

大阪大学 工学部 数値計算 講座

NEC 曾谷勝義

1. 数値解法と応用分野
2. 偏微分方程式
3. 数値解法の歴史的発展、分類と特徴

数値解法と応用分野

1. 数値計算基礎編

本論文の付録として、先ず科学技術計算の分類方法をテーマにして、行列としての数値計算の歴史に始まり、連立一次方程式が直接法と反復法に分けられる点、そして行列の分類、更に収束や誤差の概要を説明し、各数値解法のアルゴリズムと、社会全般の応用事例に至る迄の全体をまとめた。

1-1. 数値計算研究のオリジナル性

数値計算の代表である連立一次方程式の解き方は、色々な解法が提案されている。採り上げた数値解法は、固有値問題解法を含めて 52 種類にのぼる。数値解法のアルゴリズムを単に列挙しただけのものではなく、著者による色々な数値実験により、これらの数値解法の特徴と長短を分析し、更に著者の数多くのベンチマークの経験から、その解法が社会でどの様に使われているかを詳細に調査した結果、活用されている応用領域はどのような分野であるかを述べた。それは長年における実務の経験から得たものであり、他のテキストや、他の論文、数値計算レポート等では紹介されていない。数値解法の幾つもの理論と、そのアルゴリズムの研究は、参考文献に記載している他の文献やテキスト等に多く見つかっているが、これを応用されている分野として調査しまとめたのは、当論文が初めてである。これらは他の研究では存在しない。そこにこの論文の数値計算のオリジナル性がある。

数値計算応用領域の調査は、NEC SX シリーズを受注するにおいて、NEC スーパーコンピュータの担当者は、各国の研究機関や大学、産業界に至る迄、多くの技術計算プログラムのベンチマークを行ってきたことによる。このベンチマークの対象は、かなり古い時代のものに始まり、現在に至る最新のものまで数多くあった。それは計算機上で数値計算が行われる様になった 1950 年代に始まり、数値計算が一部の研究機関で研究開発が行われた 1960 年代、数値計算が活性化されてきた 1970 年代、スーパーコンピュータが社会に普及を始めた 1980 年代、スーパーコンピュータが HPC として更に普及を進めた 1990 年代、超高速なスーパーコンピュータを目指して、飛躍的に発展を続ける 2000 年代に至る現在迄、対応してきた数は、社会で使われている数百本のプログラムに及ぶ。近年においては、数値計算ライブラリが市販化され、各専門分野の研究者は、その多くが著名なライブラリを引用しているが、このライブラリが無かった古い時代は、プログラム作成者が市販の著作を見ながら、基本的な数値解法に対してプログラミングをしていた事が多くあった。NEC SX ベンチマークの時に、プログラムコードを理解して、応用領域に対してそれらを分類しながら集計してまとめた。

この論文のテーマとして、数値解法の説明や応用領域を研究しているが、数値解法を説明する上で、先ずは使用されている分野に注目し、これを重要な位置づけにしたい。それには社会の科学技術計算全体を著者の経験による一定の規則で分類を行い、数値解法の発展を含めて説明したい。

1-2. 科学技術計算の分類方法と数値解析の応用事例

この節では、科学技術計算全体に対する数値解析の応用事例を課題として採り上げる[87]。社会には数多くの工学分野がある。NEC SX シリーズを使っのての数多くの科学技術計算のベンチマークを実施した。この時、多くの工学分野に幅広く数値解法が出現するため、あるルールに従ってこれらの

工学分野を1括りにして分類を試みた。工学分野の中で、数値計算は社会の様々な領域で活用されている[40]。科学技術分野における分類方法と応用分野を研究した結果を表したい。社会における科学技術分野を、次の5つに大別してみたい。まとめかたに少し飛躍的な面もあるが、ここでは独自の観点から科学技術分野を大きく捉えて概念上の分類を試みた。

- (1) 物理量が釣り合うという考えが基礎にある釣り合いの方程式から構成された形。
- (2) 時間と空間を有限個に分割して、場を考える時間と空間の有限差分による方程式から構成された形[68]。
- (3) 時間と空間の差分に加えて、有限の和集合の考えも合わせて構成される空間差分と有限の和集合による方程式として構成された形。
- (4) 確率の考えが基礎となる乱数があり、計画計算が加わった形として、乱数と計画計算の形式を1つにまとめた形。
- (5) 無限級数が存在し、その中で有限和を考えとしてまとめて構成される計算方式、つまり無限級数の有限和の形式として構成された形。

数値計算を必要とする科学技術分野全体を、以上の様な形として大別する事が出来る。

- (1)においては、近年ではその多くが有限要素法とした名のもとで基本構成される事が多く、これを有限要素法系として1つに括る[14]。代表となる方程式は、剛性方程式やガラキン法等であり、これを差分系に導くことで数値解を求める。機械や建設の構造解析がこれにあたる。
- (2)においては、その方程式が偏微分方程式として表現する事が出来、これを大きく偏微分方程式系[105]というジャンルで1つに括る。代表となる方程式は、拡散方程式やナビエ・ストークス方程式等[73]があり、原子力分野[3]や流体解析[15]がこれにあたる。
- (3)においては、方程式系の中に積分の考えが一部組み込まれており、少し粗いまとめ方であるが、言葉で表現する上で、これを積分論系というジャンルで1つに括る。代表となる方程式は、シュレディンガー方程式やハートリーフォック方程式等があり、分子科学や天文学がこれにあたる。
- (4)においては、確率の考えが基礎にあり、計画問題も含めてこれを確率論系として1つに括る。正規分布、t分布に代表される統計学や、乱数としての品質管理、計画問題としての数理計画等がこれにあたる。
- (5)においては、無限級数の考えが基礎にあり、フーリエに代表される方法を用いる事が多く、特殊なモデルにも対応する場合もあり、これを特殊計算系というジャンルで1つに括る。これは軍事や資源探査等に見られる信号の解析、特殊モデルの計算が使われている。

特に(1)から(3)においては、行列に導き数値解法へ適合させる事となる[74]。これは方程式を離散化して差分形式に導き、線型方程式(連立一次方程式)で表現して、数値解法で解く事となる[27]。当論文では後述する第4章、第5章にて原子力分野を採り上げ、方程式を離散化して数値解法を適用させ、その中で高速解法に変更させている。(4)は統計学や計画問題の数理計画等であり、(5)は分野の特徴から、フーリエ解析等による数値解析が用いられる形が中心となっている。

この様にして代表的な方程式を離散化して、差分形式により数値解法を求める事になる[12]。科学技術計算全体を(1)から(5)として、おおざっぱに分類して表にしたものが Table1.1 である。

この表では大分類で表示している代表的な方程式の幾つかが、中分類の分野毎の各系に存在する意味である。

Table1.1 科学技術計算の分類方法と応用分野

大分類	中分類	小分類
「有限要素法系」 〔釣り合いの方程式〕 剛性方程式 ガラキン法 ウェーブフロント法 ハミルトン原理	構造解析系 土木解析系	機械系（航空機、自動車、船舶、産業機械） 建築系（一般ビル、高層ビル、耐震、熱管理） 土木系（ダム、橋梁、港湾、地下街） 土質系（トンネル、埋め立て、丘陵造成）
「偏微分方程式系」 〔時間と空間の有限差分〕 輸送方程式・拡散方程式 マクスウェル方程式 ナビエ・ストークス方程式 非線形反応拡散方程式 ポアソン方程式 ラグランジュ方程式 オイラー方程式	原子力系 核融合系 流体系 地球海洋系 宇宙科学系 電気電子系 基礎物理系	（核分裂、炉解析、動特性、炉安全） （中性子、粒子ビーム、ヘリカル、レーザ） （流れ、気象解析、大気拡散、乱流） （大気大循環、海流、自由表面） （人工衛星、電磁波、スペースシャトル） （電磁場解析、回路解析、デバイス、超伝導） （計算物理、シンクロトン）
「積分論系」 〔空間差分と有限の和集合〕 シュレディンガー方程式 分子軌道法 パウリ原理 モンテカルロ法 クオーク伝播関数 ハミルトニアン ハートリーフォック方程式 変分原理	分子科学系 素粒子系 原子核系 天文学系	（分子力学、分子動力学、分子生物学、遺伝子工学） （高エネルギー、格子ゲージ、量子色力学、量子電磁気学） （原子核振動、原子核エネルギー、中間子） （星雲、オーロラ、銀河、恒星進化）
「確率論系」 〔乱数と計画計算〕 一様乱数、正規乱数 正規分布、t分布 カイ2乗分布、F分布	OR系 データ解析系	（数理計画、需要予測、待ち行列、品質管理） （統計学、多変量解析学、実験計画法、ノンパラメトリック統計、医薬解析）
「特殊計算系」 〔無限級数の有限和〕 フーリエ変換 パルス変換 信号解析、特殊モデル計算	軍事系 資源探索系	（潜水艦探査、地下核実験、暗号解読） （地震波、リモートセンシング）

本論文の特徴として、数多くの科学技術計算のベンチマーク実績より、大規模数値計算の研究に焦点を充てて、上記に分類した方法で、社会全体の中で代表的な数値解法が、どの分野に使用されているか研究して調査した。数値解法の種類は、プログラムコードが開発された年代によっても少し変わってくるが、ここでは1950年代に始まり、21世紀初頭迄の60年近くの解法全体を1つにまとめた。数値解法の基礎となるガウス消去法やヤコビ法は、殆ど全ての分野で広く使われていることが分かる。またLU分解法やSOR法等もあらゆる分野で使われている。一方でブロック化法系や三角化法系等は限られた分野でのみ使用されており、最近研究されている前処理を行っての高速解法は一部を除いて、まだ社会に十分に広がってはならず、今後の技術進歩に期待される処が多いと考えられる。近年では、工学分野の数値解法に対して、数値計算のライブラリが使われることが多くなっているが、行列の性質を十分理解の上、最適な解法を使うことが望ましいと考えられる。

以下に示すTable1.2においては、上述して科学技術計算の分野毎と、更に教育一般やハードウェアの性能比較等もアイテムとして採り上げ、良く使用されている代表的な数値解法との関係を記して、一覧表としてまとめている。

Table1.2 数値解析応用事例

○印：使用実績有り																	◎印：良く使用されている																
	教育一般研究	産業界官公庁研究	次世代基礎材料	前処理研究	性能比較尺度研究	構造解析系	流体解析系	原子力核融合系	地球海洋系	宇宙科学系	電気電子系	基礎物理化学系	分子科学系	原子核素粒子系	天文学系	軍事資源探索系	確率論系																
ガウス消去法	◎	◎				○	○	○	○	○	○	○	○	○	○	○	○																
ガウス・ジョルダン法	○	○	◎			○	○		○	○	○	○	○	○	○		○																
ヤコビ法	◎	◎				○	○	○	○	○	○	○	○	○	○																		
ガウス・セidel法	◎	◎				○	○		○	○		○		○	○																		
コレスキー法	○			◎		○			○			○																					
LU分解法	◎			◎	◎	○	○	○	○	○	○	○	○	○	○	○																	
SOR法系	◎	◎				○	○	○	○	○		○	○	○	○	○																	
CG法	○	○				○	○			○		○																					
CR法	○	○					○					○																					
ADI法								◎				○																					
チェビシェフ法								◎				○																					
Stone法							○																										
ブロック化法系						◎																											
三角化法系											◎																						
バンド幅縮小法系						◎																											
ブロックスライン法						◎																											
MG法系		○					○																										
ハイパープレーン法		○					○																										
内積形式非対称解法											○																						
改良型直接法系		○																															
前処理付CG法系	○	○				○	○	○	○	○		○	○	○	○																		
前処理付CR法系	○	○				○	○	○		○		○	○																				
三項対角近似因子法							○			○																							
切除残差法							○																										

これらの応用領域における具体的な数値解法の種類によるまとめ方や比率は、第 35 回数値解析シンポジウムで発表した内容があり、これは別途の参考論文としたい。

数値解析応用事例における解法で、上記 Table1.2 では代表的な解法と、前処理付等まとめた形の区分も合わせて表現しているが、後述する個別の各解法[67]には、その特徴とアルゴリズム、そして応用されている分野について、著者の研究により、その細部を述べる。

こうした事を踏まえて、以降では科学技術分野を飛躍的に伸ばした行列の考えに起点を置き、数値解法の歴史から入り、当論文のオリジナルである基礎から応用へと述べ、続く章では高速化への考え方と、そして研究テーマである原子力分野[2]への数値解法高速化実現について、具体的に述べて行きたい。

1-3. 行列の歴史

この節では、数値計算のルーツを求めた別刷の数値計算と計算機の歴史を要約して、科学技術計算の発展に大きな役割を果たしている行列の誕生から発展について簡単に述べる。

[1] 行列の誕生

ライプニッツ(Leibniz)は、ニュートン(Newton)の方程式を複数個並べて表現することにより、行からなる部分と列からなる部分から構成されることを想定し、行列式の概念の表現である「行列式の原案の考察」(1690)をまとめた。後の時代の解析学者ラグランジュ(Lagrange)は、ライプニッツの行列式の原案を研究発展させて「行列式の表現」(1773)を発表した。更に次の時代、数学者コーシー(Cauchy)は、ラグランジュの行列式の研究に始まり、行列の和、差、積、除(逆)を論じ、行列式体系(「行列式」(1812)の論文発表)を総合的に確立させた。コーシーと同時代のガウス(Gauss)は、コーシーの定義した行列式から連立一次方程式(線形方程式)を構築して「ガウス消去法」(1823)と呼ばれる数値解法を提案した[10]。

[2] 行列の歴史的経緯

コーシーによる行列式の定義(1812)以来、行列には対称、非対称とした形(1860)があることが理解された。やがて行列の分解(コレスキー分解)に始まり、20世紀に入ると行列の置換(1910)の考え、行列の正定数値の考え(1930)が育まれた。次に行列には密なる行列と、疎なる行列があることが理解された(1936)[16]。

そして20世紀の半ばになると、ヘッセンベルグによる行列の変形(1955)の考えが出された。同じ頃、巡回行列の考え(1955)が出された。次に行列の中に、対角を優位とする形が(1960)あり、そこから対角部と非対角部を分離する等の行列の分離の考え(1960)が出された。

計算機が実用化に向けて発達するに連れて、大規模な行列が計算できる様になり、理工学の事象の中には規則的な形をとるものと、規則的ではない不規則な形をとる行列が考え出された(1965)。次に行列の条件数の考えが提案される(1969)。これにより、数値解法を行う為には、計算しにくい形があることが理解される様になる[24]。

近年において、計算機が更に発達するに伴い、大型行列をより速く計算させるべき方法が考え出される様になる。コレスキー分解やLU分解等を、不完全な形の分解として、これを行列の前処理(1970)と称してCG法やCR法等に結びつける研究がされ、現在更に発展を続けている。行列の比較化としての提案(1988)や、ベクトル化的な考え(2005)も出される様になって来た。

[3] 直接法の誕生、反復法の誕生

ガウス(Gauss)を原点とするガウス消去法は、行列としての数値計算の原点となっている。その後、ガウス型を改良した方法等に始まり、行列の置かれている位置に対して、計算対象とする方法を数値計算の分類上、直接法(Direct Method)と言われ様になった。

ガウスと世代の近いヤコビ(Jacobi)は、先駆者フィボナッチの影響から、無限小の概念を打ち立てたオレムの数学を研究し、行列の数値解法の研究に反復することによる無限小の繰り返しの概念を考え出した[52]。後に反復法(Iterative Method)と言われ、この反復計算は多くの解法

を生み出して行く。最近の多くの研究がされている反復法の発展を採り上げる。反復計算による無限小の概念を研究したヤコビは、新しい形の解法を提唱した。ヤコビ法は、反復法の先駆的研究として位置づけされるものである。この反復法の研究は、ザイディルによりガウス流の考えを入れて、ヤコビの改良型であるガウス・ザイディル法を提唱した。ヤコビの反復法を改良したこの形は、やがて改良型反復法として幾つもの方法を生みだして行く[54]。

特に20世紀半ばになると、多くの改良型反復法が提唱される様になる。ヤングとフランケルが提唱したSOR法(1950)は、ガウス・ザイディル法に加速パラメータを加味した解法であり、多くの理工学事象に適用され、SOR法の枝葉としてSLOR法を始めとして、Odd-EvenSOR法、SSOR法他幾つものSOR法改良型が提唱される。これらをSOR法系として分類したい。

ヘステンズとスティーフェルは、一次独立なベクトルを定義し、空間内を直交性を保ちながら探索する考え方として「共役勾配法(CG法)」(1952)を提唱した[53]。この解法は、その後の時代になるが、各種の前処理を施す事で、計算の効率化を目指したCG法の枝葉が後に幾つも研究され発展するが、その集まりをここではCG法系（共役勾配法系）と分類したい。同じ加速パラメータでも反復法の長所である高速解法の中に、直接法の長所である厳密解法の組み込み研究を行い、行方向求解、列方向求解による交互方向の解法をピースマンとラチフォードが提唱した[19]。これは「ADI法」(1955)と呼ばれ、理工学事象に広く使われている。

直接法の進展では、ジョルダンは数学の応用として行列を有界部分集合に見立てて、ある条件を満たす測度を対応させる考えから、ガウス消去法を改良して「ガウス・ジョルダン法」(1868)を提案した。この解法は、ガウス流を原点とする直接法の1つであり、今日では数値計算の基礎理論として、掃き出し法の名で各方面に活用されている。ガウス法を含めて、後に直接法と分類されて、これも後の時代に多くの解法を生み出して行く[48]。

同じ頃、コレスキーは、最小自乗法の考えから下三角行列と上三角行列(下三角行列の転置行列)に分解する事により、効率良く解ける事を提案した(1870)。後に「コレスキー法」として普及した。こうした考えは、その後計算機の発達と共に、自然科学の分野で対象としたモデルが汎用化されるにつれて、適用が広がる様になってきた。このコレスキーによる分解の考えは、今日不完全コレスキー分解として大規模疎行列の先駆的研究テーマになっている。現在の行列における前処理法の計算理論にも、大きな期待が寄せられている。この直接法の流れも発展を続け、三角化法やバンド幅縮小法、ブロック化法、大規模行列直接解法、そして高速改良型消去系等として、多くの斬新な解法を育んで発展して行く。

ここで反復法に対して、数値計算が広く普及する時代になると、多項式解法として反復に準じた方法による「チェビシェフ(Chebyshev)準反復法」(1960)が加速パラメータを加味した解法として、収束性の良さから実用に効果を発揮した。これは固有値問題の高速計算の必要性から、数値解のステップの安定性をもたらすと重宝され、当時の工学研究の花形であった原子力問題の数値解法に採り入れられた。

近年になって、CG法系からの発展として、残差に目を向けた共役残差法(CR法)とその前処理系が幾つも提唱され、理工学分野に應用が広がり始めた。代表的な解法として、PCR法、ILUCR法、MILUCR法等が提案されている。これらを1つにまとめてCR法系（共役残差法系）として分類したい。この共役残差法系は、残差を最小にする形で収束し、連立一次方程式の条件数が良い場合、収束が速い特徴がある。

その後80年代後半になると、スーパーコンピュータ[6]の発達と共に、SOR法が発展してベクトル計算機向けに研究され、「マルチカラー法」(1985)が提唱された。更に行列の不完全分解への研究が盛んになり、そしてこの前処理法としての研究進展は、行列の条件数を改良する事を目指したスケーリング効果の考えの進展に始まり、速水(K. Hayami)の「SCG法」(1985)等により大型行列の高速解法が実現された。スパース(疎)性の高い大型行列であれば、非対称行列であっても、これを対称行列に変換した後、このSCG法を適用することで、時間短縮が比較的良く期待出来るところから、今後の工学分野への幅広い活用が見込まれる。こうした数値解法の進展を含めて説明したい。

1-4. 連立一次方程式

解くべき連立一次方程式を次に定義する。

$$(1.1) \quad \mathbf{Ax} = \mathbf{b}$$

ここでは連立一次方程式を求める上で、行列計算の原点である直接法の代表、ガウス消去法を始め、反復法の代表、ヤコビ法を始めとする社会に広く普及している基礎的な解法の探求と社会で使用されている応用分野の調査を記載したい。

以下、連立一次方程式における具体的な数値解法を紹介し、その特徴や社会での応用領域について述べる。ここで数値計算が提案された時期を、年代に添って大きく分類したい。数値解法は直接法と反復法に大別されるが、それぞれにおいて、1つは19世紀迄に提案された解法を古典的解法として、次に20世紀以降1960年代迄を近代的解法として、そして反復法の前処理の考えが芽生えた頃、つまり1970年頃から現代迄を現代的解法として分類する。尚、第4節、数値計算最近の潮流に紹介する解法を先端的現代理論の解法として論述したい。 : : : : : : : : : : : : : : :

1-5. 直接法、計算のプロセス

もとの係数行列の形は各計算ステップ毎に刻々と変化し、ゼロ要素に至るまで、計算対象としてノンゼロに置き換えて逐次計算をする為、中小規模の密行列に適し、計算の安定性と厳密解が期待出来る[58]。大規模行列では一般に、次元数 n の $k(\geq 3)$ 乗回の計算ステップを必要とするため、改良型の直接法として、計算回数を減らすブロック化や、行又は列の順序付けを行うオーダリングの考えが提案されている[57]。まずは時代を分類して、古典的解法である直接法の基本から述べる。

1-5-1. 古典的解法

19世紀初期に提案されたガウス消去法を初めとして、ガウス・ジョルダン法やコレスキー法が古典的数値解法の範疇に取り入れる事が出来る。計算機の発達が充分ではなかった時代、密行列を対象にした解法であるが、数値解法の基礎となるべき、堅実な考えの基、確実に数値解が進んで行く。つまり現代解法で注意されるべく条件数等の良し悪しや、固有値問題の歪み等には依存しない長所がある。更に行列の対称非対称や、規則不規則の区分もなく解法が適用出来る。そして数値解法としての理論も分かり易く、教育関係に多く用いられ、社会に広く普及している。但し、200年近い前の時代であり、規則・不規則行列や、そして疎行列や大規模行列の考えの無かったもので、現代必要とされている大規模疎行列の解法に適用すると、多くの計算時間を必要とする[76]。

(1) ガウス消去法 (Gaussian Elimination Method)

パスカル(Pascal)の計算機(機械計算機,1642)を改良して、計算研究をしていたライプニッツ(Leibniz)は、ニュートン(Newton)方程式を複数個並べた「行列式の原案(1690)」を発表した。後世ラグランジュ(Lagrange)はライプニッツの「行列式の原案」を研究して行く中で、表現方法の工夫をした「行列式表現(1773)」の論文を発表する。1812年数学者コーシー(Cauchy)は、このラグランジュ論文を研究して行く時に解析学の立場から行列式を定義した。現在にみる行列式の理論体系である。コーシーと同時代に生きたガウス(Gauss)はコーシーの定義した行列をそれ迄の解析的考え(古典数学としての強存在)から発展させ、「ガウスの消去法(1823)」を提案した。これはそれ以前の変数逐次の消去法に比べて大革命であり、応用数学の分野において、後世に偉大な業績を残した。行列における数値解法の始まりと言える。

現在の応用領域として、教育の為の数値計算を始めとして、産業界全般の数値計算、基本的な基礎研究、基礎物理や基礎化学等、そしてあらゆる工学分野における数値計算として、社会に最も広く使われている。特に1950年代から1960年代迄の工学分野は、この解法で占められていた。他の多くの解法が条件数や対称、非対称、対角優位性等に影響されるものが多いがあるが、この解法はそうした問題に全く関係なく、あらゆる分野に適合する長所がある。

A が大きくない行列では、この古典的なガウス消去法が優先される。ガウス消去法は、基本的な解法を含めて、幾つかの方法があるが、ここでは **A** を基本変換と置換変換の繰り返しによって、上三角行列 **U** に変換する方法を述べる。

基本変換

第 k 段の消去における基本変換 $\mathbf{M}_k$ は

$$(1.2) \quad \mathbf{M}_k = \begin{pmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{k+1,k} & 1 & & \\ & & m_{k+2,k} & & \ddots & \\ & & & & & \ddots & \\ 0 & & & m_{n,k} & & & 1 \end{pmatrix}$$

ここで式 (1.2) にみる $m_{ik} (k+1 \leq i \leq n)$ は乗数 (multiplier) と呼ばれ、 $\mathbf{M}_k$ を左から **A** に掛けることで **A** の第 i 行 ($k+1 \leq i \leq n$) に第 k 行の m_{ik} 倍を加えて、新しい第 i 行を作ったとき、新しい (i, k) 要素が 0 になる様に

$$(1.3) \quad m_{ik} = -\frac{a_{ik}}{a_{kk}}, \quad i = k+1, \dots, n$$

と決められる。式 (1.3) の a_{kk} を第 k 段階の消去におけるピポット (pivot) と呼んでいる。ピポットが小さな値になったときでも、ピポット選択をしないと精度が悪くなる。

第 k 段階の消去において、 $a_{ik} (k \leq i \leq n)$ の中から絶対値最大のものをピポットに選んで、その属する行と第 k 列を入れ替えること。行の入れ替えは、置換行列を左から掛けることに相当する。このピポット選択はガウス消去法は、重要で安定した解法となる。

$\mathbf{E}$ を誤差として、連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ の数値解を $\tilde{\mathbf{x}}$ とすれば、この $\tilde{\mathbf{x}}$ は

$$(1.4) \quad (\mathbf{A} + \mathbf{E})\tilde{\mathbf{x}} = \mathbf{b}$$

の真の解とみなす事が出来る。

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \varepsilon_M$$

である。数値解 $\tilde{\mathbf{x}}$ に対する残差を $\mathbf{r}$ とするとき、式 (1.4) から

$$(1.5) \quad \mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} = \mathbf{E}\tilde{\mathbf{x}}$$

従って、式 (1.5) から相対残差は

$$\frac{\|\mathbf{r}\|}{\|\mathbf{A}\|\|\tilde{\mathbf{x}}\|} \leq \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \varepsilon_M$$

となり、ガウス消去法では イプシロン ε_M 以下になる。

(2) ガウス・ジョルダン法 (Gauss-Jordan Reduction Method)

数学者ジョルダン(Jordan)は、行列を数学の集合論における有界部分集合に見立てて、測度の概念を取り入れ Gauss の消去法を改良して、改良型直接法を提案した(1868)。ガウス・ジョルダン法と呼ばれている。ガウス消去法に比べて計算ステップは多くかかるが、よりスムーズに収束する事が出来る。

応用領域として、ガウス消去法とほぼ同じで、教育の為の数値計算を始めとして、産業界全般の数値計算、基本的な基礎研究、基礎物理や基礎化学等による数値計算として、社会に最も広く使われている。前項ガウス消去法と同じく、1950 年代に始まり 1960 年代迄の工学分野は、多くがこの解法を用いている。この解法も条件数や対称、非対称、対角優位性等に影響されることがない。計算する上で、計算方式の並列性に長所があり、現在では並列計算の面で研究されている特長がある。

計算方式

第 k 段の消去で第 k 行以外の x_k を含む項を同時に消去してしまう。

この操作では第 n 段迄の消去で方程式は次の形に帰着される。

$$(1.6) \quad \begin{array}{rcl} a_{11}x_1 & & = b_1 \\ a_{22}x_2 & & = b_2 \\ a_{33}x_3 & & = b_3 \\ & \dots\dots\dots & \\ a_{n-1,n-1}x_{n-1} & & = b_{n-1} \\ & \dots\dots\dots & \\ a_{nn}x_n & = & b_n \end{array}$$

式 (1.6) は
$$\left[\begin{array}{l} k = 1, 2, \dots, n \\ x_k = b_k \times d_k \end{array} \right]$$

の形で数値解が求められる。第 k ステップ目における行ごとの消去の演算は、第 1 行から第 n 行まで一斉に行うことが出来、各行の演算は独立に出来る。つまりこの解法は並列計算に向いた解法であると考えられる。現代では掃き出し法とも呼ばれる。必要な演算回数は行列の大きさを n とすると、

ガウス消去法が約 $\frac{1}{3}n^3$ 回であるのに比べて、ガウス・ジョルダン法では約 $\frac{1}{2}n^3$ 回であり、ガウス消去法よりも多くなる。つまり同じ数値解を求めるのに、ガウス消去法より 1.5 倍の基本演算を必要としている。

(3) コレスキー法 (Cholesky Decomposition Method)

測量士であったコレスキー(Cholesky)は最小自乗法の考えから発展させて、行列を下三角行列 (Lower triangular matrix) と上三角行列 (Upper triangular matrix) (下三角行列の転置行列 (Transposed matrix)) に分解する事により、効率良く解ける方法を提案した(1870)。コレスキーの死後、友人がコレスキー法と命名し普及させた。行列分離の考えは古くから存在していたが、行列分解を提案したコレスキーの考えは、行列による各種の分解の基礎を作り、その後における行列分解、行列不完全分解への考え方の動機付けとなり、行列分解理論の発展へ大きく貢献した。行列に関する後世の多くの研究者達は、このコレスキー分解の考え方を基礎とした研究を行い、成果を上げている。

後の時代に提案された CG 法に対して、前処理法としての役目が存在している。完全にコレスキー分解するよりも、不完全な形のコレスキー分解の後、CG 法に導くことで大規模な対称行列の場合、高速に解かせる事が証明出来た。コレスキー法の応用領域として、ガウス消去法と同じく教育の為の数値計算を始め、一般研究領域や構造解析、計算物理等、社会の多くの分野に適用されている。特に 1970 年代迄の工学分野に多く用いられている。

コレスキー法の計算ステップ

対称正定値行列 (Symmetric positive definite matrix)

$\mathbf{A} = \mathbf{A}^T$ であれば $\mathbf{A}$ は symmetric である。

$\mathbf{A}$ が $\mathbf{x} \neq \mathbf{0}$ なるすべての n 次元ベクトルに対して、

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$$

を満たすとき $\mathbf{A}$ は正定値 (positive definite) である。つまり $\mathbf{A}$ の対角成分は全て正である。

$$a_{ii} > 0$$

$\mathbf{A}$ が対称正定値行列であれば、次の方法でコレスキー分解する事が出来る。

$\mathbf{A}$ を下三角行列と上三角行列の形をそれぞれ $\mathbf{L}$ と $\mathbf{U}$ で表現する。

$\mathbf{A}$ の対角行列を $\mathbf{D}$ で表現すると $\mathbf{D}^{1/2}$ は次式になる。

$$(1.7) \quad \mathbf{D}^{1/2} = \begin{pmatrix} \sqrt{a_{11}^{(1)}} & & & \mathbf{0} \\ & \sqrt{a_{22}^{(2)}} & & \\ & & \ddots & \\ \mathbf{0} & & & \sqrt{a_{nn}^{(n)}} \end{pmatrix}$$

$$(1.8) \quad \tilde{\mathbf{L}} = \mathbf{L} \mathbf{D}^{1/2}$$

と置くことで、式 (1.8) から $\mathbf{A}$ は

$$(1.9) \quad \mathbf{A} = \tilde{\mathbf{L}} \tilde{\mathbf{L}}^T$$

で表現する事が出来る。

$\mathbf{D}$ は次の形になる。

$$(1.10) \quad \mathbf{D} = \begin{pmatrix} a_{11}^{(1)} & & \mathbf{0} \\ & a_{22}^{(2)} & \\ & & \ddots \\ \mathbf{0} & & & a_{nn}^{(n)} \end{pmatrix}$$

