.

式 (7.87) を変形し δ について整理すると,

$$\alpha\delta^2 + (\alpha\delta - D_2)\delta + \left(\frac{1}{2}\alpha\varepsilon^2 + D_1\varepsilon + r\right) \leq 0 \quad (7.93)$$

となる. $\varepsilon > 0$ とすると (第3項) > 0 となることから, 式 (7.93) を満たす $\delta > 0$ が存在するための必要十分条件は,

$$(\alpha\varepsilon - D_2)^2 - 4\alpha\left(\frac{1}{2}\alpha\varepsilon^2 + D_1\varepsilon + r\right) \geq 0 \quad (7.94)$$

$$-\frac{\alpha\varepsilon - D_2}{2\alpha} > 0 \quad (7.95)$$

となる. 整理すると,

$$\alpha^2\varepsilon^2 + 2\alpha(2D_1 + D_2)\varepsilon + 4\alpha r - D_2^2 \leq 0 \quad (7.96)$$

$$\alpha\varepsilon - D_2 < 0 \quad (7.97)$$

となる. これを満たす $\varepsilon > 0$ が存在するための必要十分条件は, $2\alpha(2D_1 + D_2) > 0$ より

$$4\alpha r - D_2^2 < 0 \quad (7.98)$$

$$D_2 > 0 \quad (7.99)$$

となる. これは式 (7.89) と同値.

式 (7.89) が満たされているとき, 式 (7.96) を満たす最大の ε は, 式 (7.90) のようになる. ここで, この ε が式 (7.97) を満たす, すなわち

$$\alpha\varepsilon - D_2 = -2D_1 - 2D_2 + \sqrt{(2D_1 + D_2)^2 + D_2^2 - 4\alpha r} \quad (7.100)$$

が負である事を示す. これは, 式 (7.89) より,

$$\begin{aligned} (2D_1 + 2D_2)^2 - ((2D_1 + D_2)^2 + D_2^2 - 4\alpha r) &= 4D_1D_2 + 2D_2^2 - 4\alpha r \\ &> 4D_1D_2 + D_2^2 \\ &= D_2(4D_1 + D_2) \\ &> 0 \end{aligned}$$

により分かる. よって, 式 (7.90) で与えられた ε に対して式 (7.93) を満たす δ が存在する. このときの δ は, 式 (7.96) の等号が成立することから一意に定まり, 式 (7.91) のようになる. \square

この補題 7.5 より, 解曲線上の点の近似 x_i が与えられたとき, そこから解曲線の追跡が出来るためには, $\|Dh(x_i) - A_i\| \leq K$ であるような $Dh(x_i)$ の近似 A_i を計算し, 部分空間 X, Y を定めたとき,

$$\frac{1}{\|A_{Y_i}^{-1}\|} - K > 2\sqrt{\alpha\|h(x_0)\|} \quad (7.101)$$

が成立すればよい.

7.6 数値例

本節では, 前節で構築したアルゴリズムの有効性をシミュレーションによって示す. 従来の予測子修正子法 (解曲線の接線方向に予測子を取り, その点を通り接線に直交する超平面内での Newton 法によって修正する) との比較を行う.

7.6.1 数値例 1

まず,

$$h(x, t) = \frac{256}{15} \left(x - \frac{1}{4}\right) \left(x - \frac{1}{2}\right) \left(x - \frac{3}{4}\right) + \frac{1}{2} - t \quad (7.102)$$

というホモトピーについて, $t = 0$ 超平面から解曲線追跡を行った例を図 7.4 に示す. $\alpha = 52$ と

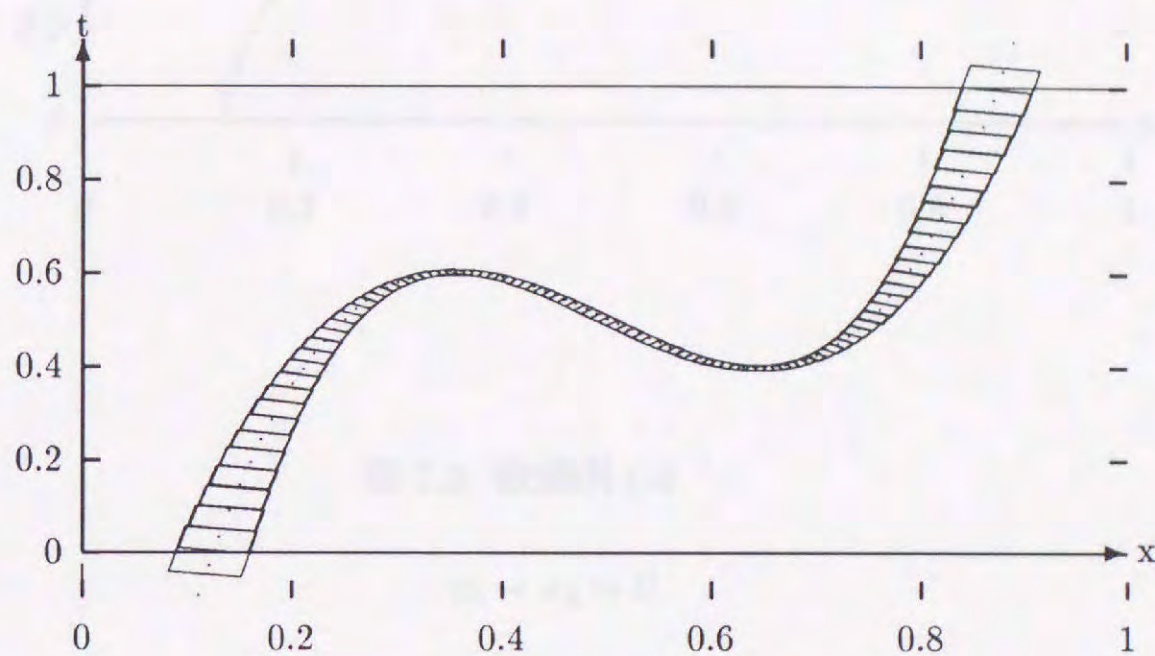


図 7.4: 数値例 1-1

し, 全部で 95 step を要した. 点線は真の解曲線である. Turnig Point の付近では自然に刻みが細かくなり, step 幅の自動調整の効果がある事が分かる.

同じ関数で従来の予測子修正子法を適用した例を, 図 7.5(step 幅 0.1), 図 7.6(step 幅 0.05) に示す. それぞれ, 14 step, 34 step を要した. これを見ると, 図 7.7では step 幅が大きすぎて正しく追跡出来ていない事が分かる.

7.6.2 数値例 2

次に, 2次元の例を示す.

$$x_1^2 + x_2^2 - 1 = 0 \quad (7.103)$$

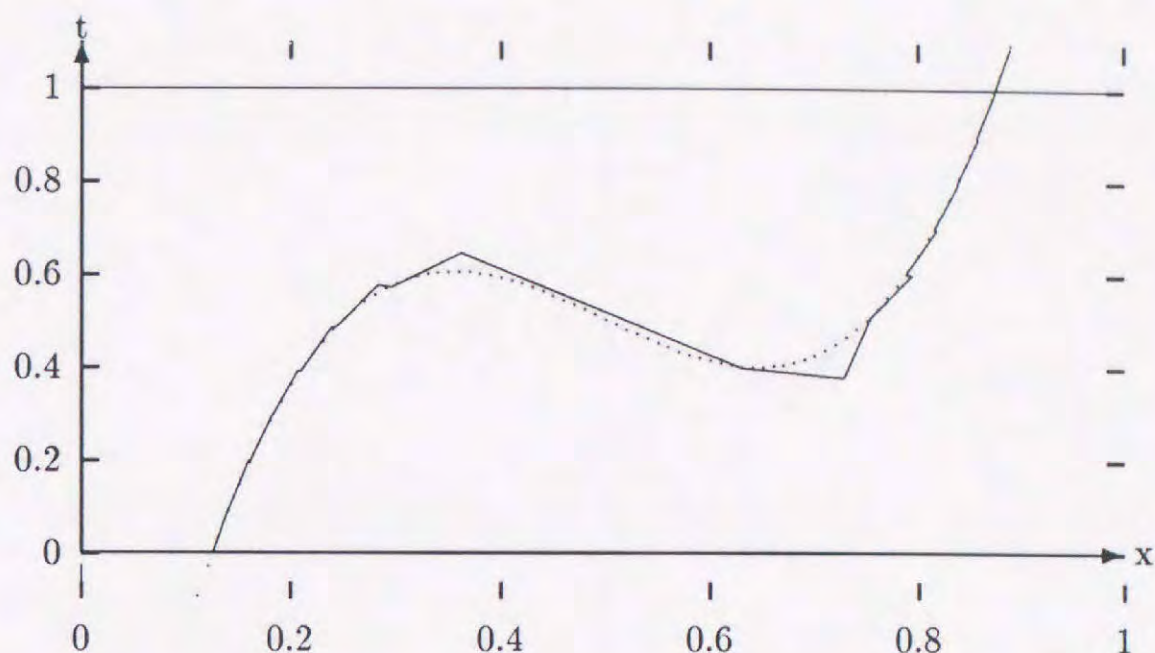


図 7.5: 数値例 1-2

$$x_1 - x_2 = 0 \quad (7.104)$$

に対して, $(1, 0)$ を初期点とする Newton ホモトピーを作ったもので (図 7.7), この場合も正しく追跡出来ている. なお $\alpha = 2.84$ とした.