式 (1.10) を用いて $\mathbf{A}$ が正定値でない場合、 $\mathbf{L}$ の対角行列を 1 とおく分解としての表現をする。

$$(1.11) \quad \mathbf{U} = \mathbf{D}\mathbf{V}$$

で表現できる。

$$\text{但し、} \quad \mathbf{V} = \begin{pmatrix} 1 & v_{12} & v_{13} & \cdots & v_{1n} \\ & 1 & v_{23} & \cdots & v_{2n} \\ & & 1 & \cdots & \cdot \\ & & & \ddots & \cdot \\ & & & & 1 \end{pmatrix}, \quad v_{ij} = \frac{a_{ij}^{(i)}}{a_{ii}^{(i)}}$$

この時

$$(1.12) \quad \mathbf{A} = \mathbf{LDV}$$

$\mathbf{A}$ は対称であるから $\mathbf{V} = \mathbf{L}^T$ であり

$$(1.13) \quad \mathbf{A} = \mathbf{LDL}^T$$

となる。

式 (1.1) により $\mathbf{Ax} = \mathbf{b}$ から、式 (1.13) を代入して

$$(1.14) \quad \mathbf{LDL}^T \mathbf{x} = \mathbf{b}$$

の形となる。ここで式 (1.14) による $\mathbf{DL}^T \mathbf{x} = \mathbf{y}$ と置くことで、三角行列を係数に持つ 2 つの方程式として

$$(1.15) \quad \mathbf{Ly} = \mathbf{b}$$

$$(1.16) \quad \mathbf{DL}^T \mathbf{x} = \mathbf{y}$$

と置くことが出来、先ず式 (1.15) を解いて $\mathbf{y}$ を求め、次に式 (1.16) を解くことで、数値解が求められる。

1.5.2. 近代的解法

第二次大戦後、クラフトの提案した LU 分解は、その後の計算機の発達に合わせて広く使用されるようになった。行列の分解の考え方は、19 世紀のコレスキーの時代に提案されているが、三角行列としての新しい考えは、計算機を使用して逐次形式の数値解法として、大きな貢献をもたらした。現代のコンピュータの性能を評価するベンチマーク用として LINPACK があるが、これは LU 分解の理論を計算として取り込んでいる。この時代には、行列のブロック的处理が着目され、代表的な解法に Stewart 法がある。このブロック的处理は、グラフ理論研究の応用として、主と部分に分けて、ブロック毎に分けての数値解法である。疎な行列は行と列を適当に入れ替える事により、適当なブロックに集める事が出来ると言う点で疎な大規模行列をブロック分割して直接的に解法する考えである。

(1) LU 分解法 (LU Decomposition Method)

20 世紀初めのクラウト(Crout)の論文による。行列の適切な三角分解による提案に基づく。コレスキーの分解を研究してゆくうち、行列がきれいな形に分解される事に気づいた。行列が下(Lower)三角行列と上(Upper)三角行列の形の分解形式となった為、LU 分解と呼ばれる様になった。

連立一次方程式の解法の中で、前進消去後退代入の考えとして普及する。LU 分解計算処理においてプロセッサ中の計算格子点 (ノード点) が遊びがなく、スムーズに計算移行する為、性能比較の統一基準として取り扱い易い解法である。こうした計算処理に優れている所から、現在コンピュータのハードウェアの CPU 性能比較評価に用いられている。

応用領域として、ガウス消去法とほぼ同じで、教育の為の数値計算を始めとして、産業界全般の数値計算、基本的な基礎研究、基礎物理や基礎化学等による数値計算として、社会に最も広く使われている。流体解析や、地球科学、海洋科学、原子核、天文学、電磁界、更に資源探査系や軍事系の研究分野にも広く使われている。特に 1980 年代迄の工学分野に多く見られる。不規則密疎混在行列である電磁界解析回路シミュレーション分野ではこの LU 分解が効力を発揮している。

コレスキー法の前処理と同じく、LU の前処理の研究が盛んに行われている。完全に LU 分解する前に、CG 法へ渡す。これは疎な大規模行列であれば、疎という性質を活かして、ノンゼロ部分のみに着目する考えが構成できる。計算機が発達して大規模な計算が可能になるに従い、不完全 LU 分解が着目される様になってきた。最近では不完全 LU 分解と CG 法のセットが数値計算研究の中心となっている。

LU 分解法の計算ステップ

$\mathbf{A}$ が正則である場合、 $\mathbf{A}$ を下三角行列 $\mathbf{L}$ と上三角行列 $\mathbf{U}$ に分解する。

$$\mathbf{L} = \begin{pmatrix} l_{11} & & \mathbf{0} \\ l_{21} & l_{22} & \\ \cdot & \cdot & \cdot \\ l_{n1} & l_{n2} & \cdot & l_n \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 1 & u_{12} & \cdot & u_{1n} \\ & 1 & \cdot & u_{2n} \\ & & \cdot & \cdot \\ \mathbf{0} & & & 1 \end{pmatrix}$$

具体的な計算方法

幾つかの方法があるが、ここではガウス消去法の行列式から導く方法を述べる。

$$(1.17) \quad \mathbf{M}_1 = \begin{pmatrix} 1 & & & & \mathbf{0} \\ & \cdot & & & \\ & -m_{21} & 1 & & \\ & -m_{31} & 0 & 1 & \\ & \cdot & 0 & 0 & \cdot \\ & \cdot & \cdot & \cdot & \cdot \\ & -m_{n1} & 0 & 0 & \cdot & \cdot & 1 \end{pmatrix}$$

式 (1.17) のこれは $\mathbf{M}_1 = \mathbf{I} - \mathbf{m}_1 \mathbf{e}_1^T$

$\mathbf{I}$ は $n \times n$ 単位行列、 $\mathbf{m}_1$ 及び $\mathbf{e}_1$ は、次式で与えられる n 次元ベクトルである。

$$\begin{aligned} \mathbf{m}_1 &= (0, m_{21}, m_{31}, \dots, m_{n1})^T \\ \mathbf{e}_1 &= (1, 0, 0, \dots, 0)^T \end{aligned}$$

$\mathbf{e}$ は単位ベクトルである。

ここでベクトルの添え字を $\mathbf{A}^{(1)} = \mathbf{A}$ 、 $\mathbf{b}^{(1)} = \mathbf{b}$ の形とする。
行列 $\mathbf{A} = \mathbf{A}^{(1)}$ に

$$(1.18) \quad \mathbf{M}_1 \mathbf{A}^{(1)} = \mathbf{A}^{(2)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdot & \cdot & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdot & \cdot & a_{2n}^{(2)} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ 0 & a_{n2}^{(2)} & \cdot & \cdot & a_{nn}^{(2)} \end{pmatrix}$$

なる演算を行う。この時、方程式は $\mathbf{M}_1 \mathbf{A}^{(1)} \mathbf{x} = \mathbf{M}_1 \mathbf{b}$ と書くことが出来る。
式 (1.17) は、一般に

$$(1.19) \quad \mathbf{M}_k = \mathbf{I} - \mathbf{m}_k \mathbf{e}_k^T = \begin{pmatrix} 1 & & & & & & & & \mathbf{0} \\ 0 & 1 & & & & & & & \\ \cdot & \cdot & \cdot & & & & & & \\ 0 & 0 & \cdot & & 1 & & & & \\ \cdot & \cdot & \cdot & -m_{k+1,k} & 1 & & & & \\ \cdot & \cdot & \cdot & -m_{k+2,k} & 0 & 1 & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & \\ 0 & 0 & \cdot & -m_{nk} & 0 & \cdot & \cdot & 1 \end{pmatrix}$$

と置くと、式 (1.19) の消去の第 k 段の操作は、 $\mathbf{M}_k \mathbf{A}^{(k)} = \mathbf{A}^{(k+1)}$ と書くことが出来る。
但し、

$$(1.20) \quad \begin{aligned} \mathbf{m}_k &= (0, \dots, 0, m_{k+1,k}, m_{k+2,k}, \dots, m_{nk})^T \\ \mathbf{e}_k &= (0, \dots, 0, 1, 0, \dots, 0)^T \end{aligned}$$

最後の第 $n-1$ 段まで消去を進めた結果を $\mathbf{U}$ とおくと

$$(1.21) \quad \mathbf{U} = \mathbf{M}_{n-1} \mathbf{M}_{n-2} \cdots \mathbf{M}_1 \mathbf{A}^{(1)} = \mathbf{A}^{(n)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdot & \cdot & a_{1n}^{(1)} \\ & a_{22}^{(2)} & & & a_{2n}^{(2)} \\ & & \cdot & & \cdot \\ & & & \cdot & \cdot \\ \mathbf{0} & & & & a_{nn}^{(n)} \end{pmatrix}$$

となる。右辺ベクトル $\mathbf{b}^{(1)} = \mathbf{b}$ にも

$$(1.22) \quad \mathbf{M}_{n-1} \mathbf{M}_{n-2} \cdots \mathbf{M}_1 \mathbf{b}^{(1)} = \mathbf{b}^{(n)} = (b_1^{(1)} \quad b_2^{(2)} \quad \cdot \quad \cdot \quad b_n^{(n)})^T$$

の操作を行うと、元の連立一次方程式 $\mathbf{A} \mathbf{x} = \mathbf{b}$ による式 (1.21) の上三角行列 $\mathbf{U}$ を用いて、同値な方程式 $\mathbf{U} \mathbf{x} = \mathbf{b}^{(n)}$ に帰着される。
式 (1.21) より

$$(1.23) \quad \begin{aligned} \mathbf{A} &= (\mathbf{M}_{n-1} \mathbf{M}_{n-2} \cdots \mathbf{M}_1)^{-1} \mathbf{U} \\ &= \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \cdots \mathbf{M}_{n-1}^{-1} \mathbf{U} \end{aligned}$$

$$(1\cdot24) \quad \mathbf{M}_k^{-1} = \mathbf{I} + \mathbf{m}_k \mathbf{e}_k^T = \begin{pmatrix} 1 & & & & & & & \mathbf{0} \\ 0 & 1 & & & & & & \\ \cdot & \cdot & \cdot & & & & & \\ 0 & 0 & \cdot & 1 & & & & \\ \cdot & \cdot & \cdot & \mathbf{m}_{k+1,k} & 1 & & & \\ \cdot & \cdot & \cdot & \mathbf{m}_{k+2,k} & 0 & 1 & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ 0 & 0 & \cdot & \mathbf{m}_{nk} & 0 & \cdot & \cdot & 1 \end{pmatrix}$$

$$(1\cdot25) \quad \begin{aligned} \mathbf{M}_k \mathbf{M}_k^{-1} &= (\mathbf{I} - \mathbf{m}_k \mathbf{e}_k^T)(\mathbf{I} + \mathbf{m}_k \mathbf{e}_k^T) \\ &= \mathbf{I} - \mathbf{m}_k (\mathbf{e}_k^T \mathbf{m}_k) \mathbf{e}_k^T \end{aligned}$$

式 (1・20) の形から $\mathbf{e}_k^T \mathbf{m}_k = 0$ であり、 $\mathbf{M}_k \mathbf{M}_k^{-1} = \mathbf{I}$ となるから

$$(1\cdot26) \quad \begin{aligned} \mathbf{L} &= (\mathbf{M}_{n-1} \mathbf{M}_{n-2} \cdots \mathbf{M}_1)^{-1} \\ \mathbf{e}_i^T \mathbf{m}_j &= 0 \quad (i \leq j) \quad \text{を用いて} \\ \mathbf{L} &= \begin{pmatrix} 1 & & & & \mathbf{0} \\ \mathbf{m}_{21} & 1 & & & \\ \mathbf{m}_{31} & \mathbf{m}_{32} & 1 & & \\ \cdot & \cdot & \cdot & \cdot & \\ \mathbf{m}_{n1} & \mathbf{m}_{n2} & \cdot & \mathbf{m}_{n,n-1} & 1 \end{pmatrix}, \quad \mathbf{m}_{ij} = \frac{\mathbf{a}_{ij}^{(j)}}{\mathbf{a}_{jj}^{(j)}} \end{aligned}$$

式 (1・23) より、与えられた連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ の係数行列 $\mathbf{A}$ が正則ならば、その対角成分がすべて 1 である下三角行列 $\mathbf{L}$ と、上三角行列 $\mathbf{U}$ の積 $\mathbf{A} = \mathbf{LU}$ の形に LU 分解 (LU decomposition) する事が出来る。

この LU 分解法は、スーパーコンピュータ性能比較 LINPACK 100×100 , LINPACK TPP 1000×1000 等で、国際標準として用いられている。

(2) Stewart 法 (Stewart Method)

ブロック化法 (Blockize Mthod) として、幾つかの解法のある中での 1 つである。Stewart より、1965 年に提案された。Stewart はグラフ理論研究の応用として、このグラフとして用いられたものを、主と部分に分けて、ブロック毎に分けて数値解法の研究を行った。この応用面での考え方は、疎な大規模行列をブロック分割して直接的に解法する考えである。疎な行列は行と列を適当に入れ替える事により、適当なブロックに集める事が出来る。並び替えで得た小行列群を新しい核に対応させた対角ブロックに変換させる。この状態の対角ブロック行列を LU 分解する。各ブロック毎に、求解への時間が短縮される事を提案した。

行列を並び替えによって、新しい核に対応した小行列を考える。扱っている小行列の各行の行カウンターのうち、最初の行 $\mathbf{r}$ を探す。この行 $\mathbf{r}$ に非ゼロ要素を持つ列を作業用リストとする。この作業用リスト中の各列について、この列と同じ番号を持つ行の上に非ゼロ要素を持つ列の作業用リストに追加する。この操作を繰り返し、作業用リストに追加すべき列が無くなったら、作業用リストの中の

列群は1つのブロックを構成する。これは三角行列になって行く。最終的に変換された行列は、上三角行列に近く LU 分解に要する時間は少なくなる方向へ向かう。そしてブロック内では軸選択も可能になる。この状態の対角ブロック行列を LU 分解する。各ブロック毎に、求解への時間が短縮される事を提案した。ブロックが生成出来る型は限られ、ある一定量の「ゼロ」要素が必要である。

応用領域として、大規模な行列をブロック化させる意味から、近年計算機の発達と共に、大規模モデルが作成された構造解析の分野に、活用が見いだされている。

Stewart 法の計算ステップ

$\mathbf{A}$ の行列を入れ替えて次の形の対角ブロック行列を作る。行列 $\mathbf{A}$ のブロック毎の LU 分解を行う。

$$(1.27) \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 & & & \\ \mathbf{C}_1 & \mathbf{A}_2 & \mathbf{B}_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \mathbf{C}_{p-2} & \mathbf{A}_{p-1} & \mathbf{B}_{p-1} \\ & & & \mathbf{C}_{p-1} & \mathbf{A}_p \end{pmatrix}$$

$$= \begin{pmatrix} \Gamma_1 & & & & \\ \Delta_1 & \Gamma_2 & & & \\ & \Delta_2 & \cdot & & \\ & & \cdot & \cdot & \\ & & & \Delta_{p-1} & \Gamma_{p-1} \end{pmatrix} \begin{pmatrix} \mathbf{I}_1 & \mathbf{M}_1 & & & \\ & \mathbf{I}_2 & \mathbf{M}_2 & & \\ & & \cdot & \cdot & \\ & & & \cdot & \mathbf{M}_{p-1} \\ & & & & \mathbf{I}_p \end{pmatrix} = \mathbf{LU}$$

ここで $\mathbf{I}_i$ は各単位行列である。

式 (1.1) において、 $\mathbf{x}$ と $\mathbf{b}$ を $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$, $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_p)$ と表現し、次の形に展開する。

$$\mathbf{M}_{i+1} = \mathbf{B}_{i+1}(\mathbf{A}_{i+1} - \mathbf{C}_i \mathbf{M}_i)^{-1}, \quad \mathbf{M}_1 = \mathbf{B}_1 \mathbf{A}_1^{-1}$$

$$\mathbf{z}_1 = \mathbf{M}_1 \mathbf{b}_1 \mathbf{B}_1^{-1}$$

$$\mathbf{z}_{i+1} = \mathbf{M}_{i+1}(\mathbf{b}_{i+1} - \mathbf{C}_i \mathbf{z}_i) \mathbf{B}_{i+1}^{-1} \quad (i = 1, \dots, p-1)$$

$$\mathbf{x}_p = \mathbf{z}_p$$

$$\mathbf{x}_i = \mathbf{z}_i - \mathbf{M}_i \mathbf{x}_{i+1} \quad (i = n-1, \dots, 1)$$

ここで、 $\mathbf{z}$ は $\mathbf{Lz} = \mathbf{b}$ の解である。解 $\mathbf{x}$ はブロックに関する計算で求める事が出来る。ブロック行列の Γ_i や Δ_i については求める必要がない長所がある。

1-5-3. 現代的解法

ブロック系の中で分割理論を取り入れた Dissection 分割法の考えが出来た。高速改良型消去法の考えとして、二列同時消去法に代表される直接の高速消去法が提案された。この解法は大型行列にも

スムーズに適用出来る。計算機の大型化に伴って、大規模行列の直接解法として、乗積形逆行列法（内積形式非対称疎行列直接解法）を始め、ブロックスカイライン法（対称疎行列解法）等が新しく提案された。一方で 非対称疎行列の直接解法がある。これは応用として流体解析、回路解析等における大型の非対称疎行列では、幾度も同じ様な系を解く場合がある。こうした疎行列では、非ゼロ要素数 $\div n$ 自乗の密度をより効果的に活用する考え方である。旧来の直接法では、大規模行列に適用すると、ゼロ要素もノンゼロ要素にして、疎である特徴が失われ、多くの計算時間を必要とした。直接法は条件数にも依存せず、正確で安定した解を求める事が出来るが、計算時間がかかる事に問題があった。現代的な新しい解法により、大規模疎行列に対して、確実に収束解へ進み、安定的に数値解が求められる新しい解法の登場である。

(1) Dissection 分割法 (Dissection Division Method)

有限要素法で現れる網目分割で出来た構造物の処理から考えられた。全体を均等な複数個の部分構造に分けて、対角ブロック行列を作る。これら境界上の節点に着目して節点集合により、2つの連結集合に分ける。そこで分割と新しい行列を作る。このブロック行列を LU 分解する。

応用領域として、構造解析の分野に現れる事があるが、それは近年多くの研究がされている有限要素法の網目分割で出来た処理に適している。それは均等な複数個の部分構造に分けて、行列を並び替えることが特徴となる。

Dissection 分割法の計算ステップ

次のブロックを対象とする。

$$(1.28) \quad \begin{pmatrix} \mathbf{A}_{kk} & \mathbf{A}_{kl} \\ \mathbf{A}_{lk} & \mathbf{A}_{ll} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{kk} \mathbf{L}_{kk}^T & \mathbf{0} \\ \mathbf{A}_{lk} & \tilde{\mathbf{A}}_{ll} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

に分割する。式 (1.28) により

$$(1.29) \quad \mathbf{A}_{kk} = \mathbf{L}_{kk} \mathbf{L}_{kk}^T$$

$$(1.30) \quad \mathbf{A}_{kl} = \mathbf{L}_{kk} \mathbf{L}_{kk}^T \mathbf{W}$$

$$(1.31) \quad \mathbf{A}_{ll} = \mathbf{A}_{lk} \mathbf{W} + \tilde{\mathbf{A}}_{ll}, \quad \tilde{\mathbf{A}}_{ll} = \mathbf{A}_{ll} - \mathbf{A}_{lk} \mathbf{W}$$

従って、式 (1.29) により $\mathbf{A}_{kk}$ をコレスキー分解した後、式 (1.30) により $\mathbf{W}$ を求め、更に式 (1.31) より $\tilde{\mathbf{A}}_{ll}$ を求める。最後に $\mathbf{A}_{pp}$ にコレスキー分解する。

代入計算は

$$(1.32) \quad \begin{pmatrix} \mathbf{L}_{kk} \mathbf{L}_{kk}^T & \mathbf{0} \\ \mathbf{A}_{lk} & \mathbf{L}_{ll} \mathbf{L}_{ll}^T \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_l \end{pmatrix} = \begin{pmatrix} \mathbf{b}_k \\ \mathbf{b}_l \end{pmatrix}$$

ここで式 (1.32) は $\mathbf{LUx} = \mathbf{b}$ の形となり $\mathbf{Lz} = \mathbf{b}$ と $\mathbf{Ux} = \mathbf{z}$ の2段階に分けて行う。

前進代入は

$$(1.33) \quad \begin{pmatrix} \mathbf{L}_{kk} \mathbf{L}_{kk}^T & \mathbf{0} \\ \mathbf{A}_{lk} & \mathbf{L}_{ll} \mathbf{L}_{ll}^T \end{pmatrix} \begin{pmatrix} \mathbf{z}_k \\ \mathbf{z}_l \end{pmatrix} = \begin{pmatrix} \mathbf{b}_k \\ \mathbf{b}_l \end{pmatrix}$$

より、 $\mathbf{L}_{kk} \mathbf{L}_{kk}^T \mathbf{z}_k = \mathbf{b}_k$ を得る。

前進代入、後退代入は次式の形となる。

$$(1.34) \quad \begin{cases} \mathbf{L}_{kk} \mathbf{y}_k = \mathbf{b}_k \\ \mathbf{L}_{kk}^T \mathbf{z}_k = \mathbf{y}_k \end{cases}$$

式 (1.33) により

$$(1.35) \quad \mathbf{A}_{lk} \mathbf{z}_k + \mathbf{L}_{ll} \mathbf{L}_{ll}^T \mathbf{z}_l = \mathbf{b}_l$$

を得るから左右の第2項を右辺へ移項して、

$$(1.36) \quad \mathbf{L}_{ll} \mathbf{L}_{ll}^T \mathbf{z}_l = \mathbf{b}_l - \mathbf{A}_{lk} \mathbf{z}_k$$

従って、式 (1.36) の $\mathbf{b}_l$ から $\mathbf{A}_{lk} \mathbf{z}_k$ を引いて $\mathbf{z}_l$ を求める。前進代入の最後には、 $\mathbf{L}_{pp} \mathbf{y}_p = \mathbf{b}_p$

を計算する。次に後退代入計算を行って $\mathbf{x}_k$ を求める。

先ず $\mathbf{L}_{pp}^T \mathbf{x}_p = \mathbf{y}_p$ を解く。

また $\mathbf{W} = (\mathbf{L}_{kk} \mathbf{L}_{kk}^T)^{-1} \mathbf{A}_{kl}$ だから

$$(1.37) \quad \begin{pmatrix} \mathbf{I} & \mathbf{W} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_l \end{pmatrix} = \begin{pmatrix} \mathbf{I} & (\mathbf{L}_{kk} \mathbf{L}_{kk}^T)^{-1} \mathbf{A}_{kl} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_l \end{pmatrix} = \begin{pmatrix} \mathbf{z}_k \\ \mathbf{z}_l \end{pmatrix}$$

となる。

式 (1.37) より

$$(1.38) \quad \mathbf{x}_k + (\mathbf{L}_{kk} \mathbf{L}_{kk}^T)^{-1} \mathbf{A}_{kl} \mathbf{x}_l = \mathbf{z}_k, \quad \mathbf{L}_{kk}^T \mathbf{x}_k + \mathbf{L}_{kk}^{-1} \mathbf{A}_{kl} \mathbf{x}_l = \mathbf{L}_{kk}^T \mathbf{z}_k$$

従って、 $\mathbf{v} = \mathbf{A}_{kl} \mathbf{x}_l$ を計算し、この $\mathbf{v}$ を使って $\mathbf{L}_{kk} \mathbf{u} = \mathbf{v}$ を解いて $\mathbf{u}$ を求める。

この $\mathbf{u}$ を式 (1.38) の第二式に代入し、 $\mathbf{L}_{kk}^T \mathbf{x}_k + \mathbf{u} = \mathbf{L}_{kk}^T \mathbf{z}_k$ つまり $\mathbf{L}_{kk}^T \mathbf{x}_k = \mathbf{L}_{kk}^T \mathbf{z}_k - \mathbf{u}$

$\mathbf{L}_{kk}^T \mathbf{z}_k$ は $\mathbf{y}_k$ に等しいから、結局 $\mathbf{x}_k$ は次式 (1.39) を解いて得ることになる。

$$(1.39) \quad \mathbf{L}_{kk}^T \mathbf{x}_k = \mathbf{y}_k - \mathbf{u}$$

構造物の計算方法として、この解法が着目されている。

(2) 乗積形逆行列法 (Product Form Inverse Matrix Method)

次数が大きくなり、メモリ容量が十分に確保できない場合、予備記憶媒体(ディスク)を使用する時、この解法が適している。

応用領域として、構造解析等の一部見いだされることがあるが、その目的が数値計算の教育用に適用された感じを受ける。外部記憶容量の適用等、数値計算の勉強として使われることがある。

乗積形逆行列法の計算ステップ

$\mathbf{A}$ を正則行列とする。 $\mathbf{A}$ の列ベクトル $\mathbf{a}_k$ を他のベクトル $\mathbf{a}$ で置き換えた行列を

$$\bar{\mathbf{A}} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}, \dots, \mathbf{a}_n)$$

とする時 $\mathbf{A}^{-1}$ から $\bar{\mathbf{A}}^{-1}$ を得る方法である。

先ず $\mathbf{A}\mathbf{y} = \mathbf{a}$ なるベクトル $\mathbf{y}$ で、単位行列 $\mathbf{I} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n)$ の第 k 列を置き換えた行列

$$\mathbf{E}_n = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{y}, \dots, \mathbf{e}_n)$$

を作る。次の関係式が成立する。

$$(1.40) \quad \mathbf{A}\mathbf{E}_k = \bar{\mathbf{A}}$$

従って

$$(1.41) \quad \bar{\mathbf{A}}^{-1} = \mathbf{E}_k^{-1} \mathbf{A}^{-1}$$

であり、 $\mathbf{E}_k^{-1}$ は

$$(1.42) \quad \mathbf{E}_k^{-1} = \begin{pmatrix} 1 & 0 & \cdot & -y_1/y_k & \cdot & 0 \\ 0 & 1 & \cdot & -y_2/y_k & \cdot & 0 \\ & & \cdot & \cdot & \cdot & \\ & & & 1/y_k & \cdot & \\ & & & \cdot & \cdot & \\ 0 & 0 & \cdot & -y_n/y_k & \cdot & 1 \end{pmatrix}$$

で与えられる。 y_i はベクトル $\mathbf{y}$ の第 i 要素である。

式 (1.1) の連立一次方程式 $\mathbf{A}\mathbf{x} = \mathbf{b}$ に対して、ここで $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$ とする。

さらに、いま

$$(1.43) \quad \begin{aligned} \mathbf{A}_0 &= (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n) = \mathbf{I} \\ \mathbf{A}_2 &= (\mathbf{a}_1, \mathbf{e}_2, \dots, \mathbf{e}_n) \\ \mathbf{A}_3 &= (\mathbf{a}_1, \mathbf{a}_2, \mathbf{e}_3, \dots, \mathbf{e}_n) \\ &\dots\dots\dots \\ \mathbf{A}_n &= (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = \mathbf{A} \end{aligned}$$

とすると

$$(1.44) \quad \begin{aligned} \mathbf{A}_1^{-1} &= \mathbf{E}_1^{-1} \mathbf{A}_0^{-1} = \mathbf{E}_1^{-1} & \mathbf{y}_1 &= \mathbf{A}_0^{-1} \mathbf{a}_1 = \mathbf{I} \mathbf{a}_1 = \mathbf{a}_1 \\ \mathbf{A}_2^{-1} &= \mathbf{E}_2^{-1} \mathbf{A}_1^{-1} & \mathbf{y}_2 &= \mathbf{A}_1^{-1} \mathbf{a}_2 \\ &\dots\dots\dots & \dots\dots\dots \\ \mathbf{A}_n^{-1} &= \mathbf{E}_n^{-1} \mathbf{A}_{n-1}^{-1} & \mathbf{y}_n &= \mathbf{A}_{n-1}^{-1} \mathbf{a}_n \end{aligned}$$

但し、 $\mathbf{E}_k = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{k-1}, \mathbf{y}_k, \mathbf{e}_{k+1}, \dots, \mathbf{e}_n)$ となるから、 $\mathbf{A}$ の逆行列は

$$(1.45) \quad \mathbf{A}^{-1} = \mathbf{E}_n^{-1} \mathbf{E}_{n-1}^{-1} \dots \mathbf{E}_1^{-1}$$

によって得られる様になる。連立一次方程式の解 $\mathbf{x}$ は

$$\begin{aligned}
 (1.46) \quad \mathbf{x} &= \mathbf{A}^{-1} \mathbf{b} \\
 &= \mathbf{E}_n^{-1} \mathbf{E}_{n-1}^{-1} \cdots \mathbf{E}_1^{-1} \mathbf{b}
 \end{aligned}$$

によって得られ、各 $\mathbf{E}_k^{-1}$ は

$$(1.47) \quad \mathbf{y}_k = \mathbf{E}_{k-1}^{-1} \mathbf{E}_{k-2}^{-1} \cdots \mathbf{E}_1^{-1} \mathbf{a}_k$$

から式 (1.42) で求めた形と同じ方法で作られる。

式 (1.46)、(1.47) の計算では、 $\mathbf{E}_k^{-1} \mathbf{a}$ の形の行列計算を繰り返し行う事になる。

この計算は $\mathbf{a}^T = (a_1, a_2, \dots, a_n)$, $\mathbf{y}_k^T = (y_{1k}, y_{2k}, \dots, y_{nk})$ としたとき、

$$\begin{aligned}
 (1.48) \quad a_i - \frac{y_{ik}}{y_{kk}} a_k & \quad (i \neq k) \\
 \frac{a_k}{y_{kk}} & \quad (i = k)
 \end{aligned}$$

を計算するだけで良く、 $\mathbf{E}_k^{-1}$ の記憶も

$$(1.49) \quad k, \left(-\frac{y_{1k}}{y_{kk}}, \dots, -\frac{y_{k-1,k}}{y_{kk}}, \frac{1}{y_{kk}}, -\frac{y_{k+1,k}}{y_{kk}}, \dots, -\frac{y_{nk}}{y_{yy}} \right)$$

$n+1$ 個の情報を記憶する。この方法では、計算途中で $n \times n$ 行列全体をコアにおく必要が無い。問題の大きさに比してコアメモリが十分ではない時、有効な方法である。

解法の過程における数値的安定性を保障するため

$$\left| \frac{y_{ik}}{y_{kk}} \right| \leq 1 \quad (i = k+1, k+2, \dots, n)$$

となるピボットを選ぶ様にする。

列ベクトルに従って変数の順序の入れ替えを行う。最初第 1 列 $\mathbf{a}_1$ を単位行列 $\mathbf{I}$ に導入する場合

$$\max_i |y_{i1}| = |y_{11}|$$

ならば、 $\mathbf{a}_1$ を $\mathbf{I}$ の第 1 列に入れて $\mathbf{E}_1$ を作る。すなわち

$$(1.50) \quad \mathbf{E}_1 = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{l-1}, \mathbf{y}_1, \mathbf{e}_{l+1}, \dots, \mathbf{e}_n)$$

$$\mathbf{e}_2 \text{ を導入する場合} \quad \max_{i \neq 1} |y_{i2}| = |y_{m2}|$$

によって第 1 列以外の列 m を定め

$$(1.51) \quad \mathbf{E}_2 = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{m-1}, \mathbf{y}_2, \mathbf{e}_{m+1}, \dots, \mathbf{e}_n)$$

以下、同様の手続きを行う事により、式 (1.46) 式 (1.47) の処理を実行する。その結果、変数の順序は入れ替わっているなので、順序を元に戻す為に、順序入れ替え情報を記憶する必要がある。

(3) 二列同時消去法 (2-Row Simultaneous Elimination Method)

Murata(村田)理論による。外積形式ガウス法では、1 回目の反復計算で 軸列 k により、残り小行列全体の消去計算が必要であり、内積形式ガウス法では、既存の η ベクトルによる第 k 列の更新計算が必要である。アルゴリズムにおいて、DO ループの判定回数の減少が、処理速度の向上につながる。ベクトル器においては、演算器の並列処理の向上につながる。従って、消去計算または更新計算で 2 列に対して並列計算するアルゴリズムが生きてくる。軸列も二列同時に選べば、更に処理速度の向上が期待できるが、アルゴリズムは複雑になる。

応用領域として、計算物理の研究者が興味を持ち、計算物理の研究として使用された事がある。それ以外は、教育研究用として適用された。理論は素晴らしいが、事例として適用するには、専門のコンサルタントが必要とされる。

二列同時消去法の計算ステップ

いま二軸列を並列消去することを考える。軸選択法として、行交換方式の部分軸選択法を使う。

行では $(r, r+1)$ 、列では $(r, r+1)$ 、 $(j, j+1)$ を選ぶ。

第 r 段階と第 $(r+1)$ 段階の連続消去は、次式になる。

$$\begin{aligned}
 (1.52) \quad a_{ij}^{(r)} &= \frac{a_{ij}^{(r-1)}}{a_{rr}^{(r-1)}} = -t_j^{(r)} \\
 a_{r+1,j}^{(r+1)} &= \frac{a_{r+1,j}^{(r)}}{a_{r+1,r+1}^{(r)}} = -t_j^{(r+1)} \\
 a_{ij}^{(r+1)} &= a_{ij}^{(r-1)} + t_j^{(r)} \cdot a_{ir}^{(r-1)} + t_j^{(r+1)} \cdot a_{i,r+1}^{(r)}
 \end{aligned}$$

但し、第 $(r+1)$ 段階の枢軸要素 $a_{rr}^{(r)}$ は軸選択後の値とする。第 3 式は 5 項演算子で、乗算と加算が各々 2 回行われており、乗算器と加算器のプロセッサには都合が良い。スカラプロセッサの場合においても、DO ループが半減するため、一定の速度向上が期待できる。更に第 j 列の他に第 $(j+1)$ 列についても、同時 2 列による連続消去を行うと、次式の形の計算方法をとる。

$$\begin{aligned}
 (1.53) \quad a_{r,j+1}^{(r)} &= \frac{a_{r,j+1}^{(r-1)}}{a_{rr}^{(r-1)}} = -t_{j+1}^{(r)} \\
 a_{r+1,j+1}^{(r+1)} &= \frac{a_{r+1,j+1}^{(r)}}{a_{r+1,r+1}^{(r)}} = -t_{j+1}^{(r+1)} \\
 a_{i,j+1}^{(r+1)} &= a_{i,j+1}^{(r-1)} + t_{j+1}^{(r)} \cdot a_{ir}^{(r-1)} + t_{j+1}^{(r+1)} \cdot a_{i,r+1}^{(r)}
 \end{aligned}$$

この場合も第 3 式は、5 項演算式で乗算と加算が各 2 回行われる。結局、乗算と加算が 4 回づつ並列処理でき、ベクトルプロセッサでは高速処理が可能になる。一般のスカラプロセッサにおいても DO ループが 1/4 に減るので、2.5 割程度の演算処理が少なくなる。これは内積形式ガウス法の応用であり、2 列の同時消去が出来る。

プログラムには次の 3 点が重要となる。

- (1) 第 k 段階と第 $(k+1)$ 段階の消去計算はスイッチの使用により、同一箇所を通る様にする
こと。
- (2) $t_j^{(r)}, t_{j+1}^{(r)}, t_j^{(r+1)}, t_{j+1}^{(r+1)}$ を配列にすること。

(3) 軸選択の為の行交換用作業領域を配列した軸選択は、行全体を一度に、一度にまとめて行うこと。

(4) 内積形式のガウス法を使用すること。

等である。

(4) ブロックスカイライン法(Block Skyline Method)

この解法は、対称疎行列に関する直接解法として提案された。方法としては、全体の対称行列の上三角部分を幾つかの列グループに分けて、各グループの列要素は、各列の非ゼロ要素から対角要素までで構成される。全体を列グループに分けるという点では、縦ブロックガウス法と同じ考えを利用している。しかし、縦ブロックガウス法は内積形式のガウス法であるのに対して、ブロックスカイライン法は縁取り法を使う事になる。

応用領域として、熱力学分野で使用された。構造解析のデータ処理の考えもある。一般には教育研究用として適用されている。

ブロックスカイライン法の解法への考え方

幾つかの列の集まりを1つのグループ (G) として、グループ 1 (G1)、グループ 2 (G2)、グループ (G3)、・・・とする。グループの個数を N 個とする。いま分解しようとしているあるグループを G_p とする。分解済みのグループ Q 内の第 i 列と G_p 内の第 j 列の内積を求めることで、G_p 内の(i,j)要素を決める。この時、第 i 列と、第 j 列に共通な行について、内積計算をする。

求める行列 **M** として、**M**(j) は各列の情報をしまっている境界線テーブルの第 j 列に対応する値で、グループ内の第 1 列の非ゼロ要素から第 j 列の対角要素までの累積要素数が入っている。

計算式

$$(1.54) \quad \begin{aligned} u_{i,j} &= a_{ij} - \sum_{N_i} u_{ki} \tilde{u}_{kj} = a_{ij} - \sum_{N_i} u_{ki} u_{ki} d_{ii} \\ d_{ii} &= a_{jj} - \sum_{j_h} u_{kj} \tilde{u}_{kj} \end{aligned}$$

式 (1.54) において、第 i 列と第 j 列の共通部分の長さは N_i であり、この部分をベクトルプロセッサ向きに内積計算をさせることになる。この2つの列の中にゼロ要素が多く含まれていても、ベクトルプロセッサに問題はない。スカラプロセッサの場合、ブロック行列ごとに分けて、小ブロックに対応した列同士の内積計算を行い、最後に、小ブロック同士の和を計算させて、最終の内積を計算することによる。

このブロックスカイライン法の場合は、ガウス法における二列同時二列消去の考えを利用し、並列演算器を並列処理させて性能を向上させる事が出来る。

この解法は、応用事例として、NASA が開発した NASTRAN は、この形の構造データを非ゼロ要素のみ詰めて格納している。

(5) 非対称疎行列の直接解法 (Non Symmetric Sparse Matrix Direct Method)

流体解析、回路解析等における大型の非対称疎行列では、幾度も同じ様な系を解く場合がある。こうした疎行列では、変数 n に対して、非ゼロ要素数 ÷ n 自乗の密度を sparsity と呼ぶ。この疎行列に対しては、ガウス法・ガウス・ジョルダン法など古典的な解法を内積形式にして、コア処理 (実記

憶装置内)で行う。

応用領域として、流体解析、回路解析分野で使用された事がある。大規模非対称疎行列では、教育的な適用が多い。一般には難しい面があり、事例として適用するには、専門のコンサルタントが必要とされる。

非対称疎行列の直接解法の計算ステップ

ガウス・ジョルダン法内積形式によると、

変換行列 $\mathbf{E}_k$ により、 $\mathbf{EA} = \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{I}$, $\mathbf{I}$ は単位行列

となる。

ここで、 $\mathbf{E}_k \mathbf{A}^{(k-1)} = \mathbf{A}^{(k)}$, $\mathbf{A}^{(0)} = \mathbf{A}$ とおくことで、

$$(1.55) \quad \mathbf{x} \mathbf{E}_k = \begin{pmatrix} 1 & & \eta_k^{(k)} & & \\ & \cdot & \cdot & & \mathbf{0} \\ & & 1 & \eta_{k-1k}^{(k)} & \\ & & \eta_{kk}^{(k)} & & \\ & & \eta_{k+1k}^{(k)} & 1 & \\ \mathbf{0} & & \cdot & & \cdot \\ & & \eta_{nk}^{(k)} & & 1 \end{pmatrix} \left\{ \begin{array}{l} \eta_{ik}^{(k)} = -\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} (i \neq k) \\ \eta_{kk}^{(k)} = \frac{1}{a_{kk}^{(k-1)}} \end{array} \right.$$

となる。連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ の解は

$$(1.56) \quad \mathbf{x} = \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{b}$$

によって求める事が出来る。式 (1.56) にみる 各 $\mathbf{E}_i$ の第 i 列のことを η ベクトルと呼び、この η ベクトルの非ゼロ要素だけを行番号として記憶する。

ガウス法内積形式では、変換行列 $\mathbf{E}_k$ により、

$$(1.57) \quad \mathbf{EA} = \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{U}, \quad \mathbf{L} = \mathbf{E}^{-1} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \cdots \mathbf{E}_n^{-1}$$

として LU 分解をする。

式 (1.57) にみる行列 $\mathbf{U}$ は、1 を対単要素とする上三角行列であり、行列 $\mathbf{L}$ と $\mathbf{E}_k$ は、下

三角行列である。連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ の両辺に $\mathbf{E}_n \cdots \mathbf{E}_1$ を乗じて

$$(1.58) \quad \mathbf{EAx} = \mathbf{Ux} = \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{b}$$

の形となる。

$$(1.59) \quad \mathbf{U} = \mathbf{U}_n \cdots \mathbf{U}_2 \mathbf{U}_1$$

式 (1.59) より

$$(1.60) \quad \mathbf{U}^{-1} \mathbf{Ux} = \mathbf{x} = \mathbf{U}^{-1} \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{b} = \mathbf{U}_1^{-1} \cdots \mathbf{U}_n^{-1} \mathbf{E}_n \cdots \mathbf{E}_2 \mathbf{E}_1 \mathbf{b}$$

解 $\mathbf{x}$ はベクトル $\mathbf{b}$ の左辺から $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_n$ 及び $\mathbf{U}_n^{-1}, \dots, \mathbf{U}_2^{-1} \mathbf{U}_1^{-1}$ をこの順に乘じる。

内積形式では行列 $\mathbf{U}_i^{-1}$ は行列 $\mathbf{E}_i$ の作成時に求める事が出来るので、 $\mathbf{E}_i$ と $\mathbf{U}_i^{-1}$ は続けて格納する必要がある。

$\mathbf{E}_i$ を $\mathbf{E}_i^{(L)}$ 、 $\mathbf{U}_i^{-1}$ を $\mathbf{E}_i^{(U)}$ と書けば、 $\mathbf{E}_1^{(U)}\mathbf{E}_2^{(U)}\cdots\mathbf{E}_n^{(U)}\mathbf{E}_n^{(L)}\cdots\mathbf{E}_2^{(L)}\mathbf{E}_1^{(L)}\mathbf{A}=\mathbf{I}$ となる。

これらに対して、格納エリアを使って fill-in で計算することによる。

1・6．反復法、計算のプロセス

もとの行列をそのまま保存し、初期値から出発して繰り返し反復計算を行う。各計算ステップに関して、この時の計算値がある一定の範囲内に入れば収束したものとみなす。反復法は大規模な疎行列に適している。係数行列の持つ性質によって収束の速さが変わる。

式 (1・1) より $\mathbf{Ax} = \mathbf{b}$

これと同値な形を次に変形する。

$$(1\cdot61) \quad \mathbf{x} = \mathbf{Mx} + \mathbf{c}$$

式 (1・61) の行列 $\mathbf{M}$ には $\mathbf{A}$ が持っている特徴 (例えば疎) をそのまま保存する形である。

$$(1\cdot62) \quad \mathbf{x}_{k+1} = \mathbf{Mx}_k + \mathbf{c}$$

なる反復代入の計算システムを構成させる。ここで $k=0,1,2,\dots,n-1$

初期値 $\mathbf{x}_0$ から出発して、 $\mathbf{x}_1, \mathbf{x}_2, \dots$ を計算して値 $\mathbf{x}_n$ が一つ前の値 $\mathbf{x}_{n-1}$ との差が一定の値 (誤差 $\varepsilon = 10^{-6}$ 等) 以下に達すると収束したと見なす方法である。この行列 $\mathbf{M}$ を反復行列と言う。 $\mathbf{x}_k$ が $\mathbf{x}$ に収束する為には反復行列 $\mathbf{M}$ の固有値の絶対値がすべて 1 より小さくなくてはならない。つまり $\mathbf{M}$ のスペクトル半径を $\rho(\mathbf{M})$ とするとき、 $\rho(\mathbf{M}) < 1$ である。反復法では一般に初期値 $\mathbf{x}^{(0)}$ を $\mathbf{0}$ と置く。次に時代に分類して、古典的解法である反復法の基本から述べる。

1・6・1．古典的解法

19 世紀のガウスと同時代に活躍したヤコビ法に代表される解法である。元の行列の性質を保存したままある形式に分離して、元の行列の対角行列を活かした形で、反復行列を定義する。特徴として疎な行列であれば、この疎という形がそのまま活かされた形を持つ。ヤコビはフィボナッチの反復理論を基礎にした、無限小の繰り返しの考えを入れたことで、その後に数値解法が大きく発展して行く礎になる。ヤコビ法を発展させ、数学の一樣収束の考えを展開させて、反復の繰り返しに対して、よりスムーズに緩やかに収束させる考えを提案したザイディルは、これをガウス・ザイディル法と名付けた。このガウス・ザイディル法も古典的範疇に入れる事が出来る。現代の多くの反復法は、この反復法の祖であるヤコビ法の考えの上に立っている。

(1) ヤコビ法 (Jacobi Method)

連立一次方程式の解法において、ガウスの消去法の提案に対して、ヤコビは別な角度からの研究を行った。ヤコビは数学史上 14 世紀以前から考えられてきた無限小の概念を研究し、これを更に発展させ、連続的反復事象が極限迄続く無限小による収束理論を行列に対して試みた。そして行列反復に

対する初期値の考え方を確立させ、行列における初めての反復解法を提唱した(1837)。ヤコビは反復理論としての土台を固め、一方では関数行列式などを始め、数多くの優れた業績を残した。このヤコビ法に見る反復法としての考え方は、後世における反復理論を大きく飛躍させる原動力となっている。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算、1950年代から1960年代にかけて、そして1970年頃までの構造解析、流体解析、基礎物理等の分野に多く見受けられる。反復法を代表する解法として、反復を理解する上においても、重要な位置づけにある。

ヤコビ法の計算ステップ

$$(1.63) \quad \mathbf{x}_{k+1} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\mathbf{x}_k + \mathbf{D}^{-1}\mathbf{b}$$

但し、 $\mathbf{A}$ の対角行列を $\mathbf{D}$ 、狭義の左下三角行列を $\mathbf{E}$ 、右上三角行列を $\mathbf{F}$ とする。

$$\mathbf{D} = \begin{pmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & a_{33} & \\ & & & \ddots \\ 0 & & & & a_{n,n} \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} 0 & & & 0 \\ a_{21} & 0 & & \\ a_{31} & a_{32} & 0 & \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & a_{n,n-1} & 0 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdot & a_{1n} \\ & 0 & a_{23} & \cdot & a_{2n} \\ & & 0 & \cdot & \cdot \\ & & & \ddots & a_{n-1,n} \\ 0 & & & & 0 \end{pmatrix}$$

初期値 $\mathbf{x}_0 = [0 \ 0 \ \cdot \ \cdot \ 0]^T$, 収束判定 $\varepsilon = 10^{-6}$

ヤコビ反復法は

$$(1.64) \quad \mathbf{A} = \mathbf{D} + \mathbf{E} + \mathbf{F}$$

の形に分離して、反復行列 $\mathbf{M}$ は次の形となる。

$$(1.65) \quad \mathbf{M} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})$$

式 (1.65) の $(\mathbf{E} + \mathbf{F})$ は $\mathbf{A}$ の非対角成分からなる行列であるから、 $\mathbf{A}$ が疎行列であれば、 $\mathbf{M}$ は疎である性質が保存される事になる。

$n \times n$ 行列 $\mathbf{A}$ の固有値を λ とする。ゲシュゴリンの定理から

$$(1.66) \quad |a_{ii} - \lambda| \leq \sum_{j \neq i} |a_{ij}|$$

を満たす i が必ず1つ存在する。

固有ベクトル $\mathbf{x}$ として、絶対値最大のものを $\mathbf{x}_i$ とする。第 i 行では

$$(1.67) \quad a_{ii} - \lambda = -\sum_{j \neq i} a_{ij} \frac{x_j}{x_i}$$

$n \times n$ 行列 $\mathbf{A}$ が対角優位 (diagonal dominant) であれば、その成分が

$$(1.68) \quad |a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, n$$

を満たすことを意味する。

式 (1.65) にみる反復行列 $\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})$ の第 ij 成分を m_{ij} とすると、

$$(1.69) \quad m_{ij} = 0, \quad m_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad i \neq j$$

式 (1.65) の $\mathbf{M}$ の固有値を λ に対して

$$(1.70) \quad |\lambda - m_{ii}| = |\lambda| \leq \sum_{j \neq i} |m_{ij}| = \frac{\sum_{j \neq i} |a_{ij}|}{|a_{ii}|}$$

を満たす i が存在する。

$\mathbf{A}$ が対角優位であると、任意の i に対して、 $\sum_{j \neq i} |a_{ij}| < |a_{ii}|$ が成り立ち、従って

$|\lambda| < 1$ となり、収束する。このヤコビ法を改良したものが、次のガウス・ザイデル法である。

(2) ガウス・ザイデル法 (Gauss-Seidel Method)

ザイデル(Seidel)はヤコビ法を研究してゆくうち、近似解へのステップが固定されている点に気づいた。一方で数学の一樣収束の考えを展開させ、繰り返し反復に対して、よりスムーズに緩やかに収束させる考えを提案した(1840)。ヤコビ法の長所そのものは活かしながら、各計算ステップでの変数の近似修正を計算毎、逐次に進めるヤコビ法の改良型の方法である。ザイデルの提案は、ガウスの考えを基礎とし、ヤコビ反復理論の長所を取り入れた所から、ガウス・ザイデル法と呼ばれている。この考えは計算解の精密性に定評がある。19 世紀前半の問題 (古典物理) を検討する上で、もとの行列の性質が崩れない反復の考えが数値計算を普及させた。

応用領域として、ヤコビ法とほぼ同じ傾向があり、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算、1950 年代から 1970 年代頃までの構造解析、流体解析、基礎物理等の分野に多く見受けられる。ヤコビ法を発展させた理論として、広く普及した。次に述べる SOR 法がパラメータの考えを主力にしているのに対して、数値計算を十分に得意としない工学研究者に理解しやすい解法である。ヤコビ法の発展という意味からも、反復法を代表する解法として、これも重要な位置づけにある。

ガウス・ザイデル法の計算ステップ

$$(1.71) \quad \mathbf{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{x}_{k+1} - \mathbf{F}\mathbf{x}_k)$$

式 (1.71) を変形すると

$$(1.72) \quad \mathbf{x}_{k+1} = -(\mathbf{D} + \mathbf{E})^{-1}\mathbf{F}\mathbf{x}_k + (\mathbf{D} + \mathbf{E})^{-1}\mathbf{b}$$

普通は式 (1.72) が用いられる。

式 (1.72) は反復行列 $\mathbf{M} = -(\mathbf{D} + \mathbf{E})^{-1}\mathbf{F}$ の形となる。

つまり $\mathbf{A}$ が疎であれば、その性質が反復解法に保存される。

$\mathbf{A}$ が対称正定値行列であれば、

$$\mathbf{A}^T = \mathbf{A}$$

正定値であると、任意のベクトル $\mathbf{y} \neq \mathbf{0}$, $\mathbf{y}^T \mathbf{A} \mathbf{y} > 0$ が成立する。

$\mathbf{A}$ が対称正定値行列であれば、必ず収束することが以下の式から説明出来る。

$$\begin{aligned} \mathbf{D}^T &= \mathbf{D} \\ \mathbf{E} &= \mathbf{F}^T \end{aligned} \quad \text{より}$$

$$(1.73) \quad \mathbf{A} = \mathbf{D} + \mathbf{F}^T + \mathbf{F}$$

が成立する。

式 (1.73) の反復行列は $\mathbf{M} = -(\mathbf{D} + \mathbf{F}^T)^{-1} \mathbf{F}$ となる。

この $\mathbf{M}$ の固有値を λ 固有ベクトルを $\mathbf{x}$ とする。

$$(1.74) \quad -(\mathbf{D} + \mathbf{F}^T)^{-1} \mathbf{F} \mathbf{x} = \lambda \mathbf{x}$$

すなわち

$$(1.75) \quad \mathbf{F} \mathbf{x} = -\lambda (\mathbf{D} + \mathbf{F}^T) \mathbf{x}$$

式 (1.73) より

$$(1.76) \quad \mathbf{A} \mathbf{x} = (\mathbf{D} + \mathbf{F} + \mathbf{F}^T) \mathbf{x} = (1 - \lambda) (\mathbf{D} + \mathbf{F}^T) \mathbf{x}$$

$$(1.77) \quad (\mathbf{D} + \mathbf{F}^T) \mathbf{x} = \frac{1}{1 - \lambda} \mathbf{A} \mathbf{x}$$

式 (1.75) より

$$(1.78) \quad \mathbf{F} \mathbf{x} = -\frac{\lambda}{1 - \lambda} \mathbf{A} \mathbf{x}$$

従って、式 (1.78) の転置をとって

$$(1.79) \quad \mathbf{x}^T \mathbf{F}^T \mathbf{x} = -\frac{\lambda}{1 - \lambda} \mathbf{x}^T \mathbf{A} \mathbf{x}$$

式 (1.77) の左から $\mathbf{x}^T$ を乗じたものを辺々引くと

$$(1.80) \quad \mathbf{x}^T \mathbf{D} \mathbf{x} = \left(\frac{1}{1 - \lambda} + \frac{\lambda}{1 - \lambda} \right) \mathbf{x}^T \mathbf{A} \mathbf{x} = \frac{1 - \lambda^2}{(1 - \lambda)^2} \mathbf{x}^T \mathbf{A} \mathbf{x}$$

となる。 $\mathbf{A}$ と $\mathbf{D}$ は共に正定値であるから、 $\frac{1 - \lambda^2}{(1 - \lambda)^2} > 0$ すなわち

$$\lambda^2 < 1$$

が導かれる。従ってガウス・ザイデル法は、収束することが証明出来る。

1-6-2. 近代的解法

古典的なガウス・ザイデル法を少しでも速く収束させる形として、パラメータを取り入れた新しい解法 SOR 法が提案された。この SOR 法は後に少し改良した形として、SLOR 法等に見受けられる幾つもの枝葉の解法が次々提案される。同じ時代に直降下の提案とした SD 法に対して、空間内を直交探索する解法として、CG 法が提案された。この CG 法は対称行列に限定されるが、その後少し改良した形として非対称も可能な解法が提案されるようになる。この時代に、交互方向の反復として ADI 法が提案され、準反復の考えとしてチェビシェフ反復法が提案される。一方 Stone 法は、LU 分

解の特徴を活かすことで、解のステップを進める考えとして提案された。分野別の対応として、構造解析に適した部分構造反復法等も提案された。

(1) SOR 法 (Successive Over Relaxation Method) 加速(過)緩和法

1950 年 Young と Frankel はガウス・ザイデル法に加速パラメータ ω を加味し漸近収束率を高め、次の近似解への修正量を拡大し、厳密解に少しでも高速収束を目指した SOR 法を提唱した。この解法は、その後いろいろな分野に対する研究対象として改良され、枝葉となる解法がいろいろ研究され、SLOR 法、SSOR 法、USSOR 法、SBOR 法、Point Block SOR 法、2-Line SOR 法、Odd-EvenSOR 法他があり全体として SOR 法系を構成している。陽解法としての SOR 法に対して、SLOR 法は陰解法に属する。これはマトリックスが対称、非対称にかかわらず適用できる長所があり、更に固有値分布の悪化傾向があったり、条件数が大きくても安定的に比較的高速性を保って確実に収束して行く事が出来る。巡回解法にも適用が出来る。

応用領域として、数値計算の教育分野を始めとして、産業界全般、1960 年代から 2000 年代の流体解析、原子力・核融合分野、地球科学、海洋科学、宇宙科学、原子核、天文学、電磁界、時間積分を伴ったデバイスシミュレーション、その他の計算物理や更に資源探索系や軍事問題に迄広く見受けられる。最近はいろいろな解法が提案されているが、この SOR 法は行列の形に依存しない為に、あらゆる分野へ適用出来る。プログラムも簡単で多くは公表されており、21 世紀の現在においても計算物理学者に根強い人気がある。

SOR 法の計算ステップ

代表的な SOR 法は、次の形となる。

行列 $\mathbf{A}$ を対角要素からなる対角行列 $\mathbf{D}$ 、狭義の下三角行列 $\mathbf{E}$ 、上三角行列 $\mathbf{F}$ として分割する。

$$(1.81) \quad \mathbf{A} = \mathbf{D} + \mathbf{E} + \mathbf{F}$$

この算法は

$$(1.82) \quad \mathbf{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{x}_{k+1} - \mathbf{F}\mathbf{x}_k)$$

$$(1.83) \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \omega(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

式 (1.82) 式 (1.83) より SOR 法反復行列 $\mathbf{M}$ は、次式で表現出来る。

$$(1.84) \quad \mathbf{M} = (\mathbf{I} + \omega \mathbf{D}^{-1}\mathbf{E})^{-1}((1 - \omega)\mathbf{I} - \omega \mathbf{D}^{-1}\mathbf{F})$$

式 (1.84) にみる形の反復計算になる。この反復法が収束するための条件は、反復行列 $\mathbf{M}$ のスペクトル半径 (spectral radius) $\rho(\mathbf{M})$ が 1 より小さいことになる。

つまり、収束する場合とは $\mathbf{M}$ の固有値を λ とすると、すべての固有値 λ_i が $|\lambda_i| < 1$ を満たすことによる。

SOR 法の一般形

$$(1.85) \quad \mathbf{x}_{k+1} = (\mathbf{I} + \omega \mathbf{D}^{-1}\mathbf{E})^{-1}((1 - \omega)\mathbf{I} - \omega \mathbf{D}^{-1}\mathbf{F})\mathbf{x}_k + \omega (\mathbf{D} + \omega \mathbf{E})^{-1}\mathbf{b}$$

加速パラメータ ω は $0 < \omega < 2$ である。

式 (1・84) の反復行列 $\mathbf{M}$ の固有値 λ 固有ベクトル $\mathbf{x}$ から

$$(1\cdot86) \quad ((1-\omega)\mathbf{D}-\omega\mathbf{F})\mathbf{x}=\lambda(\mathbf{D}+\omega\mathbf{F}^T)\mathbf{x}$$

これは

$$(1\cdot87) \quad (1-\lambda)(\mathbf{D}+\omega\mathbf{F}^T)\mathbf{x}=\omega\mathbf{Ax}$$

または

$$(1\cdot88) \quad (1-\lambda)((1-\omega)\mathbf{D}-\omega\mathbf{F})\mathbf{x}=\lambda\omega\mathbf{Ax}$$

の形となる。

これから

$$(1\cdot89) \quad (1-\lambda)(2-\omega)\mathbf{x}^T\mathbf{Dx}=(1+\lambda)\omega\mathbf{x}^T\mathbf{Ax}$$

式 (1・89) の両辺に $(1-\lambda)$ を乗じて、

$$(1\cdot90) \quad (1-\lambda)^2(2-\omega)\mathbf{x}^T\mathbf{Dx}=(1-\lambda^2)\omega\mathbf{x}^T\mathbf{Ax}$$

が得られ $\lambda^2 < 1$ となって、SOR 法は収束する事が結論される。この時、 加速パラメータ ω は、通常 $1 < \omega$ 形をとり、行列の性質により、最適な値は少し変化する。

(2) CG 法 (Conjugate Gradient Method) 共役勾配法

航空機における工学研究の中から大次元連立一次方程式の高速解法への要求があり、Hestenes と Stiefel により CG 法として提案(1952)されて普及した。これは n 次元空間を考え一次独立なベクトルを定義し、ある係数を求め空間内を直交性を保って探索する事により、 n 次元の行列は n 回の反復で収束する考えである。正定値対称行列に対して適用が出来る。

応用領域として、教育用の数値計算、教育用の一般研究、産業界、1960 年代から 1990 年代までの流体解析、構造解析、原子力問題、基礎物理等がある。CG 法は行列が対称である条件が伴うので、非対称問題では、そのままでは適用できない点がある。CG 法は既述した、不完全を伴うコレスキー法や LU 分解法と組み合わせたり、前処理法(Preconditioning Method)と言う名のもとに各種の CG 法として研究され現在広く普及している。大規模疎行列においてはその高速解法の特長から、ベクトル計算機向けとして多く研究されている。前処理法として、代表的な ICCG 法他がある。

CG 法の計算ステップ

式 (1・1) において、もし $\mathbf{A}$ が非対称であるならば、式 (1・1) の両辺に $\mathbf{A}$ の転置行列 $\mathbf{A}^T$ を乗じて

$$\mathbf{A}^T\mathbf{Ax}=\mathbf{A}^T\mathbf{b} \quad \text{つまり} \quad \mathbf{A}'\mathbf{x}=\mathbf{b}' \quad \text{として対称形に導く。}$$

(正定値：固有値がすべて正であること)

式 (1・1) の解を $\mathbf{z} (= \mathbf{A}^{-1}\mathbf{b})$ とする。つぎの関数を定義する。

$$(1\cdot91) \quad f(\mathbf{x})=(\mathbf{x}-\mathbf{z},\mathbf{A}(\mathbf{x}-\mathbf{z}))$$

式 (1・91) において $\mathbf{A}$ は正定値であるから、 $\mathbf{x} \neq \mathbf{z}$ の時 $(\mathbf{x} - \mathbf{z})^T \mathbf{A}(\mathbf{x} - \mathbf{z}) > 0$

つまり $f(\mathbf{x}) > 0$, $\mathbf{x} = \mathbf{z}$ の時 $f(\mathbf{x}) = 0$, $f(\mathbf{x})$ は $\mathbf{x}$ が $\mathbf{z}$ からかけ離れている度合いを示す。