7.6.3 数値例 3

次に, 人工的に極めて厳しい状況を作って見た場合を示す. ホモトピーを

$$h(x, t) = (0.4(t-1)^2 + 0.4 + c - x) (-0.4t^2 + 0.6 - c - x) \quad (7.105)$$

とした. ただし, $c = 0.01$ とし, $\alpha = 2.5$ とした. この関数の解集合を図 7.8 に示す.

これに対して, 予測子修正子法を適用した. step 幅が 0.05 の場合は, 22 step を要して正常に追跡が行われた (図 7.9). step 幅が 0.15 の場合は, 途中で Newton 法が発散し, 正常に追跡が行われなかった (図 7.10). step 幅が 0.4 の場合は, 途中で違う解曲線に移ってしまい, 正常に追跡が行われなかった (図 7.11). 本論文の方法を用いると, 121 step を要して正常に追跡が行われた (図 7.12).

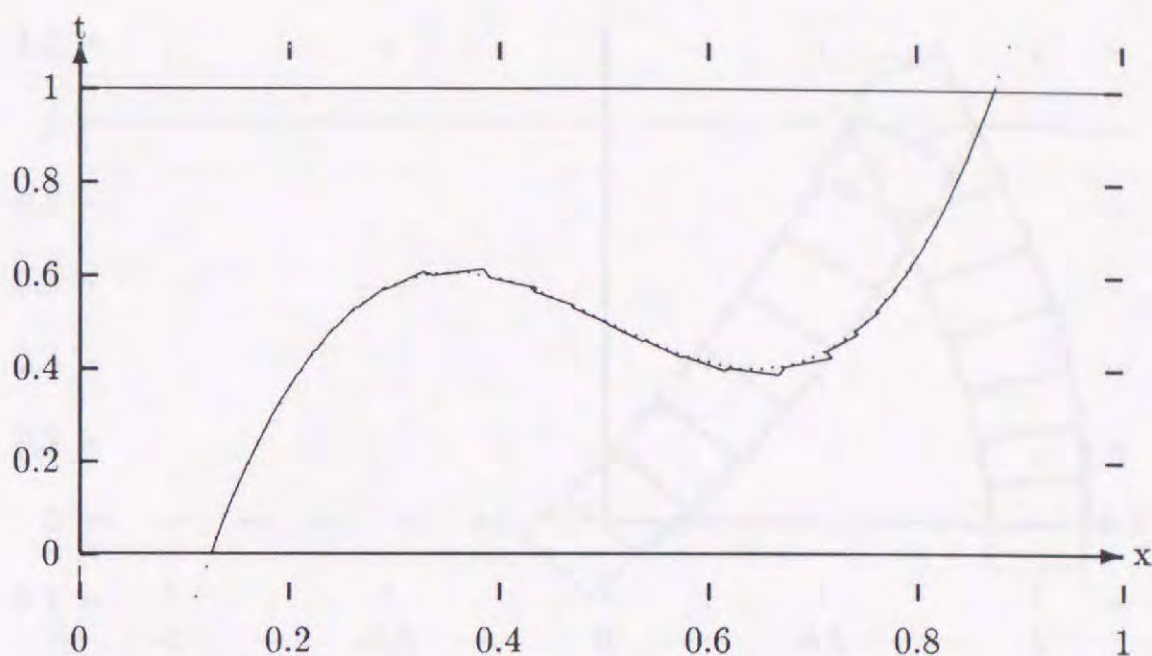


図 7.6: 数値例 1-3

7.7 むすび

本章では、占部の定理を元にして、成立する近傍が見積もれる構成的陰関数定理を導いた。また、導関数の Lipschitz 連続性を仮定した上で、同様な定理を導いた。また、後者の定理を用いて、予測子修正子法を決して追跡を失敗することがないアルゴリズムに改善する方法を導いた。更に、実際の解曲線追跡の際に発生する問題点をいくつか解決した。

本章で導いた解曲線追跡法の今後の課題としては、前節で示した問題点の内の (3), (4) の解決が挙げられる。しかし、更に重要な問題として、現時点では解曲線追跡法は完全な精度保証を行うまでには達していない点、導関数の Lipschitz 定数の見積もりを行う必要がある点は極めて重大であり、その必要がないアルゴリズムの構築を行う必要がある。この 2 つの問題点を同時に解決するには、通常の方程式に対して行われているように、区間解析の導入が効果的であると思われる。しかし、解曲線追跡に应用する場合には、区間演算の性質上直和分解を軸に添った方向にしか行えないので、効率の低下を引き起こすと予想され、今後の課題が残る。

更に、それらの問題点を解決して具体的なアルゴリズムが構築された場合、有限長の解曲線追

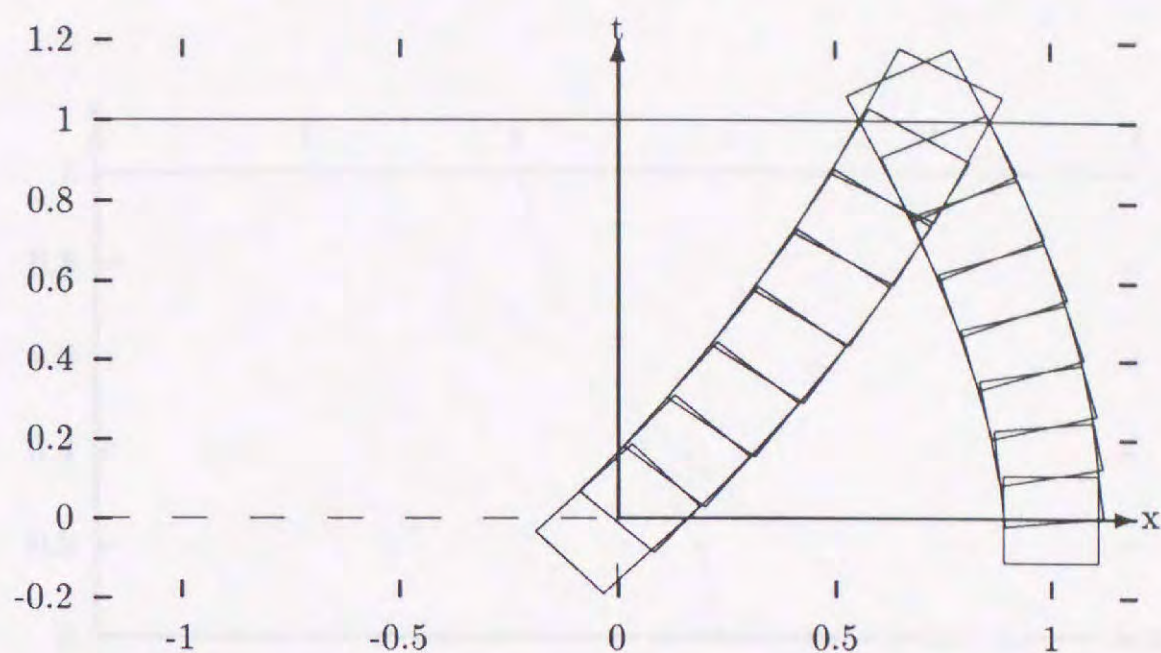


図 7.7: 数値例 2

跡が有限時間内に終了することを保証する必要がある。

また、構成的陰関数定理の別の応用例について検討する予定である。具体的には、次のようなことを考えている。システムの不確定性を余分の変数を用いて定式化し、(変数の数) > (方程式の数) であるような方程式を作る。次元の差は本節の議論と違って 2 以上でも構わない。そして、構成的陰関数定理を使ってその方程式の全ての解を包み込み、システムの振舞いを把握しようということを考えている。これには区間解析が有効に利用出来ると予想される。