適当な初期値 $\mathbf{x}_0$ から出発して、 $\mathbf{x}_1, \mathbf{x}_2, \dots$ を辿って解 $\mathbf{x} = \mathbf{z}$ に到達する。

ここで $\mathbf{x}_k$ から $\mathbf{x}_{k+1}$ は次式で定める。

$$(1 \cdot 92) \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{P}_k$$

n 次元空間でそれぞれ直交し、式 (1・92) は $\mathbf{x}_k$ からどれくらい離れた点を $\mathbf{x}_{k+1}$ とするかを示す。 α_k は変化係数であり、 $\mathbf{P}_k$ は変化への方向ベクトルを表す。

CG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 , \quad \mathbf{p}_0 = \mathbf{r}_0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = (\mathbf{p}_k, \mathbf{r}_k) / (\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$$

$$\beta_k = -(\mathbf{r}_{k+1}, \mathbf{A}\mathbf{p}_k) / (\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

continue

(3) ADI 法 (Alternating-Direction Implicit Iterative Method) 交互方向法

Peaceman と Rachford は反復法の長所である高速解法の中に、直接法の長所である厳密解法の組み込み研究を行い、行方向求解、列方向求解による交互方向の解法を提案した(1955)。これが ADI 法であり、交互方向法とも言われる。疎な対称行列である場合ハウスホルダー変換(Householder Transform)により容易に三重対角行列 (Tridiagonal Matrix) に導く事が出来る。大型な三重対角行列では、計算の高速性への著しい効果が見られる。

応用領域として、計算物理や計算化学を中心に、多くの適用例が見受けられる。核融合シミュレーションにも、この解法が良く使用されている。現在前処理法を中心とした解法に多くの研究がされ幾つも提案されているが、この ADI 法は物理学者が数学的に理解し易い点があり、21 世紀においても、計算物理の研究者に根強い人気がある。

ADI 法の計算ステップ

行列 $\mathbf{A}$ がスティルチス行列 (Stieltjes Matrix $\mathbf{A}$ は実対称正定符号、非対角要素は非正の行列) であれば $\mathbf{A}$ を次の 3 つの行列に分ける事が出来る。

$$(1.93) \quad \mathbf{A} = \mathbf{D} + \mathbf{H} + \mathbf{V}$$

但し、 $\mathbf{D}$: 非負対角行列

$\mathbf{H}, \mathbf{V}$: 正の対角要素と非正の非対角要素を持つ優対角行列

である。

式 (1.1) を次の二つの式へと導く。

$$(1.94) \quad \begin{aligned} (\mathbf{H} + \frac{1}{2}\mathbf{D} + \omega\mathbf{I})\mathbf{x} &= (\omega\mathbf{I} - \mathbf{V} - \frac{1}{2}\mathbf{D})\mathbf{x} + \mathbf{b} \\ (\mathbf{V} + \frac{1}{2}\mathbf{D} + \omega\mathbf{I})\mathbf{x} &= (\omega\mathbf{I} - \mathbf{H} - \frac{1}{2}\mathbf{D})\mathbf{x} + \mathbf{b} \end{aligned}$$

但し、 ω は加速パラメータ

$$(1.95) \quad \mathbf{H}_1 = \mathbf{H} + \frac{1}{2}\mathbf{D}, \quad \mathbf{V}_1 = \mathbf{V} + \frac{1}{2}\mathbf{D}$$

とにおいて、

式 (1.94) は次の形で定義される。

$$(1.96) \quad \begin{aligned} (\mathbf{H}_1 + \omega_{i+1}\mathbf{I})\mathbf{x}^{i+1/2} &= (\omega_{i+1}\mathbf{I} - \mathbf{V}_1)\mathbf{x}^i + \mathbf{b} \\ (\mathbf{V}_1 + \omega_{i+1}\mathbf{I})\mathbf{x}^{i+1} &= (\omega_{i+1}\mathbf{I} - \mathbf{H}_1)\mathbf{x}^{i+1/2} + \mathbf{b} \end{aligned}$$

式 (1.96) は陰反復解法になる。ここで式 (1.96) の左辺はガウス消去法に基づき、第一の方程式は水平格子線に沿って解き、第二の方程式は垂直格子線に疎って解く事から交互方向陰反復解法と称される。行列の性質にもよるが、特に 5 線対角のブロック行列である場合、この解法が大変適しており、更にモデルが単位正方形の形をしている場合、特に計算効率を上げる事が分かっている。一般的な直接法(ガウス・ジョルダン法等)で解かす場合に比べて、相当高速に解ける事が良く知られている。

(4) チェビシェフ反復法 (Chebyshev Semi-Iterative Method) 準反復法

Chebyshev は多項式とその最小化問題を研究してゆくうち、反復法の加速パラメータを一つの定義に従って変化させる事により、平均収束率がより高くなっている事に気づき、チェビシェフ反復として提案した(1960)。チェビシェフはチェザロの総和法の研究から考えを得て、数列の総和に関する式を立てて、反復として併用する特徴があり、準反復法とも呼ばれている。一般的なモデルでは SOR 法より高速収束する事が知られている。高速求解の考え方として、準反復法としての位置づけにある。

応用領域として、計算物理や計算化学を中心に、多くの適用例が見受けられる。特に 1970 年代頃から炉物理の研究者に愛用されて、多くの原子力コードに適用された。21 世紀の現在迄も米国原子力分野では、この解法は 40 年以上も愛用されている。物理学者はこの解法のステップが、逐次的な意味を持ち、数値計算として魅力を感じる傾向にある。特に高速求解として、原子力コード中の中性子の外挿法等では著しい効果が見られ、この分野で適用例が多く見られる。

チェビシェフ反復法の計算ステップ

$\mathbf{M}$ は $n \times n$ エルミット収束行列 (Elmeat Convergent Matrix

$$\mathbf{M}^* = \mathbf{M}^T, \quad \mathbf{M}^* \text{ は } \mathbf{M} \text{ の複素共役転置行列) とする。}$$

$\mathbf{M}$ の固有値は実数でその絶対値はすべてスペクトル半径より小さい。

$\mathbf{M}$ のスペクトル半径 $\rho(\mathbf{M}) < 1$ であるとする。

チェビシェフ反復の基本的考え方

一般的な反復法を次に定義する。

$$(1.97) \quad \mathbf{x}^{(m+1)} = \mathbf{M}\mathbf{x}^{(m)} + \mathbf{g}, \quad (m \geq 1)$$

式 (1.97) に対して、数列の総和に関する類似として数列の反復を次に定義する。

$$(1.98) \quad \mathbf{y}^{(m)} = \sum_{k=0}^m v_k(m) \mathbf{x}^{(k)}, \quad (m \geq 0)$$

式 (1.98) を併用させて解く。式 (1.97) の反復ベクトルを式 (1.98) で代数的に結合させる為、準反復法と呼ばれる。

チェビシェフ反復法は次の形をとる。

$$(1.99) \quad \mathbf{y}^{(m+1)} = \omega_{m+1}(\mathbf{M}\mathbf{y}^{(m)} + \mathbf{g} - \mathbf{y}^{(m-1)}) + \mathbf{y}^{(m-1)}, \quad m \geq 0$$

加速パラメータ ω_{m+1} は

$$(1.100) \quad \omega_{m+1} = 1 + \frac{C_{m-1}(1/\rho)}{C_{m+1}(1/\rho)}, \quad m \geq 1, \quad \omega_1 = 1$$

である。ここで $C_m(x)$ は、次式により定義される。

$$(1.101) \quad C_m(x) = \begin{cases} \cos(m \cos^{-1} x) & , \quad -1 \leq x \leq 1, \quad m \geq 0 \\ \cosh(m \cosh^{-1} x) & , \quad x \geq 1, \quad m \geq 0 \end{cases}$$

この多項式概念の導入は最小化問題への求解として、効率良く推進させることが出来る。SOR 法が加速パラメータを一定の値として固定させているが、このチェビシェフ反復法は加速を変化させる事が出来、最小化への単調減少の加速を目指したものであり、この方法で平均収束率をより高くさせる事が出来る。

(5) 部分構造反復法 (Partial Structural Iterative Method)

構造力学の数値計算の発展として、この部分構造反復法が考え出された。応用領域として、バンド行列を作る特徴のある構造力学に、適用事例が幾つも見つかる。大規模行列の数値計算という意味で、この解法は今後も活用されると考えられる。

部分構造反復法の計算ステップ

一般式 $\mathbf{A}_0 \mathbf{x} = \mathbf{B} \mathbf{x} + \mathbf{b}$ から得られる反復式 $\mathbf{A}_0 \mathbf{x}^{(i)} = \mathbf{B} \mathbf{x}^{(i-1)} + \mathbf{b}$ で計算する。

$\mathbf{A}_0$ の LU 分解は一度で済む。 $\mathbf{x}^{(i)}$ と真の解 $\mathbf{x}$ との差は

$$\mathbf{x} - \mathbf{x}^{(i)} = \mathbf{A}_0^{-1} \mathbf{B} (\mathbf{x} - \mathbf{x}^{(i-1)}) = \cdots = (\mathbf{A}_0^{-1} \mathbf{B})^{(i)} (\mathbf{x} - \mathbf{x}_0)$$

となるから $\mathbf{x} - \mathbf{x}^{(i)}$ のノルムは $(\mathbf{A}_0^{-1}\mathbf{B})^{(i)}$ のノルムによって抑えられる。

$$\|\mathbf{x} - \mathbf{x}^{(i)}\| \leq \|\mathbf{A}_0^{-1}\mathbf{B}\|^{(i)} \cdot \|\mathbf{x} - \mathbf{x}_0\|$$

この準陰解法が収束するためには $\|\mathbf{A}_0^{-1}\mathbf{B}\| < 1$ が必要であり、 $\|\mathbf{A}_0^{-1}\mathbf{B}\| \ll 1$ であれば、急速に収束する。構造物 p 個の部分構造物と、部分構造物境界部分に分ける。

行列 $\mathbf{A}$ を $p+1$ 個の対角ブロックに分けて $\mathbf{A}\mathbf{x} = \mathbf{b}$ を次の形に変形して扱う。

$$(1.102) \quad \begin{aligned} \mathbf{A}_k \mathbf{x}_k + \mathbf{H}_k \mathbf{x}_k &= \mathbf{b}_k \quad (k=1, \dots, p) \\ \sum_{k=1}^p \mathbf{G}_k \mathbf{x}_k + \mathbf{C} \mathbf{x}_c &= \mathbf{b}_c \end{aligned}$$

つまり

$$(1.103) \quad \begin{aligned} \mathbf{A}_k \mathbf{x}_k^{(i)} &= \mathbf{b}_k - \mathbf{H}_k \mathbf{x}_k^{(i-1)} \quad (k=1, \dots, p) \\ \mathbf{C} \mathbf{x}_c^{(i)} &= \mathbf{b}_c - \sum_{k=1}^p \mathbf{G}_k \mathbf{x}_k^{(i)} \end{aligned}$$

とする。

部分構造物境界上のベクトル $\mathbf{x}_c$ の初期値 $\mathbf{x}_c^{(0)}$ を与えて $\mathbf{x}_k^{(1)}, \mathbf{x}_c^{(1)}, \mathbf{x}_k^{(2)}, \dots$ を順に計算する。

$p=2$ の時、境界 C を $\mathbf{A}_1$ に含めて番号を付けると $\mathbf{A}\mathbf{x} = \mathbf{b}$ は次の形になる。

$$(1.104) \quad \begin{aligned} \mathbf{A}_1 \mathbf{x}_1^{(i)} &= \mathbf{b}_1 - \mathbf{H}_1 \mathbf{x}_2^{(i-1)} \\ \mathbf{A}_2 \mathbf{x}_2^{(i)} &= \mathbf{b}_2 - \mathbf{G}_1 \mathbf{x}_1^{(i)} \end{aligned}$$

となる。従って第二の部分構造物上で境界 C に接する接点群 Γ の初期値 $\mathbf{x}_0^{(0)}$ をもとに $\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \mathbf{x}_1^{(2)}, \dots$ と求めて行く事が出来る。大規模な構造物では、 Γ 上の接点の初期値を粗い網目による近似値から内挿して得た値にすれば良い。

(6) SLOR 法 (Successive Line Over Relaxation Method) 加速線(過)緩和法

Young と Frankel が提案した SOR 法に対して、後世の研究者が幾つもの SOR 法の改良型を提案した。それらを SOR 法系として 1 つにくくる事が出来るが、その中の 1 つが SLOR 法である。計算方法は古典的な、ガウス・ザイディルの考えを拡張させている。堅実な計算ステップである。これは線反復法とも言われる。

応用領域として、SOR 法の応用と同じく、計算物理の分野に幾つもの適用されている。特に 1960 年代から、炉物理の研究者に重用された。この計算方法が SOR 法の発展という意味から普及したものと考えられる。主に 1960 年代から 2000 年代の炉物理の数値解法に良く見受けられる。本論文の研究テーマである原子炉内における中性子拡散方程式の数値解法も、そのオリジナルコードは米国オークリッジの原子力研究所で研究開発されたが、そこでも主力の数値解法はこの SLOR 法を用いている。本論ではこの解法に対して、前処理をした CG 法系に変更した研究結果を詳細に論述している。

SLOR 法の計算ステップ

$\mathbf{T}$ を三重対角行列とする時、行列 $\mathbf{A}$ を次の形に分離する。

$$(1.105) \quad \mathbf{A} = \mathbf{T} + \mathbf{E} + \mathbf{F}$$

式 (1.105) で $\mathbf{E}$ は狭義の下三角行列であり、 $\mathbf{F}$ は上三角行列である。

式 (1.1) の連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ に対して

$$(1.106) \quad (\mathbf{T} + \mathbf{E})\mathbf{x} = -\mathbf{Fx} + \mathbf{b}$$

$$(1.107) \quad \mathbf{x}_k = -(\mathbf{T} + \mathbf{E})^{-1} \mathbf{Fx}_{k-1} + (\mathbf{T} + \mathbf{E})^{-1} \mathbf{b}$$

式 (1.107) を線反復法と言う。ここで加速係数 ω を導入して、

$$(1.108) \quad \mathbf{A} = \frac{1}{\omega}(\mathbf{T} + \omega\mathbf{E}) + \frac{1}{\omega}[(\omega-1)\mathbf{T} + \omega\mathbf{F}]$$

として

$$(1.109) \quad \|(\mathbf{T} + \omega\mathbf{E})^{-1}[(\omega-1)\mathbf{T} + \omega\mathbf{F}]\|$$

を出来るだけ小さくする ω を選び、

$$(1.110) \quad \mathbf{x}_k = -(\mathbf{T} + \omega\mathbf{E})^{-1}[(\omega-1)\mathbf{T} + \omega\mathbf{F}]\mathbf{x}_{k-1} + \omega(\mathbf{T} + \omega\mathbf{E})^{-1} \mathbf{b}$$

の形としたものが SLOR 法である。

$$(1.111) \quad \begin{aligned} \mathbf{x}_k &= (1-\omega)\mathbf{x}_{k-1} + \omega\mathbf{D}^{-1}(-\mathbf{Lx}_k - \mathbf{Rx}_{k-1} + \mathbf{b}) \\ (1+\omega\mathbf{D}^{-1}\mathbf{L})\mathbf{x}_k &= [(1-\omega) - \omega\mathbf{D}^{-1}\mathbf{R}]\mathbf{x}_{k-1} + \omega\mathbf{D}^{-1}\mathbf{b} \\ (\mathbf{D} + \omega\mathbf{L})\mathbf{x}_k &= [(1-\omega)\mathbf{D} - \omega\mathbf{R}]\mathbf{x}_{k-1} + \omega\mathbf{b} \end{aligned}$$

つまり

$$(1.112) \quad \mathbf{x}_k = -(\mathbf{D} + \omega\mathbf{L})^{-1}[(\omega-1)\mathbf{D} + \omega\mathbf{R}]\mathbf{x}_{k-1} + \omega(\mathbf{D} + \omega\mathbf{L})^{-1} \mathbf{b}$$

となる

反復法の収束条件 $\mathbf{Ax} = \mathbf{b}$ を $\mathbf{x} = \mathbf{Sx} + \mathbf{G}$ の形に変形すると、

$$(1.113) \quad \mathbf{x}_k - \mathbf{x} = \mathbf{S}(\mathbf{x}_{k-1} - \mathbf{x}) = \cdots = \mathbf{S}^k(\mathbf{x}^{(0)} - \mathbf{x})$$

を得る。式 (1.113) から $\mathbf{x}_k$ が真の解 $\mathbf{x}$ に収束する為には、 $\mathbf{S}^k$ のノルムが 0 に収束すれば良

い。一方、行列 $\mathbf{S}$ の最大固有値 μ とすれば、 $\mathbf{S}^k$ の最大固有値は μ^k となるから

$$(1.114) \quad \|\mathbf{S}^k\| = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{S}^k \mathbf{x}\|}{\|\mathbf{x}\|} = |\mu|^k$$

となる。従って $\|\mathbf{S}^k\|$ が 0 に収束するには、 $|\mu| < 1$ であれば良い。

(7) Stone 法 (Strongly Implicit Method)

1950 年代の数値計算の応用として広がった SOR 法や ADI 法は、行列を分離分解することによって、数値の変化を一定の収束の領域へ向わせるものであり、確実なステップの流れであるが、繰り返

しの為の多くの回数が必要とされるものであった。1968 年に発表された Stone 法は、LU 分解の特徴を活かすことで、解のステップを進めるものである。不完全 LU 分解と残差を分けた形として表現される。

応用領域として、1970 年代の石油掘削問題の数値解法、鉱脈予測問題の数値解法等[47]にその事例が見つかっている。1980 年代以降はこの解法が見られる事はなく、下火になったと考えられる。

Stone 法の計算ステップ

行列を不完全な形の LU 分解とし、その不完全 LU 分解の $\mathbf{L}$ と $\mathbf{U}$ をそれぞれ $\tilde{\mathbf{L}}$ と $\tilde{\mathbf{U}}$ で表し

$\mathbf{A} = \tilde{\mathbf{L}}\tilde{\mathbf{U}} - \mathbf{R}$ として、次の反復式

$$(1.115) \quad \tilde{\mathbf{L}}\tilde{\mathbf{U}} \mathbf{x}_k = \mathbf{R} \mathbf{x}_{k-1} + \mathbf{b}$$

とにおいて解くものである。

これは CG 法等を基本に据えて、それらを適用する前に方程式に前処理を施した形

$$(1.116) \quad (\tilde{\mathbf{L}}\tilde{\mathbf{U}})^{-1} \mathbf{A} \mathbf{x} = (\tilde{\mathbf{L}}\tilde{\mathbf{U}})^{-1} \mathbf{b}$$

として展開出来る。

最近の新しい 1 つの流派である Meijerink 流は、この考え方の上に立って新しい理論を構成している。この Stone 法の考えが、後に直交探索としての CG 法系の前処理の原動力となった。この解法は、最近では工学系の一部領域、石油の採掘や、地盤掘削などに適用事例が見受けられる。

(8) SD 法 (Steepest Descent Method)

ガウス消去法と同じ時代に、コーシーにより方程式の残差をゼロにする考えから提案された。

最急降下法と呼ばれる。近年ブースが反復法としてこの改良を行い、現在の SD 法は反復法としての位置づけにある。後に社会に広く普及した CG 法より理論面が簡単である。これは極値問題を追求する中でこの解法が考えられた。残差の 2 乗を最小にする考えから出ている。代数的表現をしたとき勾配を最も急な方向に向けて進めて行く方法になる。

応用領域として、1950 年代から 1960 年代までの熱力学問題等に事例が見つかっている。

計算のステップ

この方法が適用出来る為には、その関数 $f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ の偏微分係数 $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots$ が簡単に

計算できる必要がある。その基礎となる考えでは、残差の 2 乗を最小にすることにある。 $\mathbf{r}$ を残差とすると

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}, \quad f(\mathbf{x}) = (\mathbf{r}, \mathbf{r}) \rightarrow \min$$

残差 $\mathbf{r}$ が 0 となって、 $f(\mathbf{x})$ となれば、その時の $\mathbf{x}$ が解となる。つまり $f(\mathbf{x})$ が最小になる $\mathbf{x}$ を見つけることになる。

SD 法の考え方

最初は第一近似解 $\mathbf{x}_0$ を与える。これを用いて残差を計算する。

$$(1.117) \quad \mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$$

$\mathbf{A}$ の転置行列 $\mathbf{A}^T$ を掛けることで最急勾配方向

$$(1.118) \quad \mathbf{p}_k = \mathbf{A}^T \mathbf{r}_k$$

が求められる。最適な α を次式で計算

$$(1.119) \quad \alpha_k = \frac{(\mathbf{r}_k, \mathbf{A}\mathbf{p}_k)}{(\mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}$$

式 (1.119) の $\alpha_k, \mathbf{p}_k$ をもとに第 $k+1$ 近似値を次式で計算、収束するまで、を繰り返す。

$$(1.120) \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

SD 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$$

$$\mathbf{p}_k = \mathbf{A}^T \mathbf{r}_k$$

$$\alpha_k = \frac{(\mathbf{r}_k, \mathbf{A}\mathbf{p}_k)}{(\mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

continue

1-6-3. 現代的解法

数値計算として、空間直交のステップに入る前に、元の行列を少し変形して CG 法に繋ぐと、解法が速いことが発見された。これは行列をコレスキーの分解法とか、LU 分解法による完全な形の分解ではなく、不完全な形の分解に留めておき、行列をその変形した形で CG 法に繋ぐ等して、より高速な数値解法へ導く事にある。これは行列の前処理と呼ばれる呼ばれる様になり、新しい解法が考え出された。この前処理をする事により、最大固有値と最小固有値の比率、つまり条件数が 1 に近づき数値解法が容易になり、高速で計算出来る事を意味している。行列が大きくなるに従って、疎な性質が高くなり、前処理をした反復法は、疎行列の性質を有効に活用する事が出来る。計算機の発達と共に、大型行列の高速計算を目指して、新しい前処理の数値解法が幾つも提案されている。これは前処理を如何に効率よく作り上げ、CG 法系統に繋ぐかによっている。更に前処理を後残差最小の面から CR 法系統に繋ぐ方法も多く提案されている。これは条件数などによる行列の性質により、CG 法よりも高速な求解が期待出来るものもある。この節では、代表的な解法を紹介したい。

(1) ICCG 法 (Incomplete Cholesky Conjugate Gradient Method)

Meijerink と Van der Vorst は CG 法を研究して行く中で、行列分解との相乗効果に関心を寄せて

いた。コレスキーの死後、後の時代コレスキー研究家が提案した方法、対角行列を組み入れた修正コレスキー分解(Modified Cholesky decomposition)と称する新しい分解方法に関して、Meijerink と Van der Vorst は着目し、この分解過程で行列積の算法について研究した。彼らは数学の集合論の立場から、行列の非零要素のみの格子集合を定義し、これを計算対象とする考えで数値計算を行った[23]。これは不完全コレスキー分解(Incomplete Cholesky decomposition)と称して、この計算の後CG法を適用する事で、大規模対称疎行列はより高速に計算出来る考えを ICCG 法と名付けて提案した(1977)。これは不完全コレスキー分解により、求める行列の最大固有値と最小固有値の比が縮まり、もとの行列の固有値が重複することで、その重複する分だけ反復回数が少なくなり、数値解法が容易になる事を意味している[102]。最近ではこの考えを前処理法(Preconditioning Method)として、数値計算研究者の間で広く普及するようになった。

応用領域として、前処理法の一般教育、一般的な研究用を始めとして、構造解析や流体解析、基礎物理、基礎化学問題、更に NASA Ames リサーチセンターの宇宙科学としての大次元問題、Hanford 技術開発研究所等における原子核エネルギー問題の計算物理等に見受けられる。特に 1990 年代以降においては、地球科学、海洋科学、電気電子、電磁界、素粒子、原子核、天文学、更に資源探索系や軍事の分野、最近では生命科学、分子科学分野でも多く用いられている。前処理法の代表的存在として、多くの研究者の注目を引きつけた解法である。

計算のステップ

項番(4) の CG 法の計算ステップの手前で前処理法としての計算展開を行う。

式 (1.1) において、行列 $\mathbf{A}$ の不完全コレスキー分解を行う。

$\mathbf{U}$: 上三角行列、 $\mathbf{U}^T$: 上三角行列の転置行列、 $\mathbf{D}$: 対角行列 とすると

$$\mathbf{A} \approx \mathbf{U}^T \mathbf{D} \mathbf{U} \quad \text{で表現される。}$$

式 (1.1) について次の形にする。

$$(1.121) \quad (\mathbf{D}^{1/2} \mathbf{U})^{-T} \mathbf{A} (\mathbf{D}^{1/2} \mathbf{U})^{-1} \mathbf{x}' = \mathbf{b}'$$

式 (1.121) において $\mathbf{x}' = (\mathbf{D}^{1/2} \mathbf{U}) \mathbf{x}$ 、 $\mathbf{b}' = (\mathbf{D}^{1/2} \mathbf{U})^{-T} \mathbf{b}$

ICCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0 , \quad \mathbf{p}_0 = (\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1} \mathbf{r}_0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = \frac{(\mathbf{r}_k, (\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1} \mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A} \mathbf{p}_k)}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\beta_k = \frac{(\mathbf{r}_{k+1}, (\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1} \mathbf{r}_{k+1})}{(\mathbf{r}_k, (\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1} \mathbf{r}_k)}$$

$$\mathbf{p}_{k+1} = (\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1} \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

continue

収束判定 定数 $\varepsilon > 0$ として、 $\sqrt{(\mathbf{r}_k, \mathbf{r}_k)} < \varepsilon$ で収束。この時の $\mathbf{x}_k$ が数値解となる。

(2) PCG 法 (Preconditioned Conjugate Gradient Method)

n 次元の行列に対して、CG 法系が残差の直交生に着目して、残差多項式を定義して、方向ベクトルと k 次の多項式で表現している。空間を探索して、次のステップへ進むにおいて、もし固有値が重複 (q 個) していると残差の動きうる空間の、次元が q 個落ちることになる。従ってこの場合、 $n - q$ 個の反復で収束する。つまり固有値が密集すると空間の次元がそれだけ落ちることになる。このことがある行列を事前に掛けたりすることで、空間の次元を落として行く方法が考えられる。

応用領域として、前処理法の教育用としてのニーズが高く、数値計算の一般教育、一般的な研究用を始めとして、一部の基礎物理分野に使用されているのが見受けられる。後に前処理法の代表として位置づけられ、次の形として展開できる。

計算のステップ

適切な正則行列 $\mathbf{C}$ を定義する。

式 (1.1) の連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ に対して、 $\mathbf{C}^{-1}\mathbf{Ax} = \mathbf{C}^{-1}\mathbf{b}$ とする。

この時 $\mathbf{C} = \mathbf{SS}^T$ と分解出来る行列を選ぶ。

方程式を

$$(1.122) \quad (\mathbf{S}^{-1}\mathbf{AS}^{-T})(\mathbf{S}^T\mathbf{x}) = \mathbf{S}^{-1}\mathbf{b}$$

とおく。ここで $\mathbf{S}^{-T} = (\mathbf{S}^T)^{-1} = (\mathbf{S}^{-1})^T$ である。

$\mathbf{S}^T$ を未知ベクトルとすると、行列 $\mathbf{S}^{-1}\mathbf{AS}^{-T}$ は対称正定値行列となり、CG 法が適用できる。

この $\mathbf{C}$ の条件として

- (1) $\mathbf{A}$ に近い
- (2) $\mathbf{C} = \mathbf{SS}^T$ となる $\mathbf{S}$ が容易に計算できる。
- (3) $\mathbf{S}$ が $\mathbf{A}$ と同程度に疎である。
- (4) $\mathbf{C}^{-1}\mathbf{r}$ が容易に計算できる。

などが必要となる。

これに対して $\mathbf{S}^{-1}\mathbf{AS}^{-T}$ は単位行列近くになり、その固有値は 1 に近くに密集する。そのため、CG 法の収束は速くなる。

PCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 , \quad \mathbf{p}_0 = \mathbf{C}^{-1}\mathbf{r}_0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = \frac{(\mathbf{C}^{-1}\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$$

$$\beta_k = \frac{(\mathbf{C}^{-1}\mathbf{r}_{k+1}, \mathbf{r}_{k+1})}{(\mathbf{C}^{-1}\mathbf{r}_k, \mathbf{r}_k)}$$

$$\mathbf{p}_{k+1} = \mathbf{C}^{-1}\mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

continue

この PCG 法の後に、ICCG 法を初めとして、多くの前処理をした CG 法系への発展へと向かって行く。

(3) BCG 法 (Biconjugate Gradient Method)

CG 法は対称正定値行列に対する解法になっている。行列が非対称であっても、CG 法系として解法を目指すべきとして発展した。行列が非対称であるならば、次の 3 点が考えられる。規則非対称問題に対する接近方法として、CG 法系を用いての非対称行列への展開は、

- (1) 双対な方程式を定義して、もとの行列と合わせて(拡大した)対称形に導き求解する方法
- (2) もとの行列の自乗形式で対称形に導き求解する方法
- (3) もとの行列の転置行列を(左から)掛けて、対称形に導き求解する方法

等が考えられる。

このうちアルゴリズムとして一般に普及しているものが(1)と(2)であり、(1)は BCG 法、(2)は CGS 法等である。(3)の方法によれば、CG 法等の適用を目指して行うが、条件数が比較的少ない行列(概ね 10^2 以内)に於いては安定解が求められる。但し、この方法によると、数式モデルにより時間が多くかかる場合がある。そこで (1)と(2)で、その解法をクリロフ部分空間の点から高速安定解法を目指した解法となる。

応用領域として、非対称行列に対する前処理法の教育用としてのニーズがあり、数値計算の一般教育、一般的な研究用を始めとして、基礎物理分野に使用されており、特に非対称問題となる原子力問題や一部の素粒子ゲージ問題等があり、時代ともに非対称行列と限らない分野においても広く普及するようになった。名前だけが先行し、研究者に受け入れ易い解法である。それらは生命科学、電気電子、電磁界、原子核、天文学等に適用されているのが見受けられる。

計算のステップ

先ずは、BCG 法としては以下になる。

式 (1.1) の連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ に対して、双対な方程式

$$(1.123) \quad \mathbf{A}^T \mathbf{x}^* = \mathbf{b}^*$$

を組み合わせた $2n$ 元連立一次方程式を定義して計算する。 $2n \times 2n$ 行列と $2n$ 次ベクトル

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^T \end{bmatrix}, \quad \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \mathbf{b}^* \end{bmatrix} \quad \text{として、}$$

式 (1.1) 及び式 (1.123) から、次式が導かれる。

$$(1.124) \quad \tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

の形となる。BCG 法は、具体的には次のステップとなる。

BCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0, \quad \mathbf{p}_0 = \mathbf{r}_0 = \mathbf{p}_0^* = \mathbf{r}_0^*$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / (\mathbf{Ap}_k, \mathbf{p}_k^*)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{Ap}_k, \quad \mathbf{r}_{k+1}^* = \mathbf{r}_k^* - \alpha_k \mathbf{A}^T \mathbf{p}_k^*$$

$$\beta_k = (\mathbf{r}_{k+1}, \mathbf{r}_{k+1}^*) / (\mathbf{r}_k, \mathbf{r}_k^*)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k, \quad \mathbf{p}_{k+1}^* = \mathbf{r}_{k+1}^* + \beta_k \mathbf{p}_k^*$$

continue

この BCG 法は安定した数値解法として用いられている。

(4) PBCG 法(Preconditioned Biconjugate Gradient Method)

BCG 法に対する標準的な前処理の形として発展した。BCG 法に対する標準的な前処理法としての位置づけにある。応用領域として、BCG 法の前処理法の教育用数値計算を始めとして、一部の基礎物理分野に使用されているのが、見受けられる。

計算のステップ

式 (1.1) の連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ に対して、適切な正則行列 $\mathbf{C}$ を定義して

$$(1.125) \quad \mathbf{C}^{-1} \mathbf{Ax} = \mathbf{C}^{-1} \mathbf{b}$$

とする。

PBCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r} = \mathbf{r}^* = \mathbf{C}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}) \quad , \quad \mathbf{p} = \mathbf{p}^* = \mathbf{r}$$

$$\mu_1 = (\mathbf{r}, \mathbf{r}^*)$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , (k=0,1,2,...)

$$\mathbf{q}_k = \mathbf{C}^{-1}\mathbf{A}\mathbf{p}_k$$

$$\alpha_k = \mu_1 / (\mathbf{q}_k, \mathbf{p}_k^*)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{q}_k$$

$$\mathbf{q}_k^* = (\mathbf{C}^{-1}\mathbf{A})^T \mathbf{p}_k^*$$

$$\mathbf{r}_{k+1}^* = \mathbf{r}_k^* - \alpha_k \mathbf{q}_k^*$$

$$\mu_2 = \mu_1$$

$$\mu_1 = (\mathbf{r}_k, \mathbf{r}_k^*)$$

$$\beta = \mu_1 / \mu_2$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta \mathbf{p}_k \quad , \quad \mathbf{p}_{k+1}^* = \mathbf{r}_{k+1}^* + \beta \mathbf{p}_k^*$$

continue

(5) ILUBCG 法 (Incomplete LU Biconjugate Gradient Method)

非対称問題に適する BCG 法に、前処理として代表的な不完全 LU 分解を、前処理法として付けたものである。

応用領域として、前処理法の教育用としてのニーズが高く、数値計算の一般教育、一般的な研究用を始めとして、計算物理、計算化学、構造解析、流体解析、地球科学、海洋科学、電気電子、素粒子格子ゲージ問題、原子核、天文学、更に生命科学分野に至るまで広い範囲で使用されている事が見受けられる。BCG 法の前処理の代表的方法として位置づけられ、ILU の広い知名度から一気に広がった。1980 年代後半から徐々に浸透し、2000 年代以降は色々な分野に適用されて行く。テキストも幾つか出ており、広く使われている。

計算のステップ

行列 $\mathbf{A}$ の不完全分解 $\mathbf{A} = \mathbf{LU} + \mathbf{R}$ を経た後に BCG 法を適用する。 $\tilde{\mathbf{L}}\tilde{\mathbf{U}}$ を不完全な LU 分解とする。

ILUBCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = (\tilde{\mathbf{L}} \tilde{\mathbf{U}})^{-1} (\mathbf{b} - \mathbf{A} \mathbf{x}_0) \quad , \quad \mathbf{p}_0 = \mathbf{r}_0 = \mathbf{p}_0^* = \mathbf{r}_0^*$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do $\quad , \quad (k=0,1,2,\dots)$

$$\alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / \left((\tilde{\mathbf{L}} \tilde{\mathbf{U}})^{-1} \mathbf{A} \mathbf{p}_k, \mathbf{p}_k^* \right)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k (\tilde{\mathbf{L}} \tilde{\mathbf{U}})^{-1} \mathbf{A} \mathbf{p}_k, \quad \mathbf{r}_{k+1}^* = \mathbf{r}_k^* - \alpha_k \left((\tilde{\mathbf{L}} \tilde{\mathbf{U}})^{-1} \mathbf{A} \right)^T \mathbf{p}_k^*$$

$$\beta_k = (\mathbf{r}_{k+1}, \mathbf{r}_{k+1}^*) / (\mathbf{r}_k, \mathbf{r}_k^*)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k, \quad \mathbf{p}_{k+1}^* = \mathbf{r}_{k+1}^* + \beta_k \mathbf{p}_k^*$$

continue

(6) CGS 法 (Conjugate Gradient Squared Method)

BCG 法の項で記載した項番(2)番目「もとの行列の自乗形式で対称形に導き求解する方法」に該当する。大規模疎行列に対する高速計算として提案された。自乗形式の CG 法系として、優れた解析手法であると考えられる。

応用領域として、自乗形式の CG 法系として数値計算の一般教育、一般的な研究用を始めとして流体解析を始めとして、特に 2000 年代においては、地球科学、海洋科学、天文学、生命科学、素粒子格子ゲージ問題等にも広い範囲で使用されている事が見受けられる。テキストも幾つか出ており、広く使われている。

CGS 法アルゴリズム

初期ベクトル $\hat{\mathbf{x}}_0$, 収束判定値 eps

$$\hat{\mathbf{r}}_0 = \mathbf{b} - \mathbf{A} \hat{\mathbf{x}}_0, \quad \hat{\mathbf{p}}_0 = \mathbf{e}_0 = \hat{\mathbf{r}}_0 \quad \text{とする。}$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do $\quad , \quad (k=0,1,2,\dots)$

$$\alpha_k = (\hat{\mathbf{r}}_0, \mathbf{r}_k) / (\hat{\mathbf{r}}_0, \mathbf{A} \hat{\mathbf{p}}_k)$$

$$\mathbf{h}_{k+1} = \mathbf{e}_k - \alpha_k \mathbf{A} \hat{\mathbf{p}}_k$$

$$\hat{\mathbf{r}}_{k+1} = \hat{\mathbf{r}}_k - \alpha_k \mathbf{A} (\mathbf{e}_k + \mathbf{h}_{k+1})$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \alpha_k (\mathbf{e}_k + \mathbf{h}_{k+1})$$

$$\beta_k = (\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_{k+1}) / (\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_k)$$

$$\mathbf{e}_k = \hat{\mathbf{r}}_{k+1} + \beta_k \mathbf{h}_{k+1}$$

$$\hat{\mathbf{p}}_{k+1} = \mathbf{e}_{k+1} + \beta_k(\mathbf{h}_{k+1} + \beta_k \hat{\mathbf{p}}_k)$$

continue

この CGS 法は高速な解法として期待されている。

(7) PCGS 法 (Preconditioned Conjugate Gradient Squared Method)

CGS 法に対する標準的な前処理形を伴った解析手法である。前処理を $\mathbf{C}$ として $\mathbf{A}$ を不完全 $\mathbf{LDU}$ 分解して $\mathbf{A} = \mathbf{LDU} + \mathbf{R}$ とした時の $\mathbf{LDU}$ を用いる。

応用領域として、CGS 法の前処理法の教育用数値計算を始めとして、教育研究用、一部の基礎物理分野に使用されているのが見受けられる。

PCGS 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{C}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}) , \quad \mathbf{p} = \mathbf{e} = \mathbf{r} = \mathbf{r}_0$$

$$\mu_1 = (\mathbf{r}_0, \mathbf{r})$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{q}_k = \mathbf{C}^{-1}\mathbf{A}\mathbf{p}_k$$

$$\alpha_k = \mu_1 / (\mathbf{r}_0, \mathbf{q}_k)$$

$$\mathbf{h}_{k+1} = \mathbf{e}_k - \alpha_k \mathbf{q}_{k+1}$$

$$\mathbf{e}_{k+1} = \mathbf{e}_k + \mathbf{h}_{k+1}$$

$$\mathbf{q}_k = \mathbf{C}^{-1}\mathbf{A}\mathbf{e}_{k+1}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{e}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{q}_k$$

$$\mu_2 = \mu_1$$

$$\mu_1 = (\mathbf{r}_0, \mathbf{r}_k)$$

$$\beta_k = \mu_1 / \mu_2$$

$$\mathbf{e}_k = \mathbf{r}_{k+1} + \beta_k \mathbf{h}_{k+1}$$

$$\mathbf{p}_{k+1} = \mathbf{e}_{k+1} + \beta_k(\mathbf{h}_{k+1} + \beta_k \mathbf{p}_k)$$

continue

(8) CR 法 (Conjugate Residual Method)

CR 法は CG 法の考えから、残差に着目して、残差の内積をゼロ化する形から考えられた。条件にもよるが標準的な問題であれば、CG 法より速く解法する事がある。

応用領域として、残差問題に対する教育用数値計算を始めとして、教育研究用、1990 年代以降の一部の基礎物理分野、熱力学問題や拡散問題等の分野に使用されているのが見受けられる。

CR 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \mathbf{p}_0 = \mathbf{r}_0, \quad \mathbf{q}_0 = \mathbf{A}\mathbf{p}_0, \quad i = 0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha = (\mathbf{r}_k, \mathbf{A}\mathbf{p}_k) / (\mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha \mathbf{A}\mathbf{p}_k$$

$$\beta = -(\mathbf{A}\mathbf{r}_{k+1}, \mathbf{A}\mathbf{p}_k) / (\mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta \mathbf{p}_k \quad (k = k+1)$$

continue

(9) ILUCR 法 (Incomplete LU Conjugate Residual Method)

CR 法に対する不完全 LU 分解の前処理の形である。非対称行列 $\mathbf{A}$ を不完全 LU 分解したあと $\mathbf{A} = \mathbf{LU} + \mathbf{N}$ 共役残差法を適用する。

応用領域として、残差問題に対する教育用数値計算を始めとして、更に CR 法の前処理の研究として、不完全 ILU はニーズが高く教育研究用、計算物理、計算化学の分野、1990 年代以降の構造解析、流体解析、2000 年代以降の地球科学、海洋科学、電気電子、素粒子格子ゲージ問題、更に生命科学分野に至るまで広い範囲で使用されている事が見受けられる。前処理 ILU の知名度と共に、残差の考えを採り入れた CR 法として理解が広がった。CR 法は CG 法との相違を理解されないで、使用している研究者が多い。テキストも幾つか出ており、広く使われている。

計算のステップ

計算方法として、式 (1・1) の $\mathbf{Ax} = \mathbf{b}$ の両辺に $(\mathbf{LU})^{-1}$ を乗じて

$$(1 \cdot 126) \quad (\mathbf{LU})^{-1} \mathbf{Ax} = (\mathbf{LU})^{-1} \mathbf{b}$$

の形とする。式 (1・126) の $(\mathbf{LU})^{-1} \mathbf{A}$ は単位行列に近くなっており、共役残差法の収束が速くなることが予想される。

ILUCR 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = (\mathbf{LU})^{-1}(\mathbf{b} - \mathbf{Ax}_0) , \quad \mathbf{p}_0 = \mathbf{r}_0$$

$$\mathbf{x}_0 \text{ 初期ベクトル} , \quad \mathbf{q}_0 = (\mathbf{LU})^{-1} \mathbf{Ar}_0 , \quad \mathbf{p}_k^{(1)} = \mathbf{p}_k , \quad \mathbf{q}_k^{(1)} = \mathbf{q}_k$$

while $\|\mathbf{r}_k\| > \text{eps} * \|(\mathbf{LU})^{-1} \mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mu_k = (\mathbf{q}_k, \mathbf{q}_k)$$

$$\alpha_k = (\mathbf{r}_k, \mathbf{q}_k) / \mu_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{q}_k$$

$$\tilde{\mathbf{r}}_{k+1} = (\mathbf{LU})^{-1} \mathbf{Ar}_{k+1}$$

$$\beta_k = (\tilde{\mathbf{r}}_{k+1}, \mathbf{q}_k) / \mu_k$$

$$\beta_k^{(1)} = (\tilde{\mathbf{r}}_{k+1}, \mathbf{q}_k^{(1)}) / \mu_k^{(1)} , \quad \beta_1^{(1)} = 0$$

$$\mathbf{p}_{k+1} = \tilde{\mathbf{r}}_{k+1} - \beta_k^{(1)} \mathbf{p}_k^{(1)} - \beta_k \mathbf{p}_k$$

$$\mathbf{q}_{k+1} = \tilde{\mathbf{r}}_{k+1} - \beta_k^{(1)} \mathbf{q}_k^{(1)} - \beta_k \mathbf{q}_k$$

$$\mu_k^{(1)} = \mu_k$$

continue

この CR 法は $\mathbf{A}$ と $\mathbf{A}^T$ の平均値が、正定値であれば収束が速い。

(10) Bi-CGSTAB 法 (Bi-Conjugate Gradient Stabilized Method)

Bi-CG 法と STAB 法の組み合わせの考えによる。それは 2 つのパラメータのうち、 ς_k は残差 $\mathbf{r}_k$ のノルムを最小にする様にして、残りの η_k は、常に $\eta_k = 0$ とする積型反復法を持ち込む[28]。

応用領域として、Bi-CGSTAB 法としての教育用数値計算を始めとして、教育研究用である。未だ十分に普及している訳ではないが、2000 年代以降の一部の基礎物理分野、計算化学分野等に使用されているのが見受けられる。どの様な分野へ適用させるのが良いのか、現在専門家により研究されている[30]。

計算のステップ

1 つのパラメータ $\eta_k = 0$ とおいたことから添え字を k として表現している。

$$\mathbf{u}_k = \varsigma_k \mathbf{Ap}_k , \quad \mathbf{z}_k = \varsigma_k \mathbf{t}_k$$

と表す事が出来る。積型反復法から最も簡素な形をした Bi-CGSTAB 法が得られる[103]。

Bi-CGSTAB 法アルゴリズム

初期ベクトル $\mathbf{r}_0^*$ 、収束判定値 eps 、 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$

$$(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0, \quad \mathbf{r}_0^* = \mathbf{r}_0, \quad \beta_{-1} = 0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1} (\mathbf{p}_{k-1} - \varsigma_{k-1} \mathbf{A} \mathbf{p}_{k-1})$$

$$\alpha_k = (\mathbf{r}_0^*, \mathbf{r}_k) / (\mathbf{r}_0^*, \mathbf{A} \mathbf{p}_k)$$

$$\mathbf{t}_k = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\varsigma_k = (\mathbf{A} \mathbf{t}_k, \mathbf{t}_k) / (\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k + \varsigma_k \mathbf{t}_k$$

$$\mathbf{r}_{k+1} = \mathbf{t}_k - \varsigma_k \mathbf{A} \mathbf{t}_k$$

$$\beta_k = (\alpha_k / \varsigma_k) \cdot (\mathbf{r}_0^*, \mathbf{r}_{k+1}) / (\mathbf{r}_0^*, \mathbf{r}_k)$$

continue

(11) Bi-CGSTAB2 法 (Bi-Conjugate Gradient Stabilized2 Method)

Bi-CGSTAB 法の発展として考えられた。2つのパラメータのうち、 ς_k と η_k の決め方には、CGS 法と Bi-CGSTAB 法、更に GPBiCG 法の各々のパラメータを反復毎に交互に使う事が考えられる。Gutknecht による Bi-CGSTAB2 法は、偶数回の反復で Bi-CGSTAB 法のパラメータを使い、奇数回の反復で GPBiCG 法のパラメータを使う解法である。つまり GPBiCG 法の偶数回のパラメータ η_k を強制的に $\eta_k = 0$ と置くことで Bi-CGSTAB2 法が得られる。

応用領域として、Bi-CGSTAB 法とほぼ同じく、教育用数値計算を始めとして、教育研究用である。未だ十分に普及している訳ではないが、2000 年代以降の一部の基礎物理分野、計算化学分野等に使用されているのが見受けられる。Bi-CGSTAB 法と同じく、この解法もどのような分野へ適用させるのが良いのか、現在専門家により研究されている。

Bi-CGSTAB2 法のアルゴリズム

初期ベクトル $\mathbf{r}_0^*$ 、収束判定値 eps 、 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$

$$(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0, \quad \mathbf{r}_0^* = \mathbf{r}_0, \quad \beta_{-1} = 0, \quad \mathbf{t}_{-1} = 0, \quad \mathbf{w}_{-1} = 0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1} (\mathbf{p}_{k-1} - \mathbf{u}_{k-1})$$

$$\alpha_k = (\mathbf{r}_0^*, \mathbf{r}_k) / (\mathbf{r}_0^*, \mathbf{A} \mathbf{p}_k)$$

$$\mathbf{y}_k = \mathbf{t}_{k-1} - \mathbf{r}_k - \alpha_k \mathbf{w}_{k-1} + \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\mathbf{t}_k = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

if $n=2k$ then

$$\varsigma_k = (\mathbf{A} \mathbf{t}_k, \mathbf{t}_k) / (\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)$$

$$\eta_k = 0$$

else

$$\varsigma_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}$$

$$\eta_k = \frac{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}$$

endif

$$\mathbf{u}_k = \varsigma_k \mathbf{A} \mathbf{p}_k + \eta_k (\mathbf{t}_{k-1} - \mathbf{r}_k + \beta_{k-1} \mathbf{u}_{k-1})$$

$$\mathbf{z}_k = \varsigma_k \mathbf{r}_k + \eta_k \mathbf{z}_{k-1} - \alpha_k \mathbf{u}_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k + \mathbf{z}_k$$

$$\mathbf{r}_{k+1} = \mathbf{t}_k - \eta_k \mathbf{y}_k - \varsigma_k \mathbf{A} \mathbf{t}_k$$

$$\beta_k = (\alpha_k / \varsigma_k) \cdot (\mathbf{r}_0^*, \mathbf{r}_{k+1}) / (\mathbf{r}_0^*, \mathbf{r}_k)$$

$$\mathbf{w}_k = \mathbf{A} \mathbf{t}_k + \beta_k \mathbf{A} \mathbf{p}_k$$

continue

(12) GPBiCG 法 (Generalized Polinomial Bi-Conjugate Gradient Method)

BCG 法に前処理としての GP 法を追加した形である。2つのパラメータ ς_k と η_k について、残差 $\mathbf{r}_{k+1}$ のノルムを最小にする様に決める。この積型反復法を GPBi-CG 法と名付けている。

応用領域として、前処理としての教育用数値計算を始めとして、教育研究用である。開発されて年月を経ておらず、十分に普及している訳ではない。どの様な分野に適用させるべきであるか、専門家による今後の研究課題とされている。

GPBiCG 法アルゴリズム

初期ベクトル $\mathbf{r}_0^*$ 、収束判定値 eps 、 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$

$$(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0, \quad \mathbf{r}_0^* = \mathbf{r}_0, \quad \beta_{-1} = 0, \quad \mathbf{t}_{-1} = 0, \quad \mathbf{w}_{-1} = 0$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1} (\mathbf{p}_{k-1} - \mathbf{u}_{k-1})$$

$$\alpha_k = (\mathbf{r}_0^*, \mathbf{r}_k) / (\mathbf{r}_0^*, \mathbf{A} \mathbf{p}_k)$$

$$\mathbf{y}_k = \mathbf{t}_{k-1} - \mathbf{r}_k - \alpha_k \mathbf{w}_{k-1} + \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\mathbf{t}_k = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\varsigma_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}$$

$$\eta_k = \frac{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{t}_k)(\mathbf{A} \mathbf{t}_k, \mathbf{y}_k)}$$

$$\text{if } k=0, \text{ then } \varsigma_k = \frac{(\mathbf{A} \mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A} \mathbf{t}_k, \mathbf{A} \mathbf{t}_k)}, \quad \eta_k = 0$$

$$\mathbf{u}_k = \varsigma_k \mathbf{A} \mathbf{p}_k + \eta_k (\mathbf{t}_{k-1} - \mathbf{r}_k + \beta_{k-1} \mathbf{u}_{k-1})$$

$$\mathbf{z}_k = \varsigma_k \mathbf{r}_k + \eta_k \mathbf{z}_{k-1} - \alpha_k \mathbf{u}_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k + \mathbf{z}_k$$

$$\mathbf{r}_{k+1} = \mathbf{t}_k - \eta_k \mathbf{y}_k - \varsigma_k \mathbf{A} \mathbf{t}_k$$

$$\beta_k = (\alpha_k / \varsigma_k) \cdot (\mathbf{r}_0^*, \mathbf{r}_{k+1}) / (\mathbf{r}_0^*, \mathbf{r}_k)$$

$$\mathbf{w}_k = \mathbf{A} \mathbf{t}_k + \beta_k \mathbf{A} \mathbf{p}_k$$

continue

(13) GMRES 法 (Generalized Minimal Residual Method)

Y.Saad は直交条件を採用し、アーノルデイ原理に基づいた完全直交法 (FOR 法) を提案した (1981)[79]。その後、この考え方をもち 1986 年 Y.Saad と M.H.Schultz らによって GMRES 法が提案された[17]。これは完全直交法の直交条件に代わって、最小条件 MR を使う解法である。残差 $\mathbf{R}$ は最小条件から求められる[64]。

応用領域として、前処理としての教育用数値計算を始めとして、教育研究用である。この解法も開発されて年月を経るおらず、十分に普及している訳ではない。どのような分野に適用させるべきであるか、専門家による今後の研究課題とされている。

GMRES 法アルゴリズム

$\mathbf{x}_0$ は反復の初期値とする。

初期ベクトル $\mathbf{x}_0$ 、収束判定値 eps 、 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\hat{\mathbf{v}}_{k+1} = \mathbf{A} \mathbf{v}_k$$

for $i = 1, 2, \dots, n$ do :

$$h_{i,k} = (\hat{\mathbf{v}}_{k+1}, \mathbf{v}_i)$$

$$\hat{\mathbf{v}}_{k+1} = \hat{\mathbf{v}}_k - h_{i,k} \mathbf{v}_i$$

end

$$h_{i+1,k} = \|\hat{\mathbf{v}}_{k+1}\|$$

$$\mathbf{v}_{k+1} = \frac{\hat{\mathbf{v}}_{k+1}}{h_{i+1,k}}$$

for $i = 1, 2, \dots, n$ do :

$$\begin{pmatrix} h_{i,k} \\ h_{i+1,k} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_i & -\overline{s}_i \\ s_i & \mathbf{c}_i \end{pmatrix} \begin{pmatrix} h_{i,k} \\ h_{i+1,k} \end{pmatrix}$$

end

$$\mathbf{c}_k = \sqrt{\frac{\overline{h}_{k,k} h_{k,k}}{\overline{h}_{k,k} h_{k,k} + \overline{h}_{k+1,k} h_{k+1,k}}}$$

$$s_k = -\frac{h_{k+1,k}}{h_{k,k}} \sqrt{\frac{\overline{h}_{k,k} h_{k,k}}{\overline{h}_{k,k} h_{k,k} + \overline{h}_{k+1,k} h_{k+1,k}}}$$

$$\begin{pmatrix} \mathbf{e}_k \\ \mathbf{e}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_k & -\overline{s}_k \\ \mathbf{v} & \mathbf{c}_k \end{pmatrix} \begin{pmatrix} \mathbf{e}_k \\ \mathbf{e}_{k+1} \end{pmatrix}$$

$$h_{k,k} = -\frac{h_{k+1,k}}{h_{k,k}} \sqrt{\frac{\overline{h}_{k,k} h_{k,k}}{\overline{h}_{k,k} h_{k,k} + \overline{h}_{k+1,k} h_{k+1,k}}}$$

$$h_{i+1,k} = 0$$

if $|\mathbf{e}_{k+1}| \leq \varepsilon \|\mathbf{b}\|$, then

$$\mathbf{y}_k = \mathbf{H}_k^{-1} \|\mathbf{r}_0\| \mathbf{e}_1$$

$$\mathbf{x}_k = \mathbf{x}_0 + \sum_{i=1}^n y_i \mathbf{V}_i$$

end if and stop
end
continue

(14) GCR 法 (Generalized Conjugate Residual Method)

GCR 法は 1983 年 S.C.Eisenstat, H.C.Elman, M.H.Schultz らによって共役勾配法の非対称問題への拡張として考案された。この解法は理論的には GMRES 法と同じく残差を各反復毎に生成するが、近似解の計算を漸化式で行うので GMRES 法のように QR 分解を用いて最小二乗問題を解く必要がないので効率上有利である。アーノルディ原理を用いる GCR 法の導出方法に基づく。

応用領域として、前処理としての教育用数値計算を始めとして、教育研究用である。この解法も開発されて年月を経るおらず、十分に普及している訳ではない。どのような分野に適用させるべきであるか、専門家による今後の研究課題とされている。

正規直交系を定義して、計算ステップを進めることによる。

GCR 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps , $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$, $\mathbf{p}_0 = \mathbf{r}_0$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = (\mathbf{A} \mathbf{p}_k, \mathbf{r}_k) / (\mathbf{A} \mathbf{p}_k, \mathbf{A} \mathbf{p}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\beta_{k,i} = -(\mathbf{A} \mathbf{p}_i, \mathbf{A} \mathbf{r}_{k+1}) / (\mathbf{A} \mathbf{p}_i, \mathbf{A} \mathbf{p}_i), \quad i \leq k$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=0}^k \beta_{k,i} \mathbf{p}_i$$

continue

(15) Orthomin(k)法 (Orthomin(k) Method)

GCR 法では、反復回数 n が増えるに従い、それまでのベクトル列を保存する必要があり、演算量と記憶容量の面で問題があるとされる[70]。この時適当な正の整数 k に対して、リスタート版の 1 つとして Orthomin(k)法が提案された。(k)法の k はリスタートの回数を意味し、以下に用いている添え字の k とは意味の異なるものである。

応用領域として、前処理としての教育用数値計算を始めとして、教育研究用である。この解法も開発されて年月を経るおらず、十分に普及している訳ではない。どのような分野に適用させるべきであるか、専門家による今後の研究課題とされている。

Orthomin(k)法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps , $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$, $\mathbf{p}_0 = \mathbf{r}_0$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = (\mathbf{A} \mathbf{p}_k, \mathbf{r}_k) / (\mathbf{A} \mathbf{p}_k, \mathbf{A} \mathbf{p}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\beta_{k,i} = -(\mathbf{A} \mathbf{p}_i, \mathbf{A} \mathbf{r}_{k+1}) / (\mathbf{A} \mathbf{p}_i, \mathbf{A} \mathbf{p}_i), \quad n-h-1 \leq i \leq n$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \sum_{i=0}^{\min(h-1,k)} \beta_{k,k-i} \mathbf{p}_i$$

$$\left(\mathbf{A} \mathbf{p}_{k+1} = \mathbf{A} \mathbf{r}_{k+1} + \sum_{i=0}^{\min(h-1,k)} \beta_{k,k-i} \mathbf{A} \mathbf{p}_i \right)$$

continue

(16) MRTR 法 (Minimized Residual Method based on the Tree-term Recurrence formula)

数値解法的に残差の最小化条件に重点をおいた CR 法と同じ形式である。この時の残差は三項漸化式で生成される多項式として表現される。実際の収束計算も CR 法に近い値をとる。数値の誤差の面からは丸め誤差に頑健である長所が知られている[1]。

応用領域として、前処理としての教育用数値計算を始めとして、教育研究用である。開発されて年月を経ておらず、十分に普及している訳ではない。どの様な分野に適用させるべきであるか、専門家による今後の研究課題とされている。

MRTR 法アルゴリズム

初期ベクトル $\mathbf{x}_0$ 、収束判定値 eps 、 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0$

$$\mathbf{y}_0 = -\mathbf{r}_0, \quad \nu_0 = (\mathbf{y}_0, \mathbf{y}_0)$$

while $\|\mathbf{x}_k\|_2 > \text{eps} * \|\mathbf{b}\|_2$ do , ($k=0,1,2,\dots$)

$$\mu_k = \frac{\nu_k(\mathbf{A} \mathbf{r}_k, \mathbf{r}_k)}{\nu_k(\mathbf{A} \mathbf{r}_k, \mathbf{A} \mathbf{r}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{r}_k)(\mathbf{A} \mathbf{r}_k, \mathbf{y}_k)},$$

$$\nu_k = \frac{-(\mathbf{y}_k, \mathbf{A} \mathbf{r}_k)(\mathbf{A} \mathbf{r}_k, \mathbf{r}_k)}{\nu_k(\mathbf{A} \mathbf{r}_k, \mathbf{A} \mathbf{r}_k) - (\mathbf{y}_k, \mathbf{A} \mathbf{r}_k)(\mathbf{A} \mathbf{r}_k, \mathbf{y}_k)},$$

$$\text{if } k=0, \text{ then } \mu_k = \frac{(\mathbf{A} \mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{A} \mathbf{r}_k, \mathbf{A} \mathbf{r}_k)}, \quad \nu_k = 0$$

$$\tau_{k+1} = \nu_k(\mathbf{A} \mathbf{r}_k, \mathbf{r}_k),$$

$$\mathbf{p}_k = \mathbf{r}_k + \frac{\mu_{k-1}}{\mu_k} \nu_k \mathbf{p}_{k-1},$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mu_k \mathbf{p}_k,$$

$$\mathbf{y}_{k+1} = \nu_k \mathbf{y}_k + \nu_k \mathbf{A} \mathbf{r}_k,$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{y}_{k+1}$$

(17) 切除残差法 (Exclusive Residual Method)

残差の方程式を定義する。その部分を旧来の基礎的な反復解法 (ADI 法) を用いて、一定回の反復を行い、収束値より離れた 1 つの近似値を求める。この残差方程式の境界条件を満たす近似解を求めることで、近似解とその時の摂動量で、次の近似解を求める。この近似値が切除残差法の出発点とする。そして残差最小化係数を求める。残差最小化係数は始めの数個で良い。この時の残差にノルムを定義することで、残差の最小を急速に導く事にある。計算プロセスの中では最小自乗法の考えも採り入れている。残差最小化係数と摂動量により、線形 1 次結合で残差の合成摂動量を求めて、近似解を次々に更新して行く。残差が最小(0)になった段階で収束とみなす。

応用領域として、教育用数値計算を始めとして、教育研究用である。ソフトベンダーがソフトウェアとして商品を作り上げたこともあり、このソフトウェア専門家の努力により、2005 年以降の流体力学を始め、基礎物理分野、計算化学分野等に普及が始まった。ユーザーとして手法の意味は理解していなくても、速く計算結果が得られれば良いということから、この解法は確かに広まりつつある。

切除残差法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A} \mathbf{x}_k, \quad \text{摂動量 } \phi_k = \mathbf{x} - \mathbf{x}_k, \quad \text{近似摂動量 } \varphi_k$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\mathbf{A} \phi_k = \mathbf{r}_k$$