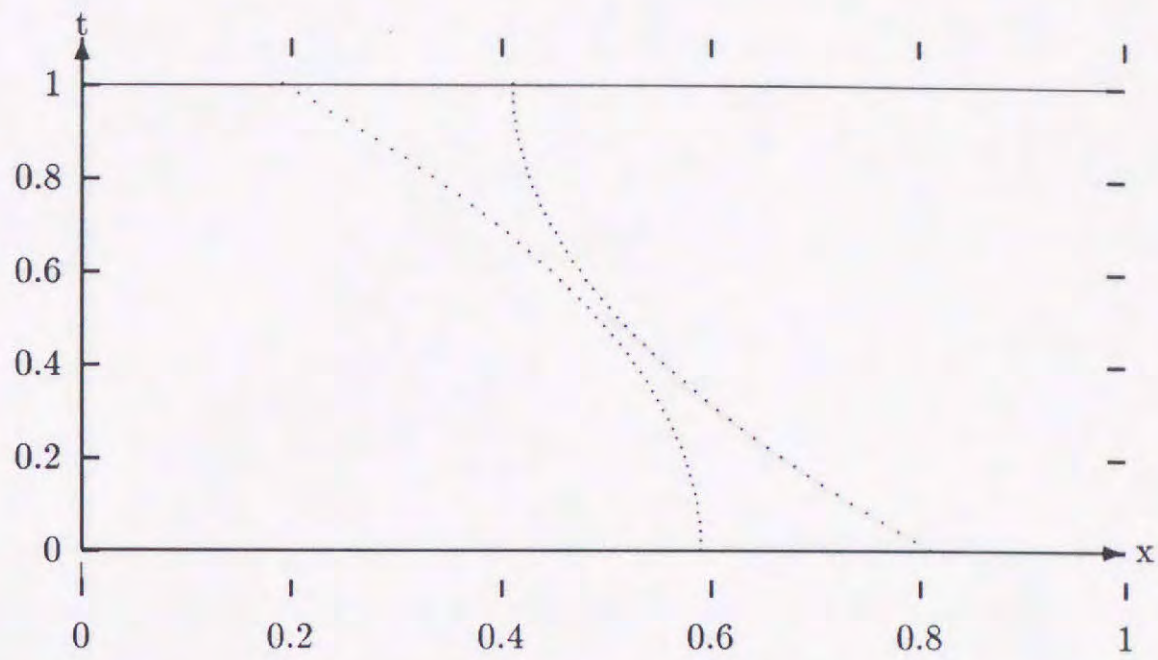


图 7.8: 解集合

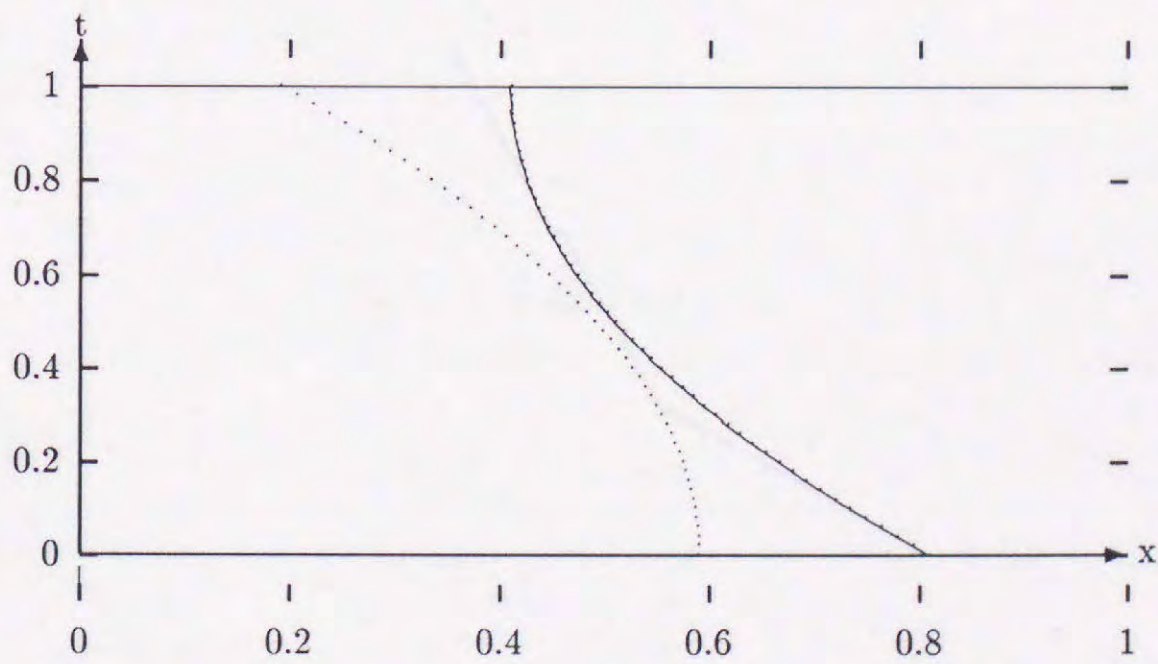


图 7.9: 数值例 3-1

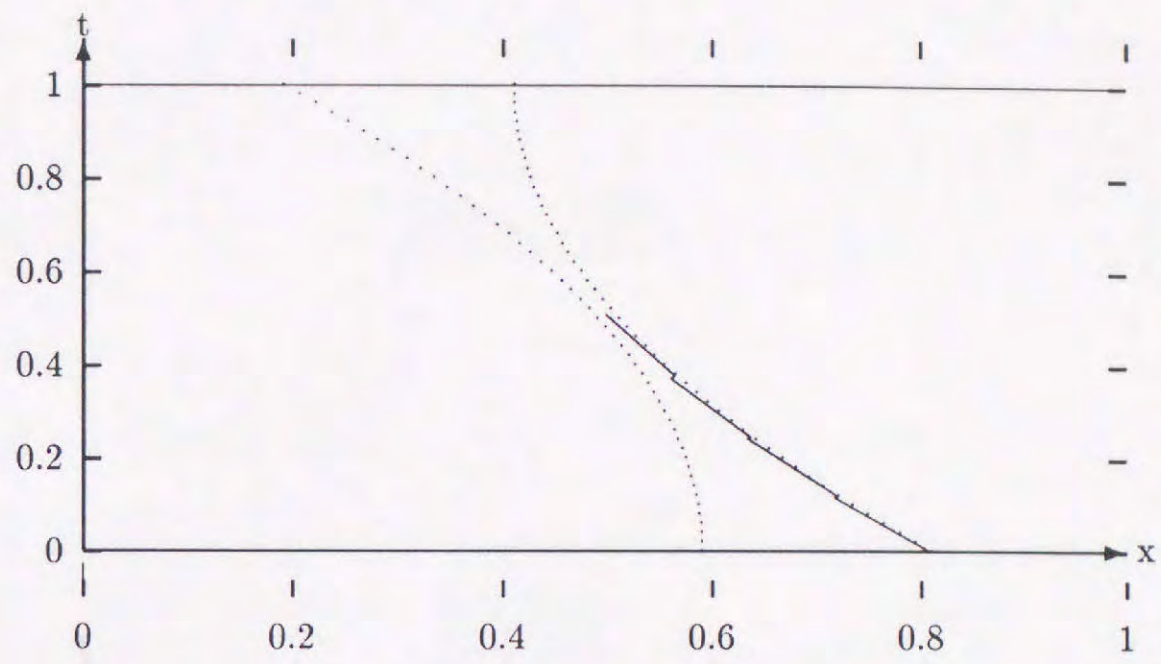


图 7.10: 数值例 3-2

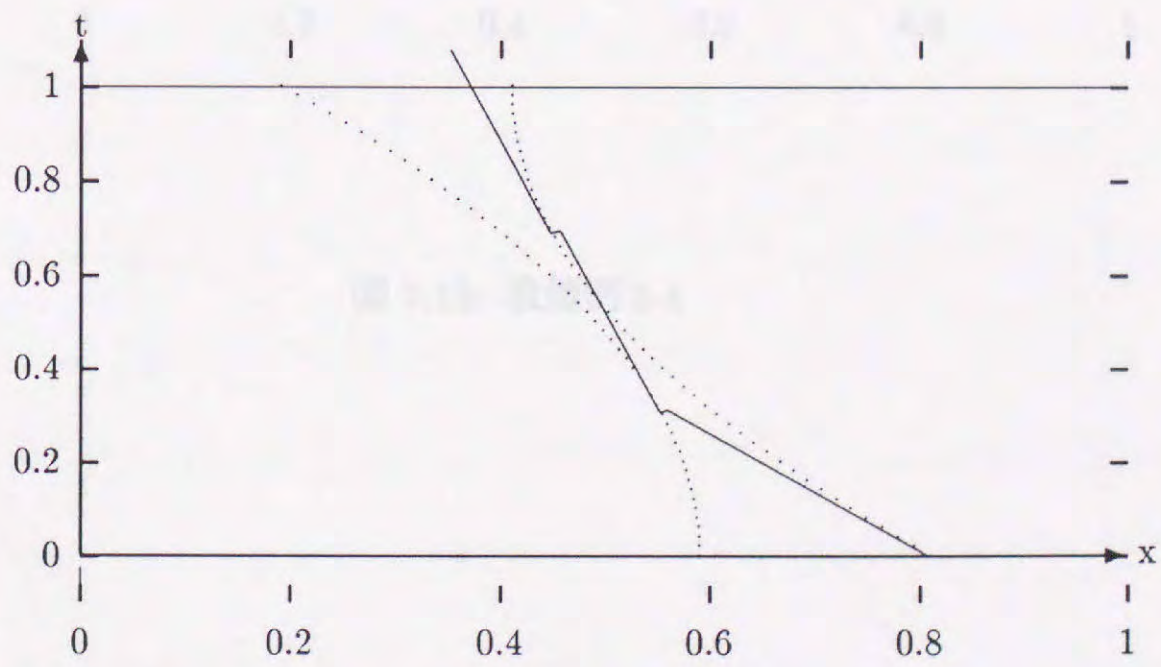


图 7.11: 数值例 3-3

第 8 章

結論

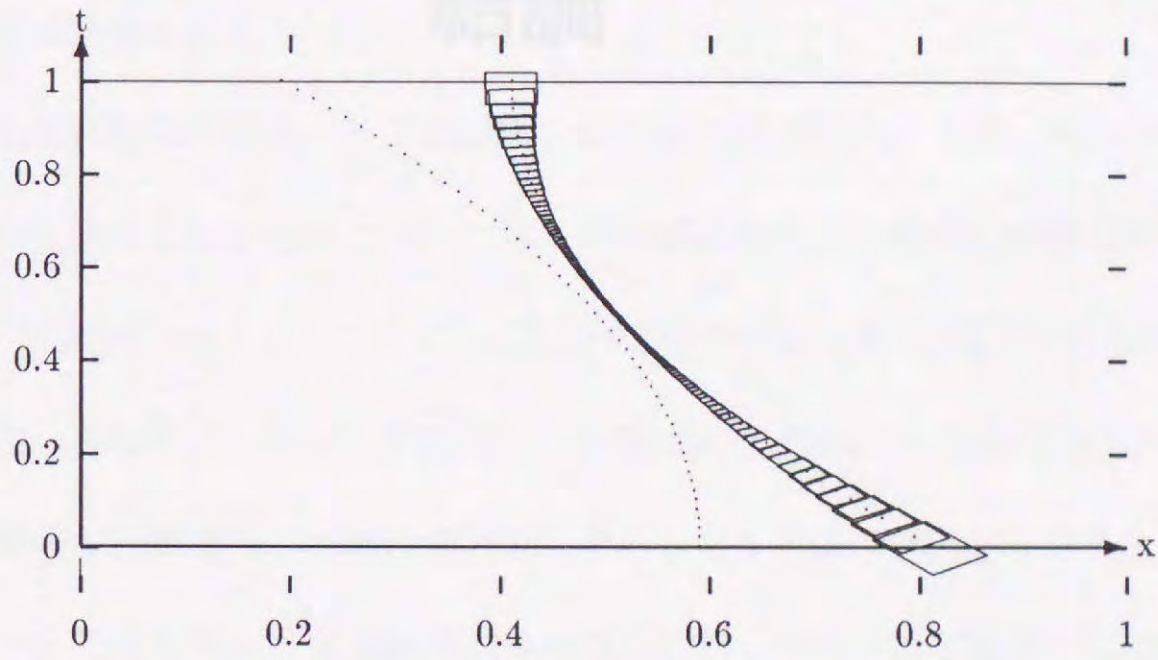


图 7.12: 数值例 3-4

本論文は、序論で述べたような精度保証技法に関する現状を踏まえて、特に非線形方程式の解の精度保証に関する研究成果をまとめたものである。

本論文の前半(第3-6章)では、有限次元の非線形方程式に対して、近似解を元にしたほぼ自動的な精度保証システムの構築に関する研究成果について述べた。本論文の後半では、方程式に幾つかのパラメータが付加された場合など、定義域の次元が値域の次元よりも高いような方程式の解の包み込みについて論じた。

以下、各章の成果をまとめる。

第1章では、本研究が行われた背景として精度保証付き数値計算に関する現状を記述し、本論文の目的、概要、構成を述べた。

第2章では、本論文の議論の基礎となる区間解析について説明した。まず、区間解析で基本となる区間演算をその用語とともに定義した。次に、区間演算を用いた非線形方程式の解の精度保証技法の基本的かつ代表的なものとして、Krawczykの区間写像を用いた方法を説明した。

第3章では、厳密な精度保証のために不可欠な、方程式の計算機上での表現方法について議論した。実数値連続関数を計算機上に厳密に情報落ちすることなく載せることを目的とし、そのために、従来の浮動小数点数に代わって有理数演算を採用した。更に、計算誤差の影響を把握出来るように区間演算を採用した。また、こうして定義された区間写像から元の関数についての情報を完全に取り出せる(元の関数の定義となっている)ことを保証するための出来る限り緩くかつ確認しやすい条件として、連続な区間包囲の概念を提案した。

第4章では、Krawczykの方法を元として、適当な方法で得られた近似解を用いる精度保証アルゴリズムを提案した。Krawczykの方法に対して、与えられた近似解を用いてその近くにあるであろう真の解を包みこむような適当な領域を決定する方法を示した。そしてこの決定法を元にし、また第3章で提案したような関数表現を仮定して、与えられた近似解を用いた真解の包み込みアルゴリズムを示した。また、このアルゴリズムが十分良い近似解に対して有効である事を示した。

第5章では、任意精度までの反復改良を行うアルゴリズムを提案した。第3章で提案した関数

表現を仮定し、また有理数演算を用いた区間反復法を提案した。この方法では、適切に制御された有理数の丸めによって、有理数の桁数の爆発の防止と任意精度までの反復改良を同時に達成した。また、それが正しく動く事を証明した。

第6章では、精度保証システムの実際の実現について議論した。まず、本論文(第3-5章)で提案した方法を実現するために必要なプログラミング技法について議論した。次に、本論文で提案したアルゴリズムを実際に試作した方法について説明した。自動微分の技法を用い、方程式と近似解の2つの入力だけで自動的に精度保証を行うシステムの構築が出来た。最後に、幾つかの例題に対する精度保証を行いその有用性を示した。

第7章では、方程式に幾つかのパラメータが付加された場合など、定義域の次元が値域の次元よりも高いような方程式の解の包み込みについて論じた。

まず、これ以降の議論の基礎となる、簡易 Newton 法の収束定理である占部の定理について議論した。次に、陰関数定理を拡張し、その成立範囲を見積もれるような構成的陰関数定理を占部の定理を元に導いた。更に、導関数の Lipschitz 連続性を仮定し、同様な定理を導いた。

また、構成的陰関数定理の応用例の一つとして、ホモトピー法の解曲線追跡の新しい方法を提案した。まず、ホモトピー法について簡単に説明した。次に、解曲線追跡法の一つである予測子修正子法の改良として、決して失敗しないような新しい解曲線追跡法を構成的陰関数定理から導いた。また、その解曲線追跡法について実際の適用に関する詳細な議論を展開した。最後に、シミュレーションによってその有効性を示した。

以下、精度保証に関する今後の研究方針をいくつか挙げる。

まず、本論文の内容の続きとして、第3章の内容についてより深く考えて行きたい。この章で与えたのは計算機における実現可能な関数の表現方法と、それが満たすべき条件である。これは、構成的数学における関数の定義と密接な関連があると思われるので、それについて探って行きたい。構成的数学では、必ずしも誤差評価という目的で関数を定義するわけではなく、また実際に計算機に載せて計算に用いるという発想ではないので、直接本章の内容と関係するわけではない。

しかし、例えば文献 [50] では、本章で述べたのとよく似た方法で関数を定義しているので、この辺りについて調査し、できれば第 3 章の定義 (区間包囲の連続性) をより美しいものへと発展的に置き換えたい。

大きな目標としては、常微分方程式の解の精度保証アルゴリズムの構築を行う予定である。無限次元方程式に対する精度保証の試みは、序論に挙げたように様々な方法が既に報告されている。これらの大半を占めるのは無限次元において不動点定理の成立を様々なノルムの評価を区間演算等の精度保証技法を用いて行うものであるが、関数空間の設定やノルムの評価法など事前に与えられた問題の性質を詳細に解析し幾つかの式を導く必要があり、関数解析の高度な知識が要求される。このような人手による事前解析は、精度保証の信頼性を損なう可能性を増大させるため、あまり好ましくない。ここで、高次の Taylor 展開と区間演算を用いて常微分方程式の解を包みこむ方法 [28, 29] が知られているが、これは前述の方法とは全く発想が異なり、精度保証の過程を自動化できる可能性が高いと思われる。この方法を元にして、本論文で有限次元非線形方程式に対して構築したものと同様の、近似解と方程式 (正規型の常微分方程式の右辺) の記述だけで、自動的に解の厳密な誤差評価と反復改良の手段を与えるような方法を構築する。詳しい方法は現時点ではまだ書けないが、これができれば多くの点で画期的であると考えられる。

区間演算と自動微分の組み合わせに関して新しいアイデアがある。自動微分と区間演算を組み合わせで作った Jacobi 行列を用いて、mean value form (定義 2.5) を使うと効率の良い区間写像が得られる。ところが、区間演算と自動微分をより密接に融合させるとこれよりはるかに効率の良い区間写像が得られそうだというアイデアがあり、それについて研究する予定である。また、伊理らは Top Down 型の自動微分と区間演算を組み合わせた効率の良い誤差評価アルゴリズムを提案している [30] が、それとの関連についても研究する予定である。これがうまくいけば、本論文の精度保証アルゴリズムは大幅に効率化できる。

関数の変数分離性を活用したアルゴリズムについても研究する予定である。変数分離性とは、関

数 $f(x_1, x_2, \dots, x_n)$ が,

$$f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) \quad (8.1)$$

と書けることである [51, 52]. 変数分離形でない方程式を補助変数を導入して変数分離形に書き換えるアルゴリズムについて新しいアイデアがある. また, 変数分離形である場合, 特に自動微分が大幅に効率化できるため, 精度保証アルゴリズムに取り入れることも検討する.