(apply to ADI Method)

$$\frac{\partial \|\mathbf{r}_{k+1}\|^2}{\partial \alpha_l} = 0, \quad (l=1,2,\dots,L)$$

$$\phi_k = \alpha_1 \varphi_k + \sum_{l=2}^L \alpha_l \phi_{k-l+1}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \phi_k$$

$$\mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A} \mathbf{x}_{k+1}$$

continue

行列の最大固有値と最小固有値の比率が大きい場合、CG 法系の反復法による解法では有効性が得られない場合がある。こうした問題に対する解法に効果的である。

1・7. 反復法、固有値問題の解法

本来固有値は、特性方程式の根として与えられる。固有値には左側だけの行列で表される標準固有値問題と、左右に行列が表現される一般固有値問題がある。この固有値問題は、一般的に初期値を与えて反復法で解が進められる[9]。

[1] 近代的解法

固有値問題は歴史的分類上、近代的解法に属する。行列には固有値があり、この固有値は行列の性質に、つまり対象となる事象に大きな影響力を及ぼしていると考えられる[92]。固有値問題が考え出されたのは、第二次対戦後の 1950 年頃であり、これを数値計算として、固有値を求める方法が盛んに探索されたのは 1960 年代である。行列の固有値には、構造物の動力学に用いられる振動問題や共振周波数を求める問題、多変量解析にみる空間軸の重み付けや、更に原子炉の臨界値問題等が挙げられる。数値解法として、基本直交して求める方法、各行列を関数の列として二分して求める方法、反復の形にして初期値から順次求める方法、直交行列と上三角行列の積に分解して、直交化に基づく計算で数値解を求めて行く方法等がある。いずれも社会で広く活用されている解法である。

[2] 固有値問題の考え方

一般固有値問題 $\mathbf{K}\mathbf{x} = \lambda\mathbf{M}\mathbf{x}$ に対して、標準固有値問題を $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ とする。

$\mathbf{A}$ は n 次の正方行列とする時、標準固有値問題は $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ の形となる。この方程式が $\mathbf{x} = \mathbf{0}$ 以外の解を持つときは、 λ が n 次方程式の $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ の解となることである。この方程式を $\mathbf{A}$ の特性方程式、又は固有方程式という。固有値は重根を含めて n 個存在する。 $\mathbf{P}$ を正則行列とすると $\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ の変換を相似変換 (similarity transformation) と言う。相似変換によって固有値は変化しない。

ここで $\mathbf{A}$ の固有値を求めるのに、特性方程式をそのまま解くのではなく、相似変換によって対角行列、または三角行列に変換するのが良い。つまりそれらの行列では、対角要素が固有値となるからである。例えば、ユニタリ行列 $\mathbf{U}$ による相似変換で誤差を $\mathbf{E}$ として

$$\mathbf{U}^H \mathbf{A} \mathbf{U} + \mathbf{E} = \mathbf{U}^H (\mathbf{A} + \mathbf{G}) \mathbf{U}$$

すなわち近似的な相似変換が $\mathbf{A} + \mathbf{G}$ に対する相似変換に等しいとすると、 $\mathbf{E} = \mathbf{U}^H \mathbf{G} \mathbf{U}$

だから $\|\mathbf{F}\|_2 = \|\mathbf{E}\|_2$ となって誤差は拡大しないことが分かる。

(1) ハウスホルダー法 (Householder Method)

標準固有値問題を解くにおいて、ユニタリ変換による相似変換によって、 $\mathbf{A}$ を上三角行列にする。 $\mathbf{A}$ がエルミート行列であれば、対角行列となる。これは鏡映変換と呼ばれる[4]。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理、又データ解析系として、統計学、多変量解析学等の固有値問題のあらゆる分野に多く見受けられる。1950 年代以降の標準固有値問題の解法は、この方法が使われている事が多い。2000 年代においても使われ、50 年以上愛用されている。標準固有値問題を解くにおいて重要な位置づけにある。

ハウスホルダー法の計算ステップ

ハウスホルダー法の変換を 2 段に分けて行う。第 1 段で上ヘッセンベルグ行列 ($i > j+1$; $a_{ij} = 0$) に変換し、第 2 段でそれを上三角行列に変換する。第 1 段に用いられるのが、ハウスホルダー法である。長さの等しい 2 つの異なるベクトル $\mathbf{x}, \mathbf{y}$ があると

$$\text{行列} \quad \mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^T \quad \text{の形となる。}$$

$$\text{但し、} \quad \mathbf{u} = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} \quad \text{を用いると} \quad \mathbf{y} = \mathbf{H}\mathbf{x} \quad \text{とする事が出来る。}$$

行列 $\mathbf{H}$ は対称な直交行列で $\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$ を満たし、鏡像変換と呼ばれる。 $\mathbf{A}$ を相似変換によって、ヘッセンベルグ行列にするには、 $\mathbf{A}$ の第 1 列に着目して

$$\mathbf{x} = (a_{21}, a_{31}, \dots, a_{n1})^T, \quad \mathbf{y} = (-\sigma, 0, \dots, 0)^T$$

として、 $\mathbf{H}_1\mathbf{x} = \mathbf{y}$ となる変換行列 $\mathbf{H}_1$ を作る。

$$\|\mathbf{x}\| = \|\mathbf{y}\| \quad \text{から}$$

$$\sigma = \text{sign}(a_{21}) \sqrt{\sum_{i=2}^n a_{i1}^2}$$

である。

$$\text{但し、} \quad \text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

である。この σ は a_{21} と同符号にすることを意味する。

$$(1.127) \quad \mathbf{H}_1 = \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^T$$

とすると、式 (1.127) の $\mathbf{u}_1$ は

$$(1.128) \quad \mathbf{u}_1 = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} = \frac{1}{\|\mathbf{x} - \mathbf{y}\|} (a_{21} + \sigma, a_{31}, \dots, a_{n1})$$

但し、

$$(1.129) \quad \|\mathbf{x} - \mathbf{y}\| = \sqrt{2\sigma(\sigma + a_{21})}$$

である。

第 1 段の変換行列 $\mathbf{Q}$ は

$$(1\cdot130) \quad \mathbf{Q}_1 = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{pmatrix}$$

の形となり、 $\mathbf{A}$ を

$$(1\cdot131) \quad \mathbf{A} = \begin{pmatrix} a_{11} & \mathbf{x}^T \\ \mathbf{x} & \mathbf{B} \end{pmatrix}$$

と表すと、式 (1・130) 式 (1・131) から、相似変換 は $\mathbf{Q}_1 \mathbf{A} \mathbf{Q}_1$ の形となる。

$$(1\cdot132) \quad \begin{aligned} \mathbf{Q}_1 \mathbf{A} \mathbf{Q}_1 &= \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{pmatrix} \begin{pmatrix} a_{11} & \mathbf{x}^T \\ \mathbf{x} & \mathbf{B} \end{pmatrix} \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{pmatrix} \\ &= \begin{pmatrix} a_{11} & (\mathbf{H}_1 \mathbf{z})^T \\ \mathbf{H}_1 \mathbf{x} & \mathbf{H}_1 \mathbf{B} \mathbf{H}_1 \end{pmatrix} \end{aligned}$$

式 (1・132) にみる第 1 列の $\mathbf{H}_1 \mathbf{x}$ は $\mathbf{y}$ だから第 1 列の第 3 行から第 n 行までがすべて 0 になる。

第 2 列に対する変換は右下の $n-1$ 次行列 $\mathbf{H}_1 \mathbf{B} \mathbf{H}_1$ に対して、第 1 列の時と同様の考え方によって変換行列 $\mathbf{H}_2$ を作って

$$(1\cdot133) \quad \mathbf{Q}_2 = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \\ \mathbf{0} & & \mathbf{H}_2 \end{pmatrix}$$

による相似変換を行う。以下同様に第 3 列、第 4 列、 $\dots$ 、第 $n-2$ に対する変換行列

$\mathbf{Q}_3, \mathbf{Q}_4, \dots, \mathbf{Q}_{n-2}$ を用いて

$$(1\cdot134) \quad \mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_{n-2}$$

$$(1\cdot135) \quad \begin{aligned} \mathbf{Q}^{-1} \mathbf{A} \mathbf{Q} &= \mathbf{Q}_{n-2} \cdots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{A} \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_{n-2} \\ &= \text{上ヘッセンベルグ行列} \end{aligned}$$

が得られる。

$\mathbf{A}$ が対称行列の時は、式 (1・132) で $\mathbf{z} = \mathbf{x}$ であるから $\mathbf{H}_1 \mathbf{x} = \mathbf{y}$ と同時に

$(\mathbf{H}_1 \mathbf{z})^T = (\mathbf{H}_1 \mathbf{x})^T = \mathbf{y}^T$ となって、第 1 行の第 3 列から先も全て 0 になるので、三重対角行列が得られる。

(2) QR 法 (The QR Transformation Method)

構造系の固有値の解法を研究していたフランシス(Francis)は、コレスキーの分解を研究している中で、行列の分解理論に特に関心を寄せた。行列をある方式に沿って適切に分解させる事が出来れば、効率的である。同時代におけるグラムシュミット(Gram-Schmidt)の直交化の研究から考えを得て、

もとの行列 $\mathbf{A}$ を、直交行列 $\mathbf{Q}$ と右上三角行列 $\mathbf{R}$ に分解する方法を提唱した。ここで

$\mathbf{A}$ が正則な行列である場合、この行列が QR 分解されたとき、これは $\mathbf{A}$ の $\mathbf{Q}$ に対する相似変換 (Similar Transformation) となる。この相似変換の繰り返しにより、 $\mathbf{A}$ の左下半分は 0 に収束し、最後に対角線上に $\mathbf{A}$ の固有値が並ぶ。これを QR 法として提案した (Computer Journal) (1961)。フランシス (Francis) と仲の良い研究者クブラノフスカヤ (Kburanofskaya) は、主に応用面で協力研究した。この解法の後、固有値解法がいろいろ研究される時代を迎える。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理、又データ解析系として、統計学、多変量解析学等の固有値問題のあらゆる分野に多く見受けられる。ハウスホルダー法と同じく基本的な方法であり、標準固有値問題を解くにおいて重要な位置づけにある。中でも代表的応用事例として、NASA 開発 NASTRAN の動的解析の標準的解法として用いられている。1960 年代以降の標準固有値問題の解法は、この QR 法が使われている事が多い。2000 年代においても使われ、ハウスホルダー法と同じく 40 年以上愛用されている。標準固有値問題を解くにおいて重要な位置づけにある。

QR 法の計算ステップ

考え方は、上ヘッセンベルグ行列 $\mathbf{A}$ に対して、相似変換を繰り返して行った上三角行列に対する方法をとる。直交行列 $\mathbf{Q}$ を左から掛けることで、上三角行列に変換する方法である。先ず行列 $\mathbf{A}$ を直交行列 $\mathbf{Q}$ と右上三角行列 $\mathbf{R}$ の積に分解する。

$$(1.136) \quad \mathbf{Q}_1^T \mathbf{A}_1 = \mathbf{R}_1$$

式 (1.136) は

$$(1.137) \quad \mathbf{A}_1 = \mathbf{Q}_1 \mathbf{R}_1$$

と書ける。式 (1.137) を QR 分解と呼んでいる。 $\mathbf{Q}$ と $\mathbf{R}$ を逆順に掛けて

$$(1.138) \quad \mathbf{A}_1 = \mathbf{Q}_1 \mathbf{R}_1 = \mathbf{Q}_1^T \mathbf{A}_1 \mathbf{Q}_1$$

式 (1.138) によって $\mathbf{A}_2$ を作る。 $\mathbf{A}_2$ は $\mathbf{A}_1$ と同じく相似である。これは上ヘッセンベルグ行列になる。

一般に

$$(1.139) \quad \mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k$$

$$(1.140) \quad \mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k$$

式 (1.139) 式 (1.140) によって相似な行列の列 $\{\mathbf{A}_k\}$ を計算する。 QR 分解は、ギブンス回転行列 (Givens rotation) を用いて行う事が出来る。 $\mathbf{A}_k$ は上ヘッセンベルグ行列であるから、 $a_{i,i-1} (2 \leq i \leq n)$ を 0 にすれば、上三角行列になる。 $\mathbf{A}_k$ は (i,i) 要素と (i,i) 要素が $\cos \theta$ 、 $(i-1,i)$ 要素が $\sin \theta$ 、 $(i,i-1)$ 要素が $-\sin \theta$ で、他は単位行列と同じ行列である。

$$(1 \cdot 141) \quad \mathbf{G}_{i,i-1} = \begin{pmatrix} 1 & & & & & & & & \\ & \cdot & & & & & & & \mathbf{0} \\ & & 1 & & & & & & \\ & & & \cos \theta & \sin \theta & & & & \\ & & & -\sin \theta & \cos \theta & & & & \\ & & & & & 1 & & & \\ & \mathbf{0} & & & & & \cdot & & \\ & & & & & & & 1 \end{pmatrix} \begin{matrix} (i-1) \\ i \\ (i-1) \\ i \\ (i-1) \\ i \\ (i-1) \\ i \end{matrix}$$

$\mathbf{A}_k$ の $(i, i-1)$ 要素を 0 とするために

$$\begin{aligned}\cos \theta &= \frac{\mathbf{a}_{i-1,i-1}}{\sqrt{\mathbf{a}_{i-1,i-1}^2 + \mathbf{a}_{i,i-1}^2}} \\ \sin \theta &= \frac{\mathbf{a}_{i,i-1}}{\sqrt{\mathbf{a}_{i-1,i-1}^2 + \mathbf{a}_{i,i-1}^2}}\end{aligned}\tag{1.142}$$

とする。式 (1・142) にみるこの変換を $i = 2, 3, \dots, n$ の順に行うと、上三角行列 $\mathbf{R}_k$ が得られる。

$$\mathbf{R}_k = \mathbf{G}_{n,n-1}, \dots, \mathbf{G}_{32}, \mathbf{G}_{21} \mathbf{A}_k$$

従って

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k$$

とすると

$$\mathbf{Q}_k = \left(\mathbf{G}_{n,n-1} \cdots \mathbf{G}_{32} \mathbf{G}_{21} \right)^{-1}$$

となる。ギブンス回転行列 $\mathbf{G}_{i,i-1}$ は直交行列であるから $\mathbf{Q}_k$ も直交行列となる。QR 法によっ

て作られた列 $\{\mathbf{A}_k\}$ は $k \rightarrow \infty$ の時、 $\mathbf{A}_k$ が上三角行列に収束する。上三角行列になれば、

その対角要素が固有値となる。収束の速さは $\mathbf{A}_k$ の要素を $\mathbf{a}_{ij}^{(k)}$ とした時

$$\begin{aligned} \mathbf{a}_{\text{ii}}^{(\text{k})} &= \lambda_{\text{i}} + \mathbf{O}(\mathbf{r}_{\text{i}}^{\text{k}}) \\ \mathbf{a}_{\text{i,i-1}}^{(\text{k})} &= \mathbf{O}(\mathbf{r}_{\text{i}}^{\text{k}}) \end{aligned} \quad (1 \cdot 143)$$

但し、

$$(1.144) \quad r_i = \max_i \left(\left| \frac{\lambda_i}{\lambda_{i-1}} \right|, \left| \frac{\lambda_{i+1}}{\lambda_i} \right| \right)$$

である。

式(1・144)において $\lambda_1, \lambda_2, \dots, \lambda_n$ は $\mathbf{A}_k$ の固有値の絶対値の大きい方から順に並べたもので、

$$\lambda_0 = \infty, \lambda_{n+1} = 0 \quad \text{としておく。}$$

この時
$$r_n = \left| \frac{\lambda_n}{\lambda_{n-1}} \right| \quad \text{であり、}$$

これが小さい時

$$\begin{aligned} a_{nn} &\rightarrow \lambda_n \\ a_{n,n-1} &\rightarrow 0 \end{aligned}$$

の収束が速い。

$a_{n,n-1}$ が 0 になれば、 $\lambda_n = a_{nn}$ として $\mathbf{A}_k$ から第 n 行、第 n 列を取り除いて次数を下げる事が出来る。QR 法は 1 つの固有値が求められると、次数が 1 ずつ小さくなるので、QR 分解の手間が少なくなる。 r_n を小さくする為に、固有値の原点を移動させたシフトを行う。シフト量を p とすると、 $\mathbf{A}_k - p\mathbf{I}$ の固有値は $\lambda_i - p$ であるから、適当な p を選ぶことで、

$$r_n' = \left| \frac{\lambda_n - p}{\lambda_{n-1} - p} \right|$$

を小さくする事が出来る。 p としては λ_n の近似値をとれば良い。

p_k を決めるシフト付き QR 法は

$$(1.145) \quad \mathbf{A}_k - p_k \mathbf{I} = \mathbf{Q}_k \mathbf{R}_k$$

$$(1.146) \quad \mathbf{A}_{k+1} = \mathbf{R}_k \mathbf{Q}_k + p_k \mathbf{I}$$

となる。

式 (1.145) 式 (1.146) において、同様に $\mathbf{A}_k$ と $\mathbf{A}_{k+1}$ は相似である。シフト量 p としては、

$\mathbf{A}_k$ の (n,n) の要素 a_{nn} とする方法と $\mathbf{A}_k$ の右下の 2×2 行列の固有値のうち a_{nn} に近いものをとる方法がある。十分大きな k に対して、 $\mathbf{A}_k$ は収束して行き対角行列に近づき、その対角線上に固有値が絶対値の大きさの順に並ぶ。行列 $\mathbf{A}$ が非対称であっても QR 法は収束する。

(3) 二分法 (Bisection Method)

$\mathbf{A}$ が実対称行列の時は、ハウスホルダー変換によって対称三重対角行列が得られるので、スツルムの定理を用いた二分法によって固有値を求める事が出来る。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理、又データ解析系として、統計学、多変量解析学等の固有値問題のあらゆる分野に多く見受けられる。ハウスホルダー法、QR 法と同じく基本的な方法であり、標準固有値問題を解くにおいて重要な位置づけにある。

二分法の計算ステップ

三重対角行列 (Tridiagonal Matrix) を $\mathbf{T}$ としてその要素を

$$t_{ii} = \alpha_i, \quad 1 \leq i \leq n$$

$$t_{i,i+1} = t_{i+1,i} = \beta_i, \quad 1 \leq i \leq n-1$$

とする。副対角要素 β_i の中に 0 のものがあるとすると、 $\mathbf{T}$ は幾つかの三重対角行列に分かれるので、それぞれの三重対角行列の固有値を計算する。 $\beta_i \neq 0 \quad (1 \leq i \leq n-1)$ と仮定する。

固有値 λ は特性方程式 $\det(\mathbf{T} - \lambda \mathbf{I}) = 0$ の根である。 k 次主小行列を $f_k(\lambda)$ とすると、

$$\begin{aligned} f_0(\lambda) &= 1 \\ f_1(\lambda) &= \alpha_1 - \lambda \\ &\dots\dots\dots \\ f_k(\lambda) &= (\alpha_k - \lambda)f_{k-1}(\lambda) - \beta_{k-1}^2 f_{k-2}(\lambda) \end{aligned} \quad (1 \cdot 147)$$

による。式 (1・147) より関数列 $\{f_k(\lambda)\}$ は、スツルム列 (Sturm sequence) の性質として、次が成り立つ。

(1) $f_0(\lambda)$ は定符号

(2) $f_k(\lambda) = 0$ の時 $f_{k+1}(\lambda)f_{k-1}(\lambda) < 0$

を満たす。

$$f_{k+1}(\lambda) = (\alpha_{k+1} - \lambda)f_k(\lambda) - \beta_k^2 f_{k-1}(\lambda)$$

であるから $f_k(\lambda) = 0$ の時

$$f_{k+1}(\lambda)f_{k-1}(\lambda) = -\beta_k^2 f_{k-1}^2(\lambda) \leq 0$$

となる

ここで仮定より $\beta_i \neq 0$ であるから、等号が成立するのは $f_k(\lambda) = 0$ の時である。

[スツルムの定理]

λ より小さい $\mathbf{T}$ の固有値の個数は、数列 $\{f_k(\lambda)\}$ の符号変化の回数 $N(\lambda)$ に等しい。

但し、 $f_k(\lambda) = 0$ の時は $f_k(\lambda) = 0$ と $f_{k-1}(\lambda) = 0$ は同符号であるとみなす。つまり $\mathbf{T}$ の固有値を全て含む区間 (a, b) を定める。この区間を二等分して中点 c に対して $N(c)$ を計算し、 $N(c) \geq m$ であれば λ_m は、区間 (a, c) の中に存在するので b を c に置き換える。そうでなければ、区間 (c, b) の中に存在するので、 a を c に置き換える。いずれにしても λ_m の存在区間の幅は、 $1/2$ に出来る。

これを繰り返すことで任意の精度で固有値が計算できる。

(4) ベキ乗法 (Power method)

行列 $\mathbf{A}$ の固有値、固有ベクトルを求める方法として、ベキ乗法が古くから知られている。これは適当な初期ベクトル $\mathbf{A}$ を次々に乗じる形をとる。最近では、固有値を求める為に、この方法をそのままの形で用いられることは少なくなっている。基礎としては、いろいろな解法にベキ乗法の考えが活かされている

応用領域として、教育の為に数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理、又データ解析系として、統計学、多変量解析学、この多変量解析学の中の主成分分析、因子分析、判別分析、正準分析等に良く用いられている。更に OR 系等の固有値問題のあらゆる分野に多く見受けられる。ハウスホルダー法、QR 法と同じく基本的な方法である。1960 年代以降の標準固有値問題の解法は、このベキ乗法が使われている事が多い。2000 年代においても使われ、QR 法と同じく 40 年以上愛用されている。本論文の研究テーマである原子炉内における中性子拡散方程式の数値解法も、そのオリジナルコードは米国オークリッジの原子力研究所で研究開発されたが、そこでも主力の数値解法の固有値問題は、このベキ乗法が用いられている。スムーズで数値解が得られ易く、社会で広く使われている。標準固有値問題を解くにおいて重要な位置づけにある。

ベキ乗法の計算ステップ

ベキ乗法の原理は、線型 1 次結合に対して固有値の比を掛けてゆく事にある。

与えられた $n \times n$ 行列 $\mathbf{A}$ の固有値 $\lambda_1, \lambda_2, \dots, \lambda_n$ は

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$$

を満たしているとする。絶対値最大の固有値 λ_1 には縮退はないと仮定する。

適当な初期ベクトル $\mathbf{x}_0$ から始めて

$$(1.148) \quad \mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}, \quad k=1, 2, \dots$$

によって、絶対値最大の固有値に対する固有ベクトルを求める方法である。この反復を続けるに従って、 $\mathbf{x}_k$ は次第に $\mathbf{A}$ の絶対値最大の固有値 λ_1 に属する固有ベクトルに収束して行く。

λ_1 に属する $\mathbf{A}$ の固有ベクトル $\mathbf{u}_1$ は

$$(1.149) \quad \mathbf{A}\mathbf{u}_1 = \lambda_1\mathbf{u}_1$$

初期ベクトル $\mathbf{x}_0$ は、 $\{\mathbf{u}_i\}$ の 1 次結合によって、

$$(1.150) \quad \mathbf{x}_0 = c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \dots + c_n\mathbf{u}_n$$

と表す事が出来る。ここで $c_1 \neq 0$ とする。

式 (1・150) を式 (1・149) に代入して

$$\begin{aligned}
 \mathbf{x}_k &= \mathbf{A}_k \mathbf{x}_0 = c_1 \lambda_1^k \mathbf{u}_1 + c_2 \lambda_2^k \mathbf{u}_2 + \cdots + c_n \lambda_n^k \mathbf{u}_n \\
 (1 \cdot 151) \quad &= c_1 \lambda_1^k \left(\mathbf{u}_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}_2 + \cdots + \frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}_n \right)
 \end{aligned}$$

となる。仮定から $\left| \frac{\lambda_j}{\lambda_1} \right| < 1$, $j \neq 1$ であるから、 k を大きくすると、 $\mathbf{x}_k$ において固有値 λ_1

に属する固有ベクトル成分 $c_1 \lambda_1^k \mathbf{u}_1$ が次第に優越する形となる。この形式がベキ乗法であり、 $\mathbf{A}$ の絶対値最大の固有値に属する固有ベクトルを求める事が出来る。

(5) 逆反復法 (Inverse Iteration Method)

逆反復法はベキ乗法(Power method)系統の方法からなる。一般には行列 $\mathbf{A}$ をハウスホルダー法によって三重対角化して、更に固有値の近似値が二分法によって求められている時、引き続きその固有ベクトルを求める為に適用する事が多い。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理、又データ解析系として、統計学、多変量解析学、この多変量解析学の中の主成分分析、因子分析、判別分析、正準分析等に良く用いられている。更に OR 系等の固有値問題のあらゆる分野に多く見受けられる。ハウスホルダー法、QR 法、ベキ乗法と同じく基本的な方法である。1960 年代以降の標準固有値問題の解法は、前項のベキ乗法、又は逆反復法が使われている事が多い。2000 年代においても使われ、QR 法と同じく 40 年以上愛用されている。標準固有値問題を解くにおいて重要な位置づけにある。

逆反復法の計算ステップ

$\mathbf{A}$ の固有値 λ_m に属する固有ベクトルを $\mathbf{u}_m$ とする。

固有値 λ_m の近似値 μ が求められたとして、

$$(1 \cdot 152) \quad \mathbf{B} = (\mu \mathbf{I} - \mathbf{A})^{-1}$$

として $\mathbf{A} \mathbf{u}_m = \lambda_m \mathbf{u}_m$ であるから、

$$(1 \cdot 153) \quad \mathbf{B} \mathbf{u}_m = \frac{1}{\mu - \lambda_m} \mathbf{u}_m$$

が成り立つ。近似値 μ は λ_m に十分に近く、他の固有値 λ_j , $j \neq m$ とは、次の関係が成り立つ。

$$(1 \cdot 154) \quad \frac{1}{|\mu - \lambda_m|} > \frac{1}{|\mu - \lambda_j|}, \quad j \neq m$$

$\frac{1}{(\mu - \lambda_m)}$ は行列 $\mathbf{B}$ の絶対値最大の固有値になる。

初期ベクトル $\mathbf{x}_0$ から始めて、 $\mathbf{B}$ にベキ乗法を適用して、

$$(1.155) \quad \mathbf{x}_k = \mathbf{B}\mathbf{x}_{k-1}, \quad k=1,2,\dots$$

なる反復を繰り返せば、 $\mathbf{x}_k$ には λ_m に属する固有ベクトル成分 $\mathbf{u}_m$ が優先される。

式 (1.155) の各反復に対応して、これと同値な方程式

$$(1.156) \quad (\mu \mathbf{I} - \mathbf{A})\mathbf{x}_k = \mathbf{x}_{k-1}$$

を $\mathbf{x}_k$ について解く事になる。近似値 μ は λ_m に十分に近ければ、式 (1.155) における

$\mathbf{u}_m$ の収束は速くなる。式 (1.156) を反復して何回も解くとき、 $\mu \mathbf{I} - \mathbf{A}$ をピボット部分選択

を行う。ガウス消去法により LU 分解をする。置換行列を $\mathbf{P}$ として

$$(1.157) \quad \mathbf{P}(\mu \mathbf{I} - \mathbf{A}) = \mathbf{L}\mathbf{U}$$

式 (1.156) の LU 分解の出来る方程式は

$$(1.158) \quad \mathbf{L}\mathbf{U}\mathbf{x}_k = \mathbf{P}\mathbf{x}_{k-1}$$

となる。 $k=0$ とおくと

$$(1.159) \quad \mathbf{U}\mathbf{x}_1 = \mathbf{L}^{-1}\mathbf{P}\mathbf{x}_0$$

となる。初期ベクトル $\mathbf{x}_0$ が任意にとれる為、 $\mathbf{L}^{-1}\mathbf{P}\mathbf{x}_0$ 全体も任意にとれる。

従って

$$(1.160) \quad \mathbf{L}^{-1}\mathbf{P}\mathbf{x}_0 = (1, 1, \dots, 1)^T$$

とする。近似値 μ は λ_m に十分に近ければ、2回目の反復

$$(1.161) \quad \mathbf{L}\mathbf{U}\mathbf{x}_2 = \mathbf{P}\mathbf{x}_1$$

十分精度の高い固有ベクトルを求める事が出来る。逆反復法では $\mathbf{A}$ がハウスホルダー法を適用して導かれた三重対角行列の場合は、式 (1.161) で得られた固有ベクトルを $\mathbf{x}$ とするとき

$$(1.162) \quad \mathbf{x} = \mathbf{P}\tilde{\mathbf{x}}$$

により元の行列の固有ベクトル $\mathbf{x}$ に戻すことで求められる。

(6) ランチョス法 (Lanczos Method)

ガウス・ザイディルの反復法や、その加速パラメータの効率化としての SOR 法の反復計算を研究してきたランチョスは、ある特定の行列、優対角行列の計算の容易性に着目した。更にその行列の持つ性質を保持する考えを入れて、効率的な反復法を試みた。特に疎な行列である場合、ランチョスの反復法が有効であると提唱した(1956)。この考え方では、行列の性質を保持したまま反復させる事が可能である。(この考えが現在の $\mathbf{M}$ 行列に影響を及ぼす。) ランチョス法は特に大規模な固有値

計算において効果を発揮することが出来る[69]。もとの行列を変形することなく三重対角化する事で、疎行列の性質がそのまま活かされる。基本的に行列とベクトルの積演算であるため、ベクトル化が容易であり、スーパーコンピュータ[96]で効力を発揮する事が出来る[108]。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理等の固有値問題のあらゆる分野に多く見受けられる。ハウスホルダー法、QR法と同じく基本的な方法であり、標準固有値問題を解くにおいて重要な位置づけにある。代表的な応用事例として、原子力工学の輸送方程式の解法などに見つける事が出来る。1980年代以降の標準固有値問題の解法は、このランチョス法が使われている事が多い。2000年代においても使われている。ランチョス法として、標準固有値問題を解くにおいて重要な位置づけにある。

計算のステップ

ランチョス法の考え方

行列 $\mathbf{A}$ が大型行列の時は、その性質を有効に活用するために行列の形式を変えない方が良い。ランチョス法はベクトルに行列 $\mathbf{A}$ を掛ける演算の繰り返しにより、実対称行列 $\mathbf{A}$ に相似な三重対角行列 $\mathbf{T}$ を計算する方法である。三重対角行列 $\mathbf{T}$ の対角要素を α_i 、副対角要素を β_i とする。

$\mathbf{A}$ は $n \times n$ 対称行列として、直交行列 $\mathbf{P}$ を選ぶ。

$$(1.163) \quad \mathbf{T} = \mathbf{P}^{-1} \mathbf{A} \mathbf{P}$$

式 (1.163) において、 $\mathbf{A}$ が対称であるから、 $\mathbf{T}$ も対称となる。

$$(1.164) \quad \mathbf{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \alpha_{n-1} & \beta_{n-1} \\ \mathbf{0} & & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

任意の正規化された初期ベクトル $\mathbf{q}_1$ を選んで

$$(1.165) \quad \begin{aligned} \mathbf{A} \mathbf{q}_1 &= \alpha_1 \mathbf{q}_1 + \beta_1 \mathbf{q}_2 \\ \mathbf{A} \mathbf{q}_2 &= \beta_1 \mathbf{q}_1 + \alpha_2 \mathbf{q}_2 + \beta_2 \mathbf{q}_3 \\ &\dots\dots\dots \\ \mathbf{A} \mathbf{q}_k &= \beta_{k-1} \mathbf{q}_{k-1} + \alpha_k \mathbf{q}_k + \beta_k \mathbf{q}_{k+1} \\ &\dots\dots\dots \\ \mathbf{A} \mathbf{q}_n &= \beta_{n-1} \mathbf{q}_{n-1} + \alpha_n \mathbf{q}_n \end{aligned}$$

によって、式 (1.165) からベクトル $\mathbf{q}_k$ 、及びスカラ α_k 、 β_k を計算する。

α_k は $\mathbf{q}_k$ と $\mathbf{q}_{k+1}$ 及び $\mathbf{q}_{k-1}$ との直交条件

$$(\mathbf{q}_k, \mathbf{q}_{k+1}) = 0, \quad (\mathbf{q}_k, \mathbf{q}_{k-1}) = 0$$

から

$$(1.166) \quad \alpha_k = (\mathbf{q}_k, \mathbf{A}\mathbf{q}_k)$$

により計算され、 β_k は $\mathbf{q}_{k-1}$ の正規化条件

$$(1.167) \quad \beta_k = \|\mathbf{A}\mathbf{q}_k - \beta_{k-1}\mathbf{q}_{k-1} - \alpha_k\mathbf{q}_k\|$$

となる。式 (1.167) の $\mathbf{q}_{k+1} \neq 0$ であれば $\beta_k \neq 0$ である。この様なベクトル列 $\{\mathbf{q}_k\}$ をランチョスベクトルと呼ぶ。ランチョス法によって得られた三重対角行列 $\mathbf{T}$ が $\mathbf{A}$ に相似である。ベクトル $\{\mathbf{q}_k\}$ を列とする行列を $\mathbf{Q}$ とすると、 $\{\mathbf{q}_k\}$ の正規直交生から $\mathbf{Q}$ は直交行列である。

式 (1.165) は $\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{T}$ と書ける。

$$(1.168) \quad \mathbf{T} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}$$

となって $\mathbf{T}$ は $\mathbf{A}$ と相似である。

ランチョス法アルゴリズム

$\|\mathbf{q}_1\| = 1$ である様な初期ベクトル $\mathbf{q}_1$ を選び、 $\mathbf{p}_1 = \mathbf{A}\mathbf{q}_1$, 収束判定値 eps

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{p}\|$ do , ($k=1,2,3,\dots$)

$$\alpha_k = (\mathbf{q}_k, \mathbf{p}_k)$$

$$\mathbf{r}_k = \mathbf{p}_k - \alpha_k \mathbf{q}_k$$

$$\beta_k = \|\mathbf{r}_k\|$$

$$\mathbf{q}_{k+1} = \frac{\mathbf{r}_k}{\beta_k}$$

$$\mathbf{p}_{k+1} = \mathbf{A}\mathbf{q}_{k+1} - \beta_k \mathbf{q}_k$$

continue

主要な計算はベクトルに $\mathbf{A}$ を掛ける計算量 $\mathbf{A}\mathbf{q}_{k+1}$ であるが、 $\mathbf{A}$ が疎行列であれば、この部分の演算量は極めて少ない。もとの行列 $\mathbf{A}$ は変形を受けず、 $\mathbf{A}$ とベクトルの積だけの計算が行われる。漸化関係により、逐次に求めて行く事が出来る。但し、この種の解法は、計算が進むと丸めの誤差の累積によって $\mathbf{q}_k$ の直交性が崩れる事があり、大規模計算の数値解においては、次の研究テーマになっている。

(7) サブスペース法 (Subspace Method)

$\mathbf{A}$ 、 $\mathbf{B}$ を正定値として $n \times n$ 行列とする。

A と **B** が大次元で、対称疎行列の場合の一般固有値問題を次式で示す。

$$(1\cdot169) \quad \mathbf{Ax} = \lambda \mathbf{Bx}$$

この方法は固有値のうち絶対値の小さい方から m 個求める為の方法で、 m が比較的小さな数の場合に適用できる方法である。この方法は、直交化付き同時反復法である。

応用領域として、教育の為の数値計算を始めとして、教育用の研究、産業界全般の数値計算の、固有値問題、構造解析、流体解析、基礎物理等の固有値問題のあらゆる分野に多く見受けられる。基本的な方法である。最近では、生命科学や分子科学分野にも見受けられる。1990 年代以降の標準固有値問題の解法は、このサブスペース法が使われている事が多い。現在においても使われている。サブスペース法として、標準固有値問題を解くにおいて重要な位置づけにある。

サブスペース法の計算ステップ

求めたい固有値の m 個に対応して、 m 個の初期ベクトル $\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, \dots, \mathbf{x}_0^{(m)}$ を適当に選び、

$\mathbf{x}_0^{(k)}$ を第 k 列に持つ $n \times m$ 初期行列

$$(1\cdot170) \quad \mathbf{X}_0 = [\mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, \dots, \mathbf{x}_0^{(m)}]$$

を作る。 $\mathbf{X}_0$ を出発の行列として、 $k=1, 2, \dots$ に対して

$$(1\cdot171) \quad \mathbf{X}_k = \mathbf{A}^{-1} \mathbf{B} \mathbf{X}_{k-1}$$

なる同時反復を行うと、べき乗法の原理により、 k が大きくなる時 $\mathbf{X}_k$ の各列は全て式 (1・169) の最小固有値に属する固有ベクトルに収束する。式 (1・169) は

$$(1\cdot172) \quad \mathbf{A}^{-1} \mathbf{B} \mathbf{x} = \frac{1}{\lambda} \mathbf{x}$$

と書ける。

ここで 1 回の反復毎に $\mathbf{X}_k$ に適当な直交化を行うと、最終的に $\mathbf{X}_k$ の各列に絶対値の小さい方から m 番目迄の固有値に属する固有ベクトルが順に並ぶことになる。直交化は、 m 個の固有値属する m 個の固有ベクトルが張る m 次元の部分空間(subspace)の中で行うことになる。

初期行列 $\mathbf{X}_0$ を出発値とする。

$$(1\cdot173) \quad \tilde{\mathbf{X}}_k = \mathbf{A}^{-1} \mathbf{B} \mathbf{X}_{k-1}$$

$\tilde{\mathbf{X}}_k$ が求められる度に

$$(1\cdot174) \quad \begin{aligned} \tilde{\mathbf{A}}_k &= \tilde{\mathbf{X}}_k^T \mathbf{A} \tilde{\mathbf{X}}_k \\ \tilde{\mathbf{B}}_k &= \tilde{\mathbf{X}}_k^T \mathbf{B} \tilde{\mathbf{X}}_k \end{aligned}$$

を計算し、式 (1・169) に対応して

$$(1\cdot175) \quad \tilde{\mathbf{A}}_k \mathbf{q}_k = \lambda_k \tilde{\mathbf{B}}_k \mathbf{q}_k$$

なる固有値問題を解く。式 (1・175) の固有値 $\lambda_k^{(1)}, \lambda_k^{(2)}, \dots, \lambda_k^{(m)}$ 及びそれらに属する固有ベクトル $\mathbf{q}_k^{(1)}, \mathbf{q}_k^{(2)}, \dots, \mathbf{q}_k^{(m)}$ を求める。

式 (1・175) の固有値問題は

$$(1\cdot176) \quad \mathbf{A}(\tilde{\mathbf{X}}_k \mathbf{q}_k) = \lambda_k \mathbf{B}(\tilde{\mathbf{X}}_k \mathbf{q}_k)$$

と表現する事が出来る。

従って $\mathbf{q}_k$ をもとの式 (1・169) の解へ戻すには

$$(1\cdot177) \quad \mathbf{x}_k = \tilde{\mathbf{X}}_k \mathbf{q}_k$$

すなわち

$$(1\cdot178) \quad \mathbf{X}_k = \tilde{\mathbf{X}}_k \mathbf{Q}_k$$

を計算することになる。

但し、 $\mathbf{Q}_k$ は式 (1・175) の第 i 番目の固有値 $\lambda_k^{(i)}$ に属する固有ベクトル $\mathbf{q}_k^{(i)}$ を第 i 列に持つ $n \times m$ 行列である。式 (1・169) は固有値問題をこの射影したものになっている。固有値問題をこの射影した部分空間で解く意味で、サブスペース法と称している。

式 (1・173) を計算する時、

$$(1\cdot179) \quad \mathbf{Y} = \mathbf{B}\mathbf{X}_{k-1}$$

を計算して、次に

$$(1\cdot180) \quad \mathbf{A}\tilde{\mathbf{X}}_k = \mathbf{Y}$$

を解く。この解法では、 $\mathbf{A}$ 及び $\mathbf{B}$ が、疎であることを維持して解く事が出来る長所がある。

1・8．数値計算最近の潮流、前処理法に伴う考察

一般的に、行列を例えば CG 法で解法するより、一定の形で前処理(Preconditioning)した方が、計算効率が優れていることが知られている。数値計算における最近の潮流として、前処理法が多く研究されている。そこでこの節では、数値計算の最近の潮流として、前処理に対する代表的な流儀を紹介したい。併せて、その動向や効率的とされる考え方を述べる。

[1] 先端的現代理論の解法

数値計算最近の潮流は、現代的な反復解法の中で、前処理としての対処に仕方は、元の行列の不完全分解の形を中心としている。先端的現代理論では、不完全分解だけではなく、元の行列の対角部分に着目を置いている点がある。この対角部分を分離分解の中に採り入れたもので、結果的にこの時の条件数は、より 1 に近づいた形を作ることにある。Meijerink 流では残差を引く事等により、対角の効果を挙げ、Gustafsson 流では更にパラメータを追加している。HAYAMI 流では、元の行列における対角行列の平方根を活用している。DOI-HARADA 流では、行列を 3 つに分割した後、元の行列の対角行列を活用して効果を挙げている。更にグラムシュミットの直交性の特徴を活かして、多項

式を施すことにより残差の収束性が高める van der Vorst 流の考え等が提案されている。こうした対角行列の有効な活用や、空間直交性の有効な活用により、高速解法への大きな飛躍があったと考える事が出来る。

(1) 前処理法の代表とその特徴

この説明文の多くは、既述した 1・7. 反復法、計算のプロセスで述べているが、最近の潮流を紹介する上で、再度説明する。

式 (1・1) において二次元問題の 5 点差分、又は三次元問題の 7 点差分によって得られる対称正定値行列 $\mathbf{A}$ に対して、前処理として不完全なコレスキー分解したものを ICCG 法と呼ばれる。対称正定値行列 $\mathbf{A}$ に対して、完全 $\mathbf{U}^T \mathbf{D} \mathbf{U}$ 分解の際に現れる fill-in 位置を 0 にするか、0 でない近似値を入れることにより、別なアルゴリズムが得られる。これには MICCG 法等がある。不完全コレスキー分解を行う考えである一般的な PCG 法を採り上げる。対称正定値行列 $\mathbf{A}$ に対して、前処理として不完全コレスキー分解したものを考える。

$\mathbf{A}$ に対して、 $\mathbf{A} = \mathbf{U}^T \mathbf{U} + \mathbf{N}$ 又は $\mathbf{A} = \mathbf{U}^T \mathbf{D} \mathbf{U} + \mathbf{N}$ の不完全コレスキー分解をする。計算手順として、

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \quad \mathbf{q} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{r}_0, \quad \mathbf{p}_0 = \mathbf{q}$$

となる。

ICCG 法は $(\mathbf{U}^T \mathbf{U})^{-1}$ であるが、この時の $(\mathbf{U}^T \mathbf{U})^{-1}$ の代わりに $(\mathbf{U}^T \mathbf{D} \mathbf{U})^{-1}$ とおけば MICCG 法になる。

[CR 法との相違]

CG 法に比べて、CR 法はどのような点が特徴であるかを次に説明する。行列 $\mathbf{A}$ の不完全 LU 分解 $\mathbf{A} = \mathbf{L} \mathbf{U} + \mathbf{R}$ の後で、BCG 法を適用する上で、Modification された不完全 LU 分解の共役残差法系である MILUCR 法の時と同じく、7 点差分の時には MILUBCG 法を検討する。対称でない行列に対しては、BCG 法と (双対共役勾配法) CGS 法 (自乗共役勾配法) とがある。BCG 法は汎用性が高く、安定している。しかし MILUCR 法の収束が順調なときに比べて、やや収束速度が劣る事があり得る。双対共役勾配法の BCG 法系は SOR 法が順調に収束する時、この SOR 法に比べても 3 倍は速い。不完全 LU 分解付き共役残差法は ILUCR 法で表現できる。

$\mathbf{A}$ が非対称行列の場合、これを不完全 LU 分解 $\mathbf{A} = \mathbf{L} \mathbf{U} + \mathbf{R}$ した後、共役残差法を適用する。

$\mathbf{A}\mathbf{x} = \mathbf{b}$ の両辺に $(\mathbf{L}\mathbf{U})^{-1}$ を乗じて、 $(\mathbf{L}\mathbf{U})^{-1} \mathbf{A}\mathbf{x} = (\mathbf{L}\mathbf{U})^{-1} \mathbf{b}$ を得る。この $(\mathbf{L}\mathbf{U})^{-1} \mathbf{A}$ はより単位行列に近くなっており、共役残差法は収束性が良い事が予想される。この形から CR 法が収束するためには $\mathbf{A}$ の対称部分 $\mathbf{S} = (\mathbf{A} + \mathbf{A}^T) / 2$ が正定値になる必要がある。

双対共役勾配法は、連立一次方程式 (1・1) に対して、次の双対な式を組み合わせること

$$(1 \cdot 181) \quad \mathbf{A}^T \mathbf{x}^* = \mathbf{b}^*$$

で求められる。

ここで BCG 法の計算ステップが、

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \quad , \quad \alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / (\mathbf{A}\mathbf{p}_k, \mathbf{p}_k^*)$$

と進めるのに対して、ILU の前処理を特徴とする ILUBCG 法は計算ステップが、

$$\mathbf{r}_0 = (\mathbf{LU})^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0) \quad , \quad \alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / ((\mathbf{LU})^{-1}\mathbf{A}\mathbf{p}_k, \mathbf{p}_k^*)$$

とおく事で求められる。

不完全 LU 分解のアルゴリズムは次の形となる。

$$\mathbf{G}_A = \{ (i, j) ; a_{i,j} \neq 0 \}$$

として、 $\mathbf{G} \supseteq \mathbf{G}_A$ なる集合 $\mathbf{G}$ を決めておき、 $\mathbf{A}$ を LU 分解するとき、 $\mathbf{L}$, $\mathbf{U}$ の要素中に属する場所のものだけ計算し、他のものを 0 にする方法である。

計算方法

$k = 1, 2, 3, \dots$

$j = 1, 2, 3, \dots, k-1$

if $(k, j) \in \mathbf{G}$ then

$$l_{kj} = (a_{kj} - \sum_{l=1}^{j-1} l_{kl} u_{lj}) / u_{jj}$$

$$l_{kk} = 1$$

$$u_{kk} = a_{kk} - \sum_{l=1}^{k-1} l_{kl} u_{lk}$$

$j = k+1, n$

if $(k, j) \in \mathbf{G}$ then

$$u_{kj} = a_{kj} - \sum_{l=1}^{k-1} l_{kl} u_{lj}$$

continue

(2) Meijerink 流

一般的な前処理としての考え方は、 $\mathbf{A}$ に対して $\mathbf{A}$ に関する上三角行列（下三角行列）を採り上げ、この転値及び逆行列を $\mathbf{A}$ の前後から掛ける事により、 $\mathbf{A}$ の条件数を改良する所にある。Meijerink 流とは、ここで $\mathbf{A}$ に近い正値対称行列 $\mathbf{M}$ を選ぶ。この選定の仕方は幾つか考えられるが、ここで前処理としての一般の考えは、この $\mathbf{M}$ には $\mathbf{A}$ の $\mathbf{U}$ を用いて、 $\mathbf{M} = \mathbf{U}^T \mathbf{U}$ とした $\mathbf{U}^T \mathbf{U}$ のコレスキー分解を施す点である。この方式により、前処理法としての一層の効率化を実現している[41]。

応用領域として、教育用の数値計算、教育用の一般研究、1990 年代後半以降の流体解析を始め、基礎物理、基礎化学や、宇宙航空、電気電子、電磁界解析あたりに、応用事例が見受けられる。

Meijerink 流の考え方[42]

もとの方程式に両辺から $\mathbf{U}^{-T}$ を掛けると

$$\mathbf{U}^{-T} \mathbf{A} \mathbf{x} = \mathbf{U}^{-T} \mathbf{b} \quad , \quad \tilde{\mathbf{x}} = \mathbf{U} \mathbf{x} \quad \text{とおけば、}$$

$$\mathbf{U}^{-T} \mathbf{A} \mathbf{U}^{-1} \tilde{\mathbf{x}} = \mathbf{U}^{-T} \mathbf{b} \quad , \quad \mathbf{U}^{-T} \mathbf{A} \mathbf{U}^{-1} = \tilde{\mathbf{A}}$$

この時、 $\mathbf{U}^{-T}\mathbf{A}\mathbf{U}^{-1}$ ($=\tilde{\mathbf{A}}$) の固有値は $\mathbf{A}$ に比較して 1 に近くなる。

Meijerink は $\mathbf{U}^T\mathbf{U}$ ではなく、 $\mathbf{A}$ 自体の特徴を保有するこの対角行列を取り入れた $\mathbf{U}^T\mathbf{D}\mathbf{U}$ のコレスキー分解に注目した。この対角行列 $\mathbf{D}$ を取り入れることにより、前処理を更に計算良くする方法である。つまり固有値は $\mathbf{U}^T\mathbf{U}$ だけの時より、ある行列を掛ける事で更に 1 に近くなる事が分かる。このことは幾何学的にも証明出来る。

算法の基礎として、添え字式で表現すると

$$\begin{aligned} a_{ij} &= u_{ii} + \sum_{k=1}^{i-1} u_{ki} d_k u_{kj} && \text{で計算され、} \\ d_i &= u_{ii}^{-1} && \text{つまり、} u_{ii} d_i = 1 \text{ となる値にした。} \\ G &\supseteq \{(i, j) ; a_{ij} \neq 0\} \end{aligned}$$

の様な格子点集合 G を指定して、 $(i, j) \in G$ の時だけ、 u_{ij} を作る形を提案出来る。これが ICCG 法である。

$$\begin{aligned} (1.182) \quad d_i^{-1} &= a_i - b_{i-1}^2 d_{i-1} - c_{i-1}^2 d_{i-n} - e_{i-mn}^2 d_{i-mn} \\ a_{ij} &= u_{ii} + \sum_{k=1}^{i-1} u_{ki} d_k u_{kj} \end{aligned}$$

で計算される。

ここで式 (1.182) における添え字で、 $\mathbf{A}$ に Meijerink 流による不完全ガウスの表現を $\mathbf{D}$ に近い $\tilde{\mathbf{D}}$ を用いて、次の形で定義する。

$$(1.183) \quad \mathbf{A} = \mathbf{L}\tilde{\mathbf{D}}\mathbf{U} - \mathbf{R}$$

ここで式 (1.183) から近似される $\mathbf{L}\mathbf{D}\mathbf{U}$ を使用して、不完全 $\mathbf{L}\mathbf{D}\mathbf{U}$ の逆を両辺から掛ける。

$$(1.184) \quad (\mathbf{L}\mathbf{D}\mathbf{U})^{-1} \mathbf{A} \mathbf{x} = (\mathbf{L}\mathbf{D}\mathbf{U})^{-1} \mathbf{b}$$

式 (1.184) が、Meijerink 流の考え方となる。この Meijerink 流は、例えば 7 点差分の行列の場合、

$\mathbf{A}$ の第 i 行を左から順に、 a_i, b_i, c_i, d_i (対角) e_i, f_i, g_i として、

$$(1.185) \quad \tilde{d}_i^{-1} = d_i - a_i g_{i-mn} \tilde{d}_{i-mn} - b_i f_{i-m} \tilde{d}_{i-m} - c_i e_{i-1} \tilde{d}_{i-1}$$

とする形である。これに対して、BCG 法や CR 法を適用する。式 (1.185) (1.168) に対して BCG 法に適用したものが ILUBCG 法であり、CR 法を適用したものが ILUCR 法と呼んでいる。この場合一般に BCG 法は汎用性に富み安定解が得られる。これに対して CR 法の収束スピードは速い事が知られているが、モデルによって常に安定解が得られるとは限らない点が上げられる。ここで実用面では、プログラミング技術の優劣で効果が左右される。たとえばこの分野の研究で、van der Vorst 流 (特殊なケースは高速化が出来る考え) に基づくと、多くの解法には適さない点のある事が指摘されている。一般の考えではベクトル計算機に適したループ長を長くする方法で使われている。この Meijerink 流の考え方は、プログラミング技術に気を付ければスムーズに収束する事が多く、旧来の上三角行列に着目した前処理法から大きく前進し、式 (1.1) にみる一般的モデルの場合では、高速数値解法がより可能になった点で、数値解法の考えは飛躍し、優れた流儀であると賛美出来る。

(3) Gustafsson 流

Meijerink の数値計算の僅かな誤差を差分式の計算過程で、小さな正数を負荷させる事で計算の正確性を追求し、合わせて各回の反復でより残差を少なくする事で、少ない反復回数で収束させる事を目的にしている。例えば、 $\mathbf{U}^T \mathbf{D} \mathbf{U}$ を計算し直すと、数値計算上の丸めが加わり、完全にもとの $\mathbf{A}$ にはならない事が分かっている。 $\mathbf{U}^T \mathbf{D} \mathbf{U}$ 不完全分解の不完全性があり、これを改良する方法として、考え出された。式 (1・185) の d_i^{-1} の右边が負やゼロになって、不完全コレスキー分解が出来なくなった場合、 a_i の係数に正数を付加するなり、 $0 < u < 1$ なる u を用いて、付加項に掛けて不完全 $\mathbf{U}^T \mathbf{D} \mathbf{U}$ を作る。こうした方法で Meijerink の数値計算を改良した。

応用領域として、Meijerink 流とほぼ同じく、教育用の数値計算、教育用の一般研究、1990 年代後半以降の流体解析を始め、基礎物理、基礎化学や、宇宙航空、電気電子、電磁界解析あたりに、応用事例が見受けられる。

Gustafsson 流の考え方[37]

Gustafsson は式 (1・185) において以下のパラメータを導入し、高速計算の実現を目指した。

$$(1 \cdot 186) \quad \begin{aligned} \tilde{d}_i^{-1} = & (1 + \varepsilon) d_i - a_i g_{i-mn} \tilde{d}_{i-mn} - b_i f_{i-m} \tilde{d}_{i-m} - c_i e_{i-1} \tilde{d}_{i-1} \\ & - u \left\{ a_i \tilde{d}_{i-mn} (e_{i-mn} + f_{i-mn}) + b_i \tilde{d}_{i-m} (e_{i-m} + g_{i-m}) + c_i \tilde{d}_{i-1} (f_{i-1} + g_{i-1}) \right\} \end{aligned}$$

式 (1・186) において ε , u は Gustafsson 流では安定化の為のパラメータとして定義している。この Gustafsson 流の効果は不完全 LU 分解の安定化の為のパラメータとして、例えば流体の場合のセレペクレ数に依存した場合や、風上差分のパラメータに依存させた場合に選び方によるが、Gustafsson 項の値によって、著しい効果が現れる場合がある。Gustafsson としての $\mathbf{M}$ 行列はパラメータの設定により、優れたものを選ぶ必要がある。

但し、選び方を誤ると、十分な効果が出ない場合がある。例えば、流体解析の場合は、離散化を風上差分で構成すると、セレペクレ数の値によりそれまで効果的であった ε , u の値が通じない場合が多く報告されている。移流拡散方程式の解法に対しては十分な研究が尽くされていない点があり、未だ研究途上である。拡散形の解法においては、数式モデルにもよるが、式 (1・1) を解く上で Multigrid の前処理 (小柳理論) より、全てが優れていると言う訳ではないが、一般的に高速化がもたらされる事が多くの数値シミュレーションにより判明している。Gustafsson の $\mathbf{M}$ 行列を用いる事で数値解法として安定性が向上し、有望であり、優れた性能を発揮する事が出来る。これは Gustafsson 効果と呼ばれている。Meijerink 流の ILUBCG 法や ILUCR 法に対して、Gustafsson 流は $\mathbf{M}$ を付けた MILUBCG 法や MILUCR 法と名付けられ、その効率的な演算処理速度により普及している。

(4) HAYAMI 流

この方法は CG 法の中でも、特に対角行列に着目した考えである。HAYAMI は不完全コレスキー分解の前処理を検討して行く上で、その計算ステップで毎回の反復における前進後退代入である

$(\mathbf{LDL}^{-T})^{-1} \mathbf{r}$ の計算がベクトル計算機の高速度性を活かすことを阻んでいるのではないかと、仮定

をたてて検証した。これには例えば、不完全コレスキー分解である ICCG 法は、それ自体計算は速いが、ベクトル計算機特有の特徴を活かす点で、ベクトル化率が低くなるものに対しては、一考を要するものであった。有限差分法では差分形式が 2 次元から 3 次元になるに従い、ベクトル化率が低くなる、そして加速率も低くなる。ICCG 法を使用する場合、こうした点を数値実験で検証し、問題として認識していた。

HAYAMI は固有値となる対角行列に着目し、この対角となる行列をあらかじめピックアップして改良した対角として $\mathbf{A}$ の前後から掛ける事で、条件数を大幅に改善しベクトル計算機に合わせた高速計算をさせる事に成功した。対角行列に着目し、高速計算させる上での大発見である。行列の解法には最大固有値と最小固有値の比が近づいた場合、つまり条件数が 1 に近づいた場合に行列の高速解法が期待される事が分かっている。

応用領域として、教育用の数値計算、教育用の一般研究、2000 年代以降の流体解析、核融合、宇宙科学、航空工学分野に幾つかの事例を見受ける事が出来る。

HAYAMI 流の考え方

$\mathbf{A}$ を対称正定値行列とする。対角が大きく優位である行列の場合、大幅な改善が期待出来る。

$\mathbf{A}$ の左右から $\mathbf{A}$ をもとにした一つの行列を掛け、条件数を改良する事をスケーリングと定義する。

$\mathbf{A}$ をスケーリングするにおいて、 $\mathbf{A}$ の対角行列に着目する。対角行列を $\mathbf{D}$ とおくと、ここで $\mathbf{D}^{-\frac{1}{2}}$ を定義し、 $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} (= \tilde{\mathbf{A}})$ の形に変形する。この時、 $\tilde{\mathbf{A}}$ も対称正定値行列となる。

$$(1.187) \quad \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{x} = \mathbf{D}^{-\frac{1}{2}}\mathbf{b}$$

となる。

$$(1.188) \quad \tilde{\mathbf{x}} = \mathbf{D}^{\frac{1}{2}}\mathbf{x}$$

$$(1.189) \quad \tilde{\mathbf{b}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{b}$$

とおく事で、

$$(1.190) \quad \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

この時の $\tilde{\mathbf{A}}$ はもとの $\mathbf{A}$ に比べて、条件数がより 1 に近くなる。対角優位な行列で有れば、これは顕著に現れる。この考えを基本にしている。これは SCG 法と呼ばれる。速水流は対角優位行列の対角を取り上げ、この平方根で単位行列を除する行列を作り、 $\mathbf{A}$ の両辺から掛ける事で（スケーリングを行う）、最大固有値から最小固有値迄のバランスを平準化させたものと考え事が出来る。これは特徴として対称行列の高速数値解法として、大きな威力を発揮する。

Meijerink 流や Gustafsson 流が、より大規模行列になると、その計算手法の特徴から、自乗関数に近いカーブで演算時間が必要とされるのに対して、この速水流は大規模行列になっても演算時間は線形に伸びて行くため、大規模行列なるほど他の流儀よりその効果は大きい。これは SCG 法として知られている。この事は Gerschgorin の定理により幾何学的に証明する事が出来る。SCG 法は対称行列を取り扱うが、活用方法として $\mathbf{A}$ が非対称行列の場合、 $\mathbf{A}$ の転置行列を左側から掛ける事により、対称行列に置き換えてこの SCG 法を使う事が出来る。

$$(1 \cdot 191) \quad \mathbf{D}^{-1} = \begin{pmatrix} 1/a_{11} & & \\ & \ddots & \\ & & 1/a_{nn} \end{pmatrix}$$

とおく。

ここで理論的には $\mathbf{D}$ の各要素が正である任意の対角行列であれば、HAYAMI の理論に合致するが、このアルゴリズムでは $\mathbf{A}$ の対角項を採用する。ベクトル計算機有効利用の面からは、式 (1・191) にみる $\mathbf{D}^{-1}$ の計算は一次元配列におくことが、ベクトル化率を高める上で効果的である。

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, & \tilde{\mathbf{x}} &= \mathbf{D}^{\frac{1}{2}} \mathbf{x}, & \tilde{\mathbf{b}} &= \mathbf{D}^{-\frac{1}{2}} \mathbf{b} \\ \tilde{\mathbf{r}}_1 &= \tilde{\mathbf{b}} - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_1, & \tilde{\mathbf{p}}_1 &= \mathbf{D}^{-1} \tilde{\mathbf{r}}_1 \end{aligned}$$

SCG 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\tilde{\mathbf{r}}_0 = \tilde{\mathbf{b}} - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_0, \quad \tilde{\mathbf{p}}_0 = \mathbf{D}^{-1} \tilde{\mathbf{r}}_0,$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , ($k=0,1,2,\dots$)

$$\alpha_k = \frac{(\tilde{\mathbf{r}}_k, \mathbf{D}^{-1} \tilde{\mathbf{r}}_k)}{(\tilde{\mathbf{p}}_k, \tilde{\mathbf{A}} \tilde{\mathbf{p}}_k)}$$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \alpha_k \tilde{\mathbf{p}}_k$$

$$\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - \alpha_k \tilde{\mathbf{A}} \tilde{\mathbf{p}}_k$$

$$\beta_k = \frac{(\tilde{\mathbf{r}}_{k+1}, \mathbf{D}^{-1} \tilde{\mathbf{r}}_{k+1})}{(\tilde{\mathbf{r}}_k, \mathbf{D}^{-1} \tilde{\mathbf{r}}_k)}$$

$$\tilde{\mathbf{p}}_{k+1} = \mathbf{D}^{-1} \tilde{\mathbf{r}}_{k+1} + \beta_k \tilde{\mathbf{p}}_k$$

continue

(5) DOI・HARADA 流

CG 法系の前処理方法の中で並列性の計算の考えに着目したものである。土肥・原田の両名の考案による。HARADA はベクトル計算機の有効活用の点から、従来からの前処理である PBCG 法や前処理付き共役残差法系 PCR 法では不完全 LU 分解を重視するあまり、ベクトル計算機の中の計算格子に対して迄十分な考えに及んでいない事に気が付いた。これら不完全 LU 分解では計算格子に対して、斜め方向の並列性があり、アレイ形式の計算方法では計算に進行に伴って、斜め方向の並列性に変化して進んで行く事であり、この時アレイプロセッサの計算格子間に遊びが生じる問題があった。これを改良するために x 方向、y 方向と行列をこの形に分解し、正方向へ計算プロセスが進んで行く前処理としての新しい理論を提案した。これはベクトル計算機の有効利用に対して、画期的な方法を