本論文のアルゴリズムの大きな欠点として, 重解の包み込みが行えないことが挙げられる. 第4章において近似解が真解に十分近いときに精度保証が行えることを証明したが, ここでは真解が単根であることが仮定されていた. この解決は非常に難しい. 与えられた方程式をより高次の空間に埋めこんで特異性を除去するという手法が考えられ, 例えば [53] などの研究があるが, これでも十分とはいえず, 今後研究を深めていきたい. 方針としては,

- (1) 重解の存在が何等かの方法で分かっていた場合, その位置を特定する.
- (2) 重解の存在そのものを数値的に証明し, 更に位置を特定する.

の2つが考えられるが, 後者は不可能に近いと考えている.

最後に, 精度保証を行うためのソフトウェアに関する今後の方針に言及したい. 現状では, 第6章で述べたように, CALC という名前のフリーソフトウェアを用いて試作したに過ぎない. 今後は, C++等の object 指向言語を用いて, 高速な有理数演算を実装し, universal な型を用いた区間演算, 自動微分を実装することによって実用的な精度保証システムを作成し, その効率等について詳細に検討したい. また, 通常の変動小数点数を用いた区間演算や多倍長の浮動小数点数なども実装し, 真に精度保証に有効な技法として定着する方法は何かを探していきたい.

謝辞

本研究は、著者の早稲田大学大学院博士後期課程在学中に行われたものである。本研究を始めるきっかけを与えて下さり、本研究を進めるにあたって終始懇切な御議論、御指導、御検討を賜り、著者の研究生活に常に暖かい御激励を頂きました指導教授大石進一教授に深く感謝の意を表します。著者の研究生活の最初にこのような素晴らしい師と研究テーマにめぐり合えたことは幸運でした。

また、著者の研究生活を通じて、終始懇切な御指導、御鞭撻を賜りました堀内和夫教授に深く感謝致します。

また、本論文をまとめるに際し適切な御示唆を頂きました情報学科廣田良吾教授、電子通信学科大附辰夫教授、電気工学科松本隆教授に深く感謝致します。

また、先輩として終始懇切な御指導、適切な御助言、多くの御助力を頂き、著者の研究生活を支えて下さった中央大学牧野光則専任講師に深く感謝致します。

また、機会ある毎に貴重な御助言を賜りました中央大学伊理正夫教授、九州大学中尾充宏教授、群馬大学山村清隆助教授、愛媛大学土屋卓也助教授、九州大学山本野人助手に感謝致します。

また、本研究を進めるにあたり、活発な討論により数多くの示唆を頂いた井上晃氏(現ソニー(株))、松田俊哉氏(現NTT(株))、山本秀彦氏(現サントリー(株))、小林裕次郎氏(現富士ゼロックス(株))、木村孝氏(現(株)三菱総合研究所)、小林龍氏(現早稲田大学修士2年)、中村晴幸氏(現早稲田大学修士2年)に感謝致します。

また、良き同僚として常に多くの御助言並びに御助力を頂きました大石研究室並びに堀内研究

室の先輩，同輩，後輩諸氏に感謝致します。

最後に，現在までの著者を支えて下さった先生方，友人達，及び家族に深く感謝致します。

参考文献

- [1] 山本直樹「スウェーデンの文学」, *Overlook*, No. 12, pp. 1-4 (1951).
- [2] L. B. Salt, "A Gentleman of Humiliated Consequence", *京都大学文学部研究紀要* 27, pp. 1-11 (1974).
- [3] 平島定家「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 16-21 (1962).
- [4] 中野文彦「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1177-1180 (1962).
- [5] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1181-1184 (1962).
- [6] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1185-1188 (1962).
- [7] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1189-1192 (1962).
- [8] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1193-1196 (1962).
- [9] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1197-1200 (1962).
- [10] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1201-1204 (1962).
- [11] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1205-1208 (1962).
- [12] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1209-1212 (1962).
- [13] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1213-1216 (1962).
- [14] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1217-1220 (1962).
- [15] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1221-1224 (1962).
- [16] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1225-1228 (1962).
- [17] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1229-1232 (1962).
- [18] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1233-1236 (1962).
- [19] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1237-1240 (1962).
- [20] 山本直樹「小説の形式の中心に於ける新説論議」, *文学*, No. 16, pp. 1241-1244 (1962).

参考文献

- [1] 山本哲朗：“エッセイ マジソン便り”，Computrol, No. 12, pp.5-8 (1985).
- [2] L. B. Rall：“Validation of Numerical Computation”，京都大学数理解析研究所講究録 673, pp.1-16 (1988).
- [3] 中尾充宏：“関数方程式の解の存在に対する数値的検証法”，数学, 42, pp.16-31 (1990).
- [4] 中尾充宏：“精度保証付き数値計算の現状と動向”，情報処理, Vol. 31, No. 9, pp.1177-1190 (1990.9).
- [5] 山本哲朗, 陳小君：“有限次元非線形方程式に対する保証付き数値計算”，情報処理, Vol. 31, No. 9, pp.1191-1196 (1990.9).
- [6] 山口昌哉, 吉原英昭, 西田孝明：“微分方程式に対する保証付き数値計算”，情報処理, Vol. 31, No. 9, pp.1197-1203 (1990.9).
- [7] 野田松太郎, 佐々木建昭, 鈴木正幸：“数式処理と数値計算の融合による精度保証”，情報処理, Vol. 31, No. 9, pp.1204-1211 (1990.9).
- [8] 久保田光一, 伊理正夫：“区間演算を用いた丸め誤差解析”，情報処理, Vol. 31, No. 9, pp.1212-1219 (1990.9).
- [9] 奥田晃, 棚町芳弘：“大型計算機向き汎用精度保証付き数値計算ソフトウェア”，情報処理, Vol. 31, No. 9, pp.1220-1227 (1990.9).

- [10] 野寺隆 : “パソコンの精度保証付き数値計算ソフトウェア”, 情報処理, Vol. 31, No. 9, pp.1228-1234 (1990.9).
- [11] S. Oishi : “The Self-Validating Numerical Method—A New Tool for Computer Assisted Proofs of Nonlinear Problems—”, IEICE Trans. Fundamentals, Vol. E75-A, No. 5, pp.595-612 (1992.5).
- [12] P. S. Dwyer : “Error of Matrix Computations”, Mathematical Methods for Digital Computation I, II (eds. A. Ralston And H. S. Wilf), John Wiley (1960, 1967).
- [13] R. E. Moore : “Interval Analysis”, Prentice-Hall, Englewood Cliffs, New Jersey (1966).
- [14] R. E. Moore : “Methods and Applications of Interval Analysis”, SIAM, Philadelphia (1979).
- [15] R. E. Moore (eds.) : “Reliability in Computing”, Academic Press, San Diego (1988).
- [16] G. Alefeld and J. Helzberger : “Introduction to Interval Computations”, Academic Press, New York (1983).
- [17] A. Neumaier : “Interval methods for systems of equations”, Cambridge University Press (1990).
- [18] 山本哲朗 : “Newton 法とその周辺”, 数学, 37, pp.1-15 (1985).
- [19] M. Urabe : “Galerkin's Procedure for Nonlinear Periodic Systems”, Arch. Rat. Mech. Anal., 20, pp.120-152 (1965).
- [20] L. Cesari : “Functional analysis and periodic solutions of nonlinear differential equations”, Contributions to Differential Equations, 1, pp.149-187 (1963).
- [21] G. Kedem : “A posteriori error bounds for two-point boundary value problems”, SIAM J. Numer. Anal., 18, pp.431-448 (1981).

- [22] E. W. Kaucher and W. L. Miranker : "Self-Validating numerics for function space problems", Academic Press, New York (1984).
- [23] E. W. Kaucher and W. L. Miranker : "Validating computation in a function space", Reliability in Computing (eds. R. E. Moore), Academic Press, San Diego, pp.403-425 (1988).
- [24] M. Plum : "Existence proofs in combination with error bounds for approximate solutions of weakly nonlinear second-order elliptic boundary value problems", ZAMM, 71, pp.T660-T662 (1991).
- [25] Shin'ichi Oishi : "Numerical Verification of Existence and Inclusion of Solutions for Non-linear Operator Equation" (submitted to Journal of Computational and Applied Mathematics).
- [26] Ja. G. Sinai and E. B. Vul : "Discovery of closed orbits of dynamical systems with the use of computers", J. Statistical Physics, 23, pp.27-47 (1980).
- [27] 山口昌哉, 吉原英昭, 西田孝明 : "Duffing 方程式の周期解", 数理解析研究所講究録, 673, pp.80-95 (1988).
- [28] R. J. Lohner : "Enclosing the Solutions of Ordinary Initial and Boundary Value Problems", In E. Kaucher, U. Kulisch and Ch. Ullrich (eds.) : "Computer Arithmetic, Scientific Computation and Programming Languages", B. G. Teubner, Stuttgart, pp.255-286 (1987)
- [29] 雨宮 治郎 : "常微分方程式の初期値問題の精度保証付き数値解法における技術的諸問題", 数値解析とそのアルゴリズム研究集会報告集, 数理科学講究録 791, pp.126-139 (1992.6) .
- [30] 伊理正夫, 久保田光一 : "高速自動微分法 (I), (II)", 応用数理, Vol. 1, No. 1, No. 2, pp.17-35, pp.53-63 (1991).

- [31] L. B. Rall : "Automatic Differentiation : Techniques and Applications", Lecture Notes in Computer Science No. 120, Springer, New York (1981).
- [32] "IBM High-Accuracy Arithmetic Subroutine Library (ACRITH)", Program Description and User's Guide, SC 33-6164-02, 3rd edition (1986).
- [33] J. H. Bleher, S. M. Rump, U. Kulisch, M. Metzger, Ch., Ullrich, W. Walter : "FORTRAN-SC : A Study of a FORTRAN Extension for Engineering / Scientific Computation with Access to ACRITH", Computing, 39, pp.93-110 (1987).
- [34] U. Kulisch : "PASCAL-SC : A PASCAL Extension for Scientific Computation; Information Manual and Floppy Disks", Wiley-Teubner Series in Computer Science, Stuttgart (1987).
- [35] O. Aberth : "Precise Numerical Analysis" Wm. C. Brown Publishers, Dubuque, Iowa (1988).
- [36] O. Aberth and M. J. Schaefer : "Precise computation using range arithmetic, via C++" ACM Trans. Math. Software, 18, pp.481-491 (1992).
- [37] O. Aberth and M. J. Schaefer : "Precise matrix eigenvalues using range arithmetic" SIAM J. Matrix Analysis and Applications, 14, pp.235-241 (1993).
- [38] R. Krawczyk : "Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken", Computing, 4, pp.187-201 (1969).
- [39] R. E. Moore : "A Test for Existence of Solutions to Nonlinear Systems", SIAM J. Numer. Anal., Vol. 14, pp.611-615 (1977).
- [40] O. Aberth : "Computable Analysis" McGraw-Hill (1980).

- [41] C. B. Garcia and W. I. Zangwill : "Pathways to Solutions, Fixed Points and Equilibria", Prentice-Hall (1981).
- [42] 小島政和 : "相補性と不動点-アルゴリズムによるアプローチ", 産業図書 (1981).
- [43] E. L. Allgower and K. Georg : "Numerical Continuation Methods", Springer-Verlag (1990).
- [44] 牧野光則, 大石進一 : "無限次元非線形システムの構成的解析法-ホモトピー法の無限次元への拡張-", 信学論 (A), J73-A, 3, pp.470-477 (1990.3).
- [45] S. M. Rump : "Solving Nonlinear Systems with Least Significant Bit Accuracy", Computing, 29, pp.183-200 (1982).
- [46] A. Inoue, M. Kashiwagi, S. Oishi and M. Makino : "A Modified Newton Method with Guaranteed Accuracy Based on Rational Arithmetic", IEICE Trans. Fundamentals, Vol.E76-A, No.5, pp.795-807 (1993.5).
- [47] M. Komuro and T. Saito : "Lost Solution in a piecewise-linear system", IEICE Trans., E74, 11, pp.3625-3627 (1991.11).
- [48] A. Sard : "The measure of critical values of differentiable maps", Bull. AMS., Vol. 48, pp.883-890 (1942).
- [49] L. E. Brouwer : "Über Abbildung von Mannigfaltigkeiten", Mathematische Annalen, Vol. 71, pp.97-115 (1912).
- [50] S. G. Simpson : "Which set existence axiom is needed to prove the Cauchy/Peano theorem for ordinary differential equations?", Journal of Symbolic Logic, 49, pp.783-802 (1984).
- [51] 山村清隆, 村山泰子 : "多変数関数を一変数関数の和で表現するアルゴリズム I", 電子情報通信学会技術研究報告, CAS93-51, NLP93-39, pp.67-74 (1993.6).

[52] 山村清隆, 村山泰子 : “多変数関数を一変数関数の和で表現するアルゴリズム II”, 電子情報通信学会技術研究報告, CAS93-52, NLP93-40, pp.75-82 (1993.6).

[53] N. Yamamoto : “Regularization of solutions of nonlinear equation with singular Jacobian matrices”, J. Inf. Proc., 7, pp.16-21 (1984).

博士論文

(1) 山村清隆 : “多変数関数を一変数関数の和で表現するアルゴリズム”, 電子情報通信学会技術研究報告, CAS93-52, NLP93-40, pp.75-82 (1993.6).

論文

(1) Masahiko KASHIWAZAKI, Masahiko OKADA, Masahiko MARINO and Kazuo DOKIUCHI, “An Order Type Extension Technique for a Conjugate Gradient Method in Infinite Dimensional Spaces”, IEEE, ICCE, Vol. 872, No. 11, pp. 1770-1774 (1990. 11).

(2) Masahiko MARINO, Kazuo DOKIUCHI, Masahiko KASHIWAZAKI and Kazuo DOKIUCHI, “Computational Complexity of Conjugate Gradient Solution for a Class of Deeply Sparse Nonlinear Equations by Hierarchy Method”, IEEE, Vol. 872, No. 11, pp. 1840-1847 (1990. 11).

(3) Masahiko MARINO, Masahiko OKADA, Masahiko KASHIWAZAKI and Kazuo DOKIUCHI, “An Order Type A Prediction Solving Criterion and a Global Convergence Property of the Extended Approximate Hierarchy Method”, IEEE Trans., Vol. 774, No. 4, pp. 1040-1044 (1991. 4).

関連業績

修士論文

- (1) 柏木 雅英：“ホモトピー法に対するニュートン法の応用に関する研究”，平成2年度早稲田大学修士論文 (1991.2.14).

論文

- (1) Masahide KASHIWAGI, Shin'ichi OISHI, Mitsunori MAKINO and Kazuo HORIUCHI :
“An Urabe Type Convergence Theorem for a Constructive Simplified Newton Method in Infinite Dimensional Spaces”, Trans. IEICE, Vol.E73, No.11, pp.1789-1791 (1990.11).
- (2) Mitsunori MAKINO, Shin'ichi OISHI, Masahide KASHIWAGI and Kazuo HORIUCHI :
“Computational Complexity of Calculating Solutions for a Certain Class of Uniquely Solvable Nonlinear Equation by Homotopy Method”, Trans. IEICE, Vol.E73, No.12, pp.1940-1947 (1990.12).
- (3) Mitsunori MAKINO, Shin'ichi OISHI, Masahide KASHIWAGI and Kazuo HORIUCHI :
“An Urabe Type A Posteriori Stopping Criterion and a Globally Convergent Property of the Simplicial Approximate Homotopy Method”, IEICE Trans., Vol.E74, No.6, pp.1440-1446 (1991.6).

- (4) 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫 : “非線形強単調抵抗回路方程式のホモトピー法による求解の計算量の上限”, 電子情報通信学会論文誌 (A), Vol.J74-A, No.8, pp.1151-1159 (1991.8).
- (5) Mitsunori MAKINO, Shin'ichi OISHI, Masahide KASHIWAGI and Kazuo HORIUCHI : “Infinite Dimensional Homotopy Method of Calculating Solutions for Fredholm Operator with Index 1 and A-Proper Operator Equations”, IEICE Trans. Fundamentals, Vol.E75-A, No.5, pp.613-615 (1992.5).
- (6) Mitsunori MAKINO, Masahide KASHIWAGI, Shin'ichi OISHI and Kazuo HORIUCHI : “A Sufficient Condition of A Priori Estimation for Computational Complexity of the Homotopy Method”, IEICE Trans. Fundamentals, Vol.E76-A, No.5, pp.786-794 (1993.5).
- (7) Akira INOUE, Masahide KASHIWAGI, Shin'ichi OISHI and Mitsunori MAKINO : “A Modified Newton Method with Guaranteed Accuracy Based on Rational Arithmetic”, IEICE Trans. Fundamentals, Vol.E76-A, No.5, pp.795-807 (1993.5).
- (8) Mitsunori MAKINO, Masahide KASHIWAGI, Shin'ichi OISHI and Kazuo HORIUCHI : “An Estimation Method of Region Guaranteeing Existence of a Solution Path in Newton Type Homotopy Method”, IEICE Trans. Fundamentals, Vol.E76-A, No.7, pp.1113-1116 (1993.7).

国際会議

- (1) Shin'ichi OISHI, Kazuo HORIUCHI, Mitsunori MAKINO and Masahide KASHIWAGI : “On the Information Needed for Solving Nonlinear Equations”, Proc. 1990 International Symposium on Information Theory and Its Applications (ISITA'90), pp.307-310 (Hawai, U.S.A., Nov. 29, 1990).

- (2) Shin'ichi OISHI, Masahide KASHIWAGI, Mitsunori MAKINO and Kazuo HORIUCHI :
“An Urabe Type Convergence Theorem for a Constructive Simplified Newton Method in
Infinite Dimensional Spaces”, Proc. 1991 IEEE International Symposium on Circuits and
Systems (ISCAS'91), pp.1236-1239 (Singapore, 11-14 June 1991).
- (3) Shin'ichi OISHI, Masahide KASHIWAGI, Mitsunori MAKINO and Kazuo HORIUCHI
: “Constructive implicit function theorem and its application”, 13th IMACS WORLD
CONGRESS ON COMPUTATION AND APPLIED MATHEMATICS (Dublin, Ireland,
22-26 July 1991).
- (4) Masahide Kashiwagi and Shin'ichi Oishi : “Krawczyk-Based Numerical Validation Using
Rational Arithmetic”, Proc. 1993 International Symposium on Nonlinear theory and its
Applications (NOLTA '93), pp.399-402 (Hawaii, U.S.A., 5-10 December 1993).
- (5) Mitsunori Makino, Masahide Kashiwagi and Shin'ichi Oishi : “A Priori Estimation for
Computational Complexity of Homotopy Method for Calculating Solutions of Strongly
Monotone Equations”, Proc. 1993 International Symposium on Nonlinear theory and its
Applications (NOLTA '93), pp.1053-1056 (Hawaii, U.S.A., 5-10 December 1993).

研究会資料

- (1) 柏木 雅英, 大石 進一, 牧野 光則, 須見 祐三 : “ホモトピー法による解の計算可能性の必要十
分条件と解の計算可能性の延長定理”, 電子情報通信学会技術研究報告, NLP89-2, pp.9-15
(1989.6.23).
- (2) 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫 : “ホモトピー法と大域的収束性”, 電子情報通
信学会技術研究報告, CAS89-98, NLP89-42, pp.39-44 (1989.11.20).

- (3) 柏木 雅英, 大石 進一, 牧野 光則 : “非線形方程式の大域的解法の収束性の補完について”, 電子情報通信学会技術研究報告, CAS89-125, pp.39-46 (1990.2.7).
- (4) 牧野 光則, 大石 進一, 柏木 雅英 : “唯一解をもつ非線形方程式のホモトピー法による解の計算の複雑度”, 電子情報通信学会第 3 回回路とシステム軽井沢ワークショップ論文集, pp.230-237 (1990.5.25).
- (5) 柏木 雅英, 竹市 正彦, 牧野 光則, 大石 進一 : “無限次元における構成的ニュートン法について”, 電子情報通信学会技術研究報告, NLP90-3, pp.15-22 (1990.6.30).
- (6) 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫 : “占部の定理の大域化について”, 第 1 回非線形理論とその応用学術研究集会論文集, pp.139-146 (1990.7.26).
- (7) 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫 : “非線形方程式を解くために必要な情報量について”, 電子情報通信学会技術研究報告, IT90-88, pp.13-18 (1990.11.17).
- (8) 柏木 雅英, 大石 進一, 牧野 光則, 堀内 和夫 : “構成的陰関数定理とその応用”, 電子情報通信学会技術研究報告, NLP90-36, pp.13-18 (1990.11.22).
- (9) Mitsunori MAKINO, Shin'ichi OISHI, Masahide KASHIWAGI and Kazuo HORIUCHI : “Computational Complexity of Homotopy Method for Monotone Resistive Circuits”, Proc. 1990 Joint Technical Conference on Circuits/Systems, Computers and Communications (JTC-CSCC'90), pp.359-364 (1990.12.10).
- (10) Masahide KASHIWAGI, Shin'ichi OISHI, Mitsunori MAKINO and Kazuo HORIUCHI : “Constructive Implicit Function Theorem and its Application to Circuit Simulation”, Proc. 1990 Joint Technical Conference on Circuits/Systems, Computers and Communications (JTC-CSCC'90), pp.524-529 (1990.12.11).

- (11) 竹市 正彦, 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫: “無限次元非線形システムの分岐の構成的解析法”, 電子情報通信学会技術研究報告, CAS90-124, NLP90-64, pp.21-28 (1991.1.19).
- (12) 井上 晃, 大石 進一, 牧野 光則, 柏木 雅英, 堀内 和夫: “単体近似による非線形方程式の精度保証付き数値解析法”, 電子情報通信学会技術研究報告, NLP90-72, pp.37-44 (1991.3.15).
- (13) 松田 俊哉, 大石 進一, 牧野 光則, 柏木 雅英: “有限回の手順で終了することを保証した全解探索アルゴリズム”, 電子情報通信学会第4回回路とシステム軽井沢ワークショップ論文集, pp.25-30 (1991.5.24).
- (14) Akira INOUE, Shin'ichi OISHI, Mitsunori MAKINO, Masahide KASHIWAGI and Kazuo HORIUCHI: “Self Validating Numerics for Nonlinear Equations Using Simplicial Approximation and Rational Number Arithmetics”, 第2回非線形理論とその応用学術研究集会 (NOLTA'91 シンポジウム) 論文集, pp.25-28 (1991.7.15).
- (15) Shin'ichi OISHI, Hisa-Aki TANAKA, Mitsunori MAKINO, Masahide KASHIWAGI and Kazuo HORIUCHI: “Self-Validating Numerical Integration Method for Initial Value Problems of Nonlinear Ordinary Differential Equations”, 第2回非線形理論とその応用学術研究集会 (NOLTA'91 シンポジウム) 論文集, pp.29-30 (1991.7.15).
- (16) Mitsunori MAKINO, Shin'ichi OISHI, Akira INOUE, Hisa-Aki TANAKA and Masahide KASHIWAGI: “A Self-Validating Numerical Method for Solving Nonlinear Circuits”, Proc. 1991 Joint Technical Conference on Circuits/Systems, Computers and Communications (JTC-CSCC'91), pp.11-16 (1991.7.21).
- (17) 柏木 雅英, 大石 進一: “導関数のリプシッツ行列を用いた区間写像による近似解の誤差評価”, 電子情報通信学会技術研究報告, CAS91-131, NLP91-74, pp.19-24 (1992.1.29).
- (18) 井上 晃, 柏木 雅英, 大石 進一, 牧野 光則: “有理数演算による精度保証付きニュートン法”,

- 電子情報通信学会技術研究報告, CAS91-133, NLP91-76, pp.29-36 (1992.1.29).
- (19) 山本 秀彦, 柏木 雅英, 大石 進一: “パラメータ依存非線形方程式の全解探索アルゴリズム - 構成的陰関数定理の応用 -”, 電子情報通信学会技術研究報告, CAS91-141, NLP91-84, pp.13-20 (1992.1.30).
- (20) 柏木 雅英, 木村 孝, 井上 晃, 大石 進一: “有理数演算を用いた精度保証付き区間反復法”, 電子情報通信学会技術研究報告, NLP91-96, pp.23-29 (1992.3.6).
- (21) 柏木 雅英, 大石 進一: “区間演算による構成的陰関数定理を用いた解曲線追跡アルゴリズム”, 電子情報通信学会第5回回路とシステム軽井沢ワークショップ論文集, pp.315-320 (1992.4.22).
- (22) 大石 進一, 井上 晃, 柏木 雅英, 小林 龍, 牧野 光則: “離散力学系の解軌道の精度保証”, 電子情報通信学会技術研究報告, NLP92-5, pp.29-32 (1992.5.21).
- (23) 井上 晃, 柏木 雅英, 大石 進一, 牧野 光則: “有理数演算による非線形方程式の近似解の精度保証~有理数演算による精度保証付きニュートン法~”, 短期共同研究‘非線形問題の数値解析’報告集, 数理解析研究所講究録 787, pp.1-27 (1992.6).
- (24) 柏木 雅英, 大石 進一: “導関数のリプシッツ行列を用いた区間写像について”, 短期共同研究‘非線形問題の数値解析’報告集, 数理解析研究所講究録 787, pp.72-94 (1992.6).
- (25) Akira INOUE, Masahide KASHIWAGI, Shin'ichi OISHI and Mitsunori MAKINO: “A Modified Newton Method Based on Rational Arithmetic”, Proc. 1992 Symposium on Nonlinear Theory and its Applications (NOLTA'92 Symposium), pp.43-46 (1992.7.14).
- (26) Mitsunori MAKINO, Shin'ichi OISHI, Masahide KASHIWAGI and Kazuo HORIUCHI: “A Sufficient Condition of A Priori Estimation for Computational Complexity of the

- Homotopy Method”, Proc. 1992 Symposium on Nonlinear Theory and its Applications (NOLTA'92 Symposium), pp.147-150 (1992.7.16).
- (27) 田中 久陽, 岡田 淳, 大石 進一, 柏木 雅英, 中村 晴幸: “非線形ダイナミカルネットワークの積分可能性と積分不可能性”, 電子情報通信学会技術研究報告, NC92-77, pp.63-70 (1992.11.24).
- (28) 牧野 光則, 柏木 雅英, 大石 進一: “強単調な非線形方程式に対するホモトピー法における解曲線の存在領域の評価”, 電子情報通信学会技術研究報告, CAS92-96, NLP92-76, pp.17-24 (1993.1.13).
- (29) 小林 龍, 大石 進一, 柏木 雅英: “常微分方程式の周期解数値検証システム”, 電子情報通信学会技術研究報告, NLP92-115, pp.103-108 (1993.3.27).
- (30) 中村 晴幸, 柏木 雅英, 大石 進一, 堀内 和夫: “区間演算による有理数演算を用いた非線形方程式の全解の精度保証”, 電子情報通信学会技術研究報告, NLP92-116, pp.109-114 (1993.3.27).
- (31) 牧野 光則, 柏木 雅英, 大石 進一: “ニュートンホモトピー方程式の解曲線の存在範囲の評価”, 電子情報通信学会技術研究報告, NLP92-118, pp.121-126 (1993.3.27).
- (32) 柏木 雅英, 大石 進一: “有理数演算による精度保証付き区間反復法”, 統計数理研究所共同研究レポート 45, 最適化:モデリングとアルゴリズム 3, pp.386-396 (1993.3).
- (33) 柏木 雅英, 大石 進一: “区間解析と有理数演算による非線形方程式の近似解の精度保証”, 短期共同研究‘精度保証付き数値計算法とその応用’報告集, 数理解析研究所講究録 831, pp.53-72 (1993.4).
- (34) 大石 進一, 柏木 雅英: “Numerical Verification of Existence and Inclusion of Solutions for Nonlinear Operator Equations”, 短期共同研究‘精度保証付き数値計算法とその応用’報告集, 数理解析研究所講究録 831, pp.115-128 (1993.4).

- (35) 牧野 光則, 柏木 雅英, 大石 進一: “ニュートンホモトピー方程式に対する解曲線の存在範囲と計算量に関する検討”, 電子情報通信学会技術研究報告, CAS93-48, NLP93-36, pp.43-49 (1993.6.19).
- (36) 柏木 雅英, 大石 進一: “区間演算と有理数演算を用いた非線形方程式の近似解の精度保証”, 電子情報通信学会技術研究報告, CAS93-53, NLP93-41, pp.83-90 (1993.6.19).

講演

- (1) 柏木 雅英, 大石 進一, 牧野 光則, 須見 祐三: “ホモトピー法による解の計算可能性とその延長定理”, 1989年電子情報通信学会秋季全国大会講演論文集, A-29, p.1-32 (1989.9.12).
- (2) 柏木 雅英, 大石 進一, 牧野 光則: “単体近似ホモトピー法の収束性の補完”, 1990年電子情報通信学会春季全国大会講演論文集, A-2, p.1-2 (1990.3.21).
- (3) 牧野 光則, 大石 進一, 柏木 雅英: “唯一解をもつ非線形方程式のホモトピー法による解の計算の複雑度”, 1990年電子情報通信学会春季全国大会講演論文集, A-3, p.1-3 (1990.3.21).
- (4) 牧野 光則, 大石 進一, 柏木 雅英, 堀内 和夫: “簡易ニュートン法を援用したホモトピー法による非線形方程式の解の計算複雑度”, 1990年電子情報通信学会秋季全国大会論文集, A-8, p.1-8 (1990.10.3).
- (5) 柏木 雅英, 牧野 光則, 大石 進一, 堀内 和夫: “有限次元射影を用いた無限次元における構成的ニュートン法”, 1990年電子情報通信学会秋季全国大会論文集, A-9, p.1-9 (1990.10.3).
- (6) 松田 俊哉, 大石 進一, 牧野 光則, 柏木 雅英: “占部の定理を援用した全解探索アルゴリズム”, 1991年電子情報通信学会秋季大会講演論文集, A-29, p.1-29 (1991.9.7).
- (7) 柏木 雅英, 大石 進一, 牧野 光則: “構成的陰関数定理と予測子修正子法の改良”, 1991年電子情報通信学会秋季大会講演論文集, A-30, p.1-30 (1991.9.7).

- (8) 牧野 光則, 大石 進一, 柏木 雅英: “Fredholm かつ A-プロパーな無限次元非線形方程式の無限次元ホモトピー法による数値解析法”, 1991 年電子情報通信学会秋季大会講演論文集, A-32, p.1-32 (1991.9.7).
- (9) 井上 晃, 大石 進一, 牧野 光則, 柏木 雅英: “有理数演算による非線形方程式の近似解の精度保証”, 1991 年電子情報通信学会秋季大会講演論文集, A-35, p.1-35 (1991.9.7).
- (10) 牧野 光則, 柏木 雅英, 大石 進一, 堀内 和夫: “唯一解を持つ非線形方程式のニュートンホモトピーによる求解の計算量の上界の事前評価”, 1992 年電子情報通信学会春季大会講演論文集, A-85, p.1-85 (1992.3.27).
- (11) 山本 秀彦, 柏木 雅英, 大石 進一: “パラメータ依存非線形方程式の全解探索アルゴリズム - 構成的陰関数定理の応用 -”, 1992 年電子情報通信学会春季大会講演論文集, A-86, p.1-86 (1992.3.27).
- (12) 松田 俊哉, 大石 進一, 牧野 光則, 柏木 雅英: “成分毎の占部の定理による全解探索アルゴリズム”, 1992 年電子情報通信学会春季大会講演論文集, A-87, p.1-87 (1992.3.27).
- (13) 柏木 雅英, 大石 進一: “区間演算を用いた近似解の誤差評価について”, 1992 年電子情報通信学会春季大会講演論文集, A-89, p.1-89 (1992.3.27).
- (14) 木村 孝, 井上 晃, 柏木 雅英, 大石 進一: “有理数を端点に持つ区間演算による非線形方程式の近似解の精度保証”, 1992 年電子情報通信学会春季大会講演論文集, A-91, p.1-91 (1992.3.27).
- (15) 井上 晃, 柏木 雅英, 大石 進一, 牧野 光則: “有理数演算による精度保証付きニュートン法”, 1992 年電子情報通信学会春季大会講演論文集, A-93, p.1-93 (1992.3.27).
- (16) 牧野 光則, 柏木 雅英, 大石 進一, 堀内 和夫: “非線形強単調方程式の数値解析におけるホモトピー法の計算量の上界の事前評価”, 1992 年電子情報通信学会秋季大会講演論文集, A-38, p.1-38 (1992.9.29).

- (17) 柏木 雅英, 大石 進一: “区間演算を用いた近似解の誤差評価について (II)”, 1992 年電子情報通信学会秋季大会講演論文集, A-40, p.1-40 (1992.9.29).
- (18) 大石 進一, 小林 龍, 井上 晃, 柏木 雅英, 牧野 光則: “離散力学系の解軌道の精度保証について”, 1992 年電子情報通信学会秋季大会講演論文集, SA-2-3, pp.1-245-1-246 (1992.9.30).
- (19) 柏木 雅英, 大石 進一: “非線形方程式の近似解に対する精度保証システムの試作”, 1993 年電子情報通信学会春季大会講演論文集, A-61, p.1-61 (1993.3.28).
- (20) 中村 晴幸, 柏木 雅英, 大石 進一, 堀内 和夫: “区間演算による非線形方程式の与えられた区間における全解の精度保証”, 1993 年電子情報通信学会春季大会講演論文集, A-67, p.1-67 (1993.3.28).
- (21) 牧野 光則, 柏木 雅英, 大石 進一, 堀内 和夫: “唯一解をもつ非線形方程式に対するニュートンホモトピーによる解曲線の存在範囲の評価”, 1993 年電子情報通信学会春季大会講演論文集, A-86, p.1-86 (1993.3.29).
- (22) 牧野 光則, 柏木 雅英, 大石 進一: “ニュートンホモトピー方程式の解曲線の存在範囲と計算量の事前評価”, 1993 年電子情報通信学会秋季大会講演論文集, A-44, p.1-44 (1993.9.8).

理工社謹製