発明し、理論を確立させる事が出来た。計算理論の概要は以下の形である。良く用いられている不完全LU分解では、計算ステップの L 、 U による前進後退代入により、計算上の並列性に問題が生じる事を発見した。

応用領域として、教育用の数値計算、教育用の一般研究、2000 年代以降の原子力・核融合問題、電磁流体や流体解析等に幾つか事例を見受ける事が出来る。

DOI・HARADA 流の考え方

2次元5点差分の場合、前進代入として

$$(1\cdot192) \quad \mathbf{v}_{i,j} = (\mathbf{g}_{i,j} - \mathbf{l}_{i,j-1} \cdot \mathbf{v}_{i,j-1} - \mathbf{l}_{i-1,j} \cdot \mathbf{v}_{i-1,j}) / \mathbf{l}_{i,j}$$

を定義する。式 (1・192) は格子座標方向 i (または j) の増加方向に計算すると、この時 $\mathbf{v}_{i-1,j}, \mathbf{v}_{i,j}$ の間に参照関係から並列処理が出来ない。しかし $i+j$ = 一定となる $\mathbf{v}_{i,j}$ 、つまり斜め方向の格子点群が同時処理できる。この時ベクトル計算機ではリストベクトルを用いる事で強制ベクトル化が行われている。しかしこのリストベクトルを用いた間接メモリアクセスは等間隔の直接メモリアクセスに比べて転送速度が遅く、ベクトル計算機の高性能を十分に出来ない事が分かっている。そこで HARADA は5点や7点差分行列において、 $\mathbf{A}$ を対角行列 $\mathbf{D}$ と x 方向微分に関する非対角要素からなる $\mathbf{A}_x$ 、 y 方向微分に関する非対角要素からなる行列に分解して $\mathbf{A}_y$ を定義した。

$$(1\cdot193) \quad \mathbf{A} = \mathbf{D} + \mathbf{A}_x + \mathbf{A}_y$$

この条件での3つの行列から構成される為、三項対角近似因子分解 (Tridiagonal Approximate Factorization Method) と名付けた。

一般的な前処理である $\mathbf{M}_{ILU}^{-1}$ は $\mathbf{L}, \mathbf{U}$ による前進消去、後退代入による計算を行う。但し $\mathbf{L}, \mathbf{U}$ は $\mathbf{A}$ の下三角、上三角と同じ非零要素パターンを持つ下三角行列、上三角行列である。この方式による計算では、ベクトル計算機はリストベクトルを用いて強制的にベクトル化する事が行われる。しかしリストベクトルを用いた間接メモリアクセスは、等間隔直接メモリアクセスに比べて転送速度が低く、ベクトル計算機の性能を十分に引き出せない点が考えられる。一般的な ILU 前処理法に対する並列性は、計算ノードのグリッドポイントが斜め方向並列計算として進む。

HARADA はベクトル計算機的高速性に向けて、以下の3点を注目して、ベクトル計算機有効利用の為の3要素と定義し数式を導いた。

① 逆行列操作の容易性

反復の前処理で因子毎に LU 分解する。

$$(1\cdot194) \quad \mathbf{M}_{TF} = \mathbf{L}_x \mathbf{U}_x \mathbf{D}^{-1} \mathbf{L}_y \mathbf{U}_y$$

これにより逆行列計算が容易に行える。

③ 並列性

近似誤差行列 $\mathbf{R}_{TF}$ を定義して、前処理行列 $\mathbf{M}_{TF}$ とおく

$$(1\cdot195) \quad \begin{aligned} \mathbf{R}_{TF} &= \mathbf{M}_{TF} - \mathbf{A} \\ &= \mathbf{A}_x \mathbf{D}^{-1} \mathbf{A}_y \end{aligned}$$

つまり前処理行列 $\mathbf{M}_{TF}$ は $\mathbf{A}$ に似た行列が望ましい点。

③ 並列性

並列計算機を特徴とする計算機では、 $\mathbf{M}_{\text{TF}}^{-1}$ が計算機に適した並列性を持つことが重要となる。
 $\mathbf{x}$ 方向因子の近似計算として、

$$(1.196) \quad (\mathbf{D} + \omega \mathbf{A}_x) \mathbf{v} = \mathbf{g}$$

を定義する。つまり式 (1.196) に見るこれは $\mathbf{A}_x$ と $\mathbf{A}_y$ とは独立に計算出来る。

HARADA は、これらを前処理としての基礎反復法の高速度性を提案した。特徴として規則、疎な非対称行列の反復法として、大きな効果をもたす事が出来、大規模な行列に対しても、効果をもたらす事が顕著である。収束の面では ILUBCG 法と同程度の有効性がある事が認められており、これら不完全 LU 分解における前処理に比べて、反復回数が少ない分だけ、ILUBCG 法や MILUBCG 法よりも高速計算出来る事が判明している。この前処理法では、将来のベクトル計算機にも適用出来る点で注目されている。

式 (1.194) より $\mathbf{M}_{\text{TF}} = \mathbf{L}_x \mathbf{U}_x \mathbf{D}^{-1} \mathbf{L}_y \mathbf{U}_y$ の $\mathbf{M}_{\text{TF}}$ を用いて

TF 法アルゴリズム

初期ベクトル $\mathbf{x}_0$, 収束判定値 eps

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0, \quad \mathbf{p}_0 = \mathbf{M}_{\text{TF}}^{-1} \mathbf{r}_0, \quad \mathbf{p}_0^* = \mathbf{r}_0^* = \mathbf{r}_0,$$

while $\|\mathbf{r}_k\| > \text{eps} * \|\mathbf{b}\|$ do , (k=0,1,2,...)

$$\alpha_{k+1} = \frac{(\mathbf{r}_k, \mathbf{r}_k^*)}{(\mathbf{A} \mathbf{p}_k, \mathbf{p}_k^*)}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{k+1} \mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_{k+1} \mathbf{A} \mathbf{p}_k, \quad \mathbf{r}_{k+1}^* = \mathbf{r}_k^* - \alpha_{k+1} (\mathbf{M}_{\text{TF}}^{-1})^T \mathbf{A}^T \mathbf{p}_k^*$$

$$\beta_{k+1} = \frac{(\mathbf{r}_{k+1}, \mathbf{r}_{k+1}^*)}{(\mathbf{r}_k, \mathbf{r}_k^*)}$$

$$\mathbf{p}_{k+1} = \mathbf{M}_{\text{TF}}^{-1} \mathbf{r}_{k+1} + \beta_{k+1} \mathbf{p}_k, \quad \mathbf{p}_{k+1}^* = \mathbf{r}_{k+1}^* + \beta_{k+1} \mathbf{p}_{k+1}^*$$

continue

(6) DOI 流

Odd-Even SOR 法にみる多色オーダリングの考えの拡張である。データを並び替え出来る限り長いベクトル長を作り、計算の効率を目指す解法である。DOI は多色 ILU 法の考えとして、係数行列 $\mathbf{A}$ を構成する際の格子点の並びを多色 Ordering に従い変更する事により前進後退代入での並列性が増す考えで技術面から追求した。ベクトル計算機の特長を活かす上で、データの並び替えを行い、高速計算を実現させた。

応用領域として、教育用の数値計算、教育用の一般研究等に一部存在している。どのような分野に

適用させるか今後の研究課題とされている。

DOI 流の考え方

多色 Ordering では色の数を nc とするとき、3次元格子点の部分集合 $S_1, S_2, \dots, S_{nc}$ を $S_l = \{ (i, j, k) \mid \text{mod}(i+j+k-3, nc)+1=1 \}$ とおき、格子点を S_1 に含まれるものから順に S_{nc} をかけて順序付けする。

この Ordering に従い係数行列 A を構成すると、下記ブロック対角部 D_l ($l=1, 2, \dots, nc$) が対角行列になるため前進後退代入の計算が各色で並列に実行可能になる

多色 Ordering により構成される係数行列

$$\left[\begin{array}{cccccccc} D_1 & U_{1,2} & 0 & 0 & 0 & \cdot & \cdot & U_{1,nc} \\ L_{2,1} & D_2 & U_{2,3} & 0 & 0 & \cdot & \cdot & 0 \\ 0 & L_{3,2} & D_3 & & & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & & & \cdot & 0 \\ 0 & 0 & 0 & 0 & \cdot & & D_{nc-1} & U_{nc-1,nc} \\ L_{nc,1} & 0 & 0 & 0 & \cdot & \cdot & U_{nc,nc-1} & D_{nc} \end{array} \right] \left| \begin{array}{l} \langle S_1 \\ \langle S_2 \\ \langle S_3 \\ \cdot \\ \cdot \\ \cdot \\ \langle S_{nc-1} \\ \langle S_{nc} \end{array} \right.$$

多色 ILU 法における前処理は前述の標準 ILU 前処理法と異なる結果を与え、一般に収束するまでの反復数が増加する。異方性の大きな問題やグフタフソン加速手法を用いるときは、この現象が顕著になる。多色 ILU 法において、グフタフソン加速手法では、前処理の高速性が打ち消される場合があり得る。

重複多色 ILU 前処理法による前処理は一部で重複して前進後退代入を行い、収束性の悪化を防ぐものである。一般にグフタフソン加速を行った場合も標準 ILU 前処理手法と同様の収束性を示す。重複度が大きくなるほど計算量は増えるが、収束性は良くなる。重複度の既定値を加速パラメータが 0.1 以下の時には 5 に加速パラメータが 0.1 以上の時には 10 に設定する。すべての色の数 $n_x - 1$ に比べて重複度が小さい場合重複によりオーバーヘッドは非常に小さくなる。

3次元7点差分用では、前進後退代入の高速化を目指して色の数を $n_x - 1$ に固定し、係数行列 A 初期値 u_0 右辺 b をサブルーチン内部で多色 Ordering の順に並び替え A , b は終了時もとに戻す。これによりベクトル計算がメモリへの連続アクセスのみで行える用にするバンク衝突を気にする必要がない。但し、並び替えた後のデータ領域のサイズが $n = n_x(n_y + 1)n_z$ となるのでものとサイズもこれ以上に確保しておく必要がある。

多色 Ordering では標準的なデータ並びで与えられた各ベクトルに対応する配列 $(u_i)_{1 \leq i \leq n}$ も作業配列 $(w_i)_{1 \leq i \leq n}$ を用いて次のように並び替える。

$$\begin{array}{llll} i_z = 1 & \text{から} & n_z & \text{に対して} \\ i = 1 & \text{から} & n_{xy} & \text{に対して} \\ & & w_{(nxy+inc)(iz-1)+1} = v_{nxy}(i_z - 1) + 1 & \\ i_x = 1 & \text{から} & n_x - 1 & \text{に対して} \\ j = 1 & \text{から} & mc & \text{に対して} \end{array}$$

$$V_{mc(ix-1)+1} = W_{ix+(ix-1)(nx-1)}$$

ここで $n_{xy} = n_x \cdot n_y$ であり、 inc は $\text{mod}(n_x + inc - 1, n_x - 1) = 0$ を満たす最小の非負整数である。最初の $\mathbf{w}$ への入れ替えにより各成分が $n_x - 1$ の間隔で並ぶ。 mc は一つの色に含まれる成分の個数の最大値でありもとの配列 $\mathbf{v}$ に $\mathbf{S}_1$ から $\mathbf{S}_{n_x-1}$ へかけて各色の成分が連続に並ぶ様に入れ替える。なお、この並び替え手法は各ベクトル変数に付き $mc \cdot (n_x - 1)$ 以上の実数領域が必要であり、このためには $\mathbf{n} = n_x(n_y + 1)n_z$ としておけば十分である。

(7) Axelsson 流

アクレソン[65]は、行列 $\mathbf{A}$ を対角行列、下三角行列、上三角行列と分けて、下三角、上三角にそれぞれパラメータを負荷し、更に対角行列を加えることによって、加速を伴って前処理が有効に働く事を確認した[8]。

応用領域として、教育用の数値計算、教育用の一般研究として用いられている。

Axelsson 流の考え方[66]

$\mathbf{A}$ の対角要素、下三角要素、上下三角要素からなる、対角行列、下三角行列、上三角行列を採り上げる。

SSOR 前処理行列 $\mathbf{M}_{\text{SSOR}}$ は加速パラメータ ω ($0 \leq \omega \leq 2$) を用いて

$$(1.197) \quad \mathbf{M}_{\text{SSOR}} = (\mathbf{A}_D + \omega \mathbf{A}_L) \mathbf{A}_D^{-1} (\mathbf{A}_D + \omega \mathbf{A}_U)$$

で定義される。

特徴： $\mathbf{A}_D + \omega \mathbf{A}_L$ 、 $\mathbf{A}_D^{-1}$ 、 $\mathbf{A}_D + \omega \mathbf{A}_U$ は ILU 行列の $\mathbf{L}$ 、 $\mathbf{D}$ 、 $\mathbf{U}$ それぞれ同じ零／非零パターンを持ち、行列反転の手順は同じである。 ω は通常 1.6 ～ 1.8 が最適値であり高速性が発揮出来る。次に詳細な資料がある。

Axelsson, O., "Solution of Linear Systems of Equations", Lecture Notes in Mathematics, 572, Springer-Verlag pp.1-51.

(8) van der Vorst 流

行列の次数が大きくなると、本質的には特性方程式が複雑になり、数値計算が困難になる事が知られている。この高次方程式に対して、より効率的な収束条件を追求する必要性から考え出された。クリロフ部分空間 $\text{Span} \{ \mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{n-1}\mathbf{r}_0 \}$ を定義し、その中で Hermite 行列に対して、ガレルキン条件 (Galerkin condition) $\mathbf{r}_{n+1} \perp \mathbf{K}_{n+1}(\mathbf{A} : \mathbf{r}_0)$ を満たす残差を導く。

この時、クリロフ部分空間の直交系となり、この残差にグラムシュミットの直交化法 (Gram-Schmidt orthogonalization) を施す。これにランチョス・プロセスを施すこれにランチョス・プロセスを施すことにより 3 項漸化式が導かれる。

応用領域として、前処理法の一般教育、一般的な研究用を始めとして、2000 年代以降の構造解析や流体解析、基礎物理問題、地球科学、海洋科学分野で一部の使用事例が見受けられる。

van der Vorst 流の考え方

次の形の3項漸化式を定義する。

$$(1\cdot198) \quad \mathbf{v}_{n+1} = \alpha_n \left\{ \frac{(\mathbf{A}\mathbf{v}_n, \mathbf{v}_n)}{(\mathbf{v}_n, \mathbf{v}_n)} \mathbf{v}_n + \frac{(\mathbf{A}\mathbf{v}_{n-1}, \mathbf{v}_n)}{(\mathbf{v}_{n-1}, \mathbf{v}_{n-1})} \mathbf{v}_{n-1} - \mathbf{A}\mathbf{v}_n \right\}$$

これらからグラムシュミットの直交性に基づいて、双共役勾配法を形成させる。それ自体にランチョス多項式を施すことにより残差の収束性が高くなる事が知られている。つまりランチョス法を施すことにより初期残差の積で表現され、厳密性も満たされる事が可能になっている。ランチョス多項式では $\mathbf{r}_n \cong \mathbf{R}_n(\mathbf{A})\mathbf{R}_n(\mathbf{A})\mathbf{r}_0$ とした2つの多項式の積の形をとる事で、この関数の性質に依存される事になる。その選び方により、BCG法、CGS法、BCGSTAB法、GPBCG法があげられる。

非対称モデルに対して、BCG法は双対な方程式を組み合わせた2n元連立一次方程式を定義する。この解法は非対称行列に対する安定した解法である。多くの非対称行列にスムーズに求解することが知られている。これに対してCGS法は残差の効率的収束の追求から研究された解法である。高い収束性が期待されるが、モデルにより不規則な収束性が見られる場合があり、正確な反復計算が行えないことがある。このCGS法の欠点を回避し、比較的高速で安定収束を保持させる様に工夫したものがBCGSTAB法である。

特徴：残差への理論構成が良好な時、クリロフ部分空間解法には残差の最小条件と直交条件を用いる。ランチョス・多項式法を施すことにより、効率的な収束条件を満たした3項漸化式が得られる。これはベクトル計算に適しており数値解法的高速性が実現する。

(9) Jacobi-Davidson 流

大規模疎行列の固有値計算は Lanczos/Arnoldi 系の方法が優れている。しかしこの方法では、固有値間の分離が十分でない場合、固有値を正確に計算させる事に困難が付きまとう。

固有値問題 $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ に対して、行列 $\mathbf{A}$ の Ritz 値 θ_k に対応する近似固有ベクトル $\mathbf{u}_k$ の直交補空間 $\mathbf{u}_k^\perp$ を定義する。これにより計算のスムーズな流れを定義した。

応用領域として、前処理法の一般教育、一般的な研究用を始めとして、2000年代以降の流体解析に一部使用例が見受けられる。

Jacobi-Davidson 流の考え方

行列 $\mathbf{A}$ の $\mathbf{u}_k^\perp$ への直交射影を

$$(1\cdot199) \quad \mathbf{A}_p = (\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*) \mathbf{A} (\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*)$$

で表現する。

ここで修正ベクトルを $\mathbf{z}$ 、残差ベクトルを $\mathbf{r}$ と置くことで、式 (1・199) は $(\mathbf{A}_p - \lambda \mathbf{I}) \mathbf{z} = -\mathbf{r}$ の計算に帰着される。この計算を行う前に、近似解が直交する空間に限定される性質を利用して次の近似演算子

$$(1\cdot200) \quad \tilde{\mathbf{M}}_k = (\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*) \mathbf{M}_k (\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*)$$

を用いて、左から掛ける事を強調した左前処理として $\tilde{\mathbf{M}}_k \tilde{\mathbf{A}}$ を用いる。

ここで行列 $\mathbf{A}$ の部分 Schur 形を正規直交基底から射影行列を定義する事で、固有値を昇順に並べる事が容易に出来、複数個の固有値計算を高速で求解する事が出来る。

特徴：固有値解法として著名な QR 法に対して、並列性の面から一步凌駕した解法である。大規模

疎行列の解法として有望である。

1-9. 数値計算の高速化への発展

連立一次方程式を解く上で、数値計算の一般論を述べると、古典的な直接法（代表ガウス消去法）より、古典的な反復法（代表ヤコビ法）の方が計算速度は速く、このヤコビ法よりヤコビの改良型であるガウス-ザイデル法の方が速く収束する事が知られており、このガウス-ザイデル法より、ガウス-ザイデル法に加速パラメータを加えた近代的反復法である SOR 法（パラメータの取り方による）の法が速く収束し、一方で物理モデル等では ADI 法の方が速く収束する事が知られている。更にモデルにもよるが Stone 法等も安定高速収束する事が知られている。この SOR 法より多項式概念を組み入れたチェビシェフ法の方が速く収束する事が知られている。このチェビシェフ法は準反復法とも呼ばれ、SOR 法と同じく加速パラメータを効率的に用いて、その高速性が左右される解法である。これらの処理速度の倍率は、取り扱う行列の性質によりいろいろ変わる。更に行列の性質は制限されるが、このチェビシェフ法より CG 法の方が速く収束する事が知られている。CG 法はいろいろな系統があり、中でも最近では前処理をした CG 法の研究が盛んに行われている。前処理付き CG 法として高速性への追求はいろいろの研究事例がある。前処理をした CG 法の中で高速性をもたらす ILUBCG 法を採り上げて、バンド型に当てはめ検証すると、こうした中で ILUBCG 法が古典的な直接法（代表ガウス消去法）の 10 倍の処理速度というのもおよそ妥当なものと考えられる。

一般論で述べると、処理速度の比較において、代表的な数値解法の中では概念的に次の形となる。

＜ 印 はより速い処理速度を表す。

ガウス法＜ヤコビ法＜ガウスザイデル法＜SOR 法

SOR 法＜チェビシェフ法＜CG 法＜前処理付 CG 法＜最近の研究

一部の物理モデル SOR 法＜ADI 法

一部の物理モデル SOR 法＜Stone 法

この形が分かっているならば、処理速度の速い前処理付き CG 法だけが優先して使われるべきと考えがちであるが、現実の数値解法は様々な数式モデルがある。

式 (1・1) を解いて行く上で、いろいろなことが加味される。代表的なものに条件数が関わってくる。また $\mathbf{A}$ が密行列か疎行列か、更に規則か不規則か、対称か非対称か、ランダムスパースか、正則行列か、正則分離可能な行列か、優対角行列か、サイクリック行列か、バンド行列か、ブロック化行列か、ヘッセンベルグ行列か、フェルミオン行列か、トエプリッツ行列か、フランク行列か、エルミート行列となるか、スティルチェス条件を満たすか、その他により解法の長短が出てくる。最近の高速数値解法の研究の多くは、これらのある部分が整った条件のもとで仮定し、理論式を組み立てている。つまり式 (1・1) を解く上で、こうした条件がたまたまうまく適合する場合と、中にはうまく適合しない場合があり得る。こうした中で解法を選んだ場合、例えば $\mathbf{A}$ の固有値分布が大きくばらつく場合、前処理付き CG 法の反復法を用いた場合では、数値解がうまく求められない事が知られている。つまり高速数値解法の各種類には、式 (1・1) に対する適材適所が必要とされている。

一方直接法にもいろいろあり、ベクトル計算機の発達と共に、各種高速計算の研究提案がされている。これを簡単に述べると、旧来の直接法（ガウス消去法他）は各計算ステップにおいて、係数行列の形は刻々と変化し、ゼロ要素に至るまで、ノンゼロ要素に置き換えて計算して行くため、中小規模な密行列に適し、理論的にも厳密解は期待できる。一般には計算時間がかかる。最近ではベクトル計

算機を意識したブロック型の直接法も提案され、計算時間を縮小した解法も幾つか提案されている。反復法は大規模な疎行列に適しており、この大規模行列は規則性を持つことが多い。不規則な行列は中規模の行列に多い。この大規模行列ではもとの行列の形を崩さないで、反復収束させるものが一般的な反復法である。この行列の性質は条件数により左右される事があり、反復法により可成りの高速性が発揮できるが、条件数が適さない（ある一定に入らない）場合、収束そのものが困難になる場合もある。

計算機の発達と共に、反復法も最近では大型行列を取り扱う場合が多く、この場合前処理を行ってから反復計算させる様になっている。この前処理は上記にみる不完全 LU 分解やコレスキー分解にターゲットが当てられた研究論文が多く出されている。不完全 LU 分解としての ILU は前処理として LU 分解の不完全性の特徴を活かし、効果的な反復法である CG 系の非対称に適する BCG 法にも適用され、優れた解法である事が多くの数値実験で確かめられている。

ILUBCG 法における最近の研究では、特に 5 点差分や 7 点差分の研究で対象としたものでは、行列のメッシュサイズが大きくなると、反復回数も多くなり、CPU 時間を食う事も知られている。つまり CPU 時間が線形に延びて行くのではなく、非線形な形で反復回数が多くなって行く事が数値シミュレーション結果判明している。最近ではベクトル計算機を効率よく活用する上で、ILU では演算過程における計算格子に対して、斜め方向の並列性を保ちながら、計算が進んで行く事が判明している。これが計算の進行に伴って、変化して行く為、プロセッサがフル稼働しない状態がある事も分かってきた。ILU 分解に伴う前進後退代入が逐次処理となる計算のステップがあり、どうしてもマシンのフル機能が発揮出来ない点も判明している。これらの点から色々な改良方法が研究されている。

偏微分方程式

2. 偏微分方程式から数値解法

付録1. で述べた科学技術計算の分類において、時間と空間の有限差分を特徴とした偏微分方程式のグループは様々な分野に存在し、その活用されている比率も相当高い。偏微分方程式は、多くの技術計算問題の定式化として活用されている[98]。数値解析研究者は、その多くが理論式を用いて、解法を展開して収束状況や安定性、効率性を目指して理論的研究論文を出している。理論式は行列の並びが滑らかであったり、正確に収束するなど都合の良いものがある。これはアルゴリズムを開発する時は、こうした事が必要であると考えられる。

これに対して、実問題では定式化は出来ても、様々な局面において行列の並びに特異性があったり、条件数が悪い場合やモデルにより固有値分布に特性があって、解法がうまく進まない場合がある。つまり、理論式でうまく解法が当てはまり、スムーズに収束出来ても、実モデルにおいては、スムーズに収束が進まない場合があり得る。

当論文では、理論面ではなく実モデルとして、原子力分野に焦点を充てて、本論第2章にみる原子力問題[38]の事例を採り上げて研究した。それは原子力コードとして、重要な位置を占める中性子拡散方程式のシュミュレーションコードである。この中性子拡散方程式は、代表的な偏微分方程式で表される為、この章においては偏微分方程式とはどのようなものであるかを説明する為、中性子拡散方程式を表現できる偏微分方程式の概論を述べる。この付録2においては、偏微分方程式から差分式に展開して[36]、数値解法を適用し、その収束方法や収束への安定性などを数式で説明する。

2-1. 偏微分方程式の数値解法の説明

偏微分方程式は解析的に表現出来ても、現実的な数値を得る事が困難な場合が多い。そこで境界条件を与える事により、数値解法を導く。偏微分方程式はその形状に関して、楕円型、放物型や双曲型に分類する事が可能である。偏微分方程式の中で楕円型は、ポテンシャル論に代表されるラプラス方程式やポアソン方程式、炉物理等で表現される中性子拡散方程式があり、物体の平衡状態を記述する時に用いられる事が多い。その多くが境界値問題として扱われている。境界条件の与え方により、その数値解が変化する事が知られており、これは閉境界である事が望ましい。ディリクレ、ノイマン等[104]の条件を閉境界線上で与えると、解は一意的に求められる。楕円型偏微分方程式の数値解[20]は滑らかで求めやすい特徴がある。この方程式は差分法で近似すると、多くの未知数からなる連立一次方程式として表現される。差分法は基本的な考えとして、未知の関数 U にある定義域を格子網で覆い、その全ての格子点上における近似値を求める事を意味する。その格子が細くなればなるほど、近似解は厳密解に近づいて行くことが知られている[22]。

この章では、偏微分方程式の特徴となる式の表現に始まり、その偏微分方程式から差分式に導く具体的な式の展開を中心として、その時の境界条件や定義領域が、その近似式の中でどのような形として導かれるかについて述べる[81]。一部は基礎編で述べた数値解法と、解法の名前は少し重複する点も存在するが、ここでは代表的な数値解法を選択して、差分形式からその数値解法への展開式[106]を導く点について述べる。後に述べる繰り返し計算の反復法において、差分形式から計算式へ展開する

上で、実用的なブロック法を用いて進めている点に特徴がある。ここでの数値解法は、その繰り返し計算の反復法の狙いとして、反復行列を求めて、行列の性質の吟味を初め、数値解法として収束する為の条件、収束の速さ、収束の割合の評価や、加速パラメータの選択方法等を考えて、第1章の基礎編で充分説明していない部分にスポットをあてている。つまり、偏微分方程式を基にして、具体的な差分形式から導かれる数値解法の収束に向けた重要な点を中心として論述している点にある。

2・2．2階線形偏微分方程式

2階線形偏微分方程式の一般式を示す。

$$(2\cdot1) \quad A \frac{\partial^2 u}{\partial x^2} + 2H \frac{\partial^2 u}{\partial x \partial y} + B \frac{\partial^2 u}{\partial y^2} = D(x, y) \frac{\partial u}{\partial x} + E(x, y) \frac{\partial u}{\partial y} + F(x, y)u + G(x, y)$$

ここで $H^2 - AB > 0$ 楕円型、 $H^2 - AB = 0$ 放物型、 $H^2 - AB < 0$ 双曲型となる。

本論では応用分野として原子力領域に焦点を充てた。この原子力領域を代表する中性子拡散方程式が、楕円型となるところから、この章では楕円型の偏微分方程式を採り上げる。

式(2・1)は $\mathbf{u}$ を未知ベクトル、 $\mathbf{A}$ は正方行列として、 $\mathbf{Au} + \mathbf{d} = 0$ の形となる。 $\mathbf{Au} = -\mathbf{d}$ として、この $-\mathbf{d}$ を右辺ベクトル $\mathbf{b}$ とおき、未知ベクトル $\mathbf{u}$ を $\mathbf{x}$ と置き換えて、式(1・1)に定義する $\mathbf{Ax} = \mathbf{b}$ の連立一次方程式に導く事が出来る。

2・3．偏微分方程式を差分式への展開

2変数の楕円型偏微分方程式について、差分近似式を導く。更に定義領域の形や境界条件が一般的な場合について述べる。

[1] 偏微分方程式を差分式へ導き

$$(2\cdot2) \quad -\frac{\partial}{\partial x} \left(P(x, y) \frac{\partial U(x, y)}{\partial x} \right) - \frac{\partial}{\partial y} \left(P(x, y) \frac{\partial U(x, y)}{\partial y} \right) + q(x, y)U(x, y) = S(x, y)$$

$$(x, y) \in \Omega, \quad \Omega : 0 \leq x \leq R_x, \quad 0 \leq y \leq R_y$$

境界 Γ 上では 0 とする。

$$(2\cdot3) \quad U(x, y) = 0, \quad (x, y) \in \Gamma$$

$P(x, y)$, $q(x, y)$, $S(x, y)$ は、ともに連続で $P(x, y) > 0$, $q(x, y) > 0$ とする。

区間 $[0, R_x]$, $[0, R_y]$ をそれぞれ $I+1$, $J+1$ 等分し、その時の分割点を

$$x_i = ih \quad (0 \leq i \leq I+1), \quad y_j = jk \quad (0 \leq j \leq J+1) \quad \text{とする。}$$

$$\text{但し、} \quad h = \frac{R_x}{(I+1)}, \quad k = \frac{R_y}{(J+1)} \quad \text{である。}$$

式(2・1)を長方形で積分する。

$$(2.4) \quad w_{ij} \left\{ \begin{array}{l} x_i - \frac{h}{2} \leq x \leq x_i + \frac{h}{2} \\ y_j - \frac{k}{2} \leq y \leq y_j + \frac{k}{2} \end{array} \right.$$

$$(2.5) \quad - \int \int_{w_{ij}} \frac{\partial}{\partial x} \left(P(x, y) \frac{\partial}{\partial x} U(x, y) \right) dx dy = - \int_{y_j - k/2}^{y_j + k/2} \left(P(x, y) \frac{\partial}{\partial x} U(x, y) \right)_{x=x_i + h/2} dy \\ + \int_{y_j - k/2}^{y_j + k/2} \left(P(x, y) \frac{\partial}{\partial x} U(x, y) \right)_{x=x_i - h/2} dy$$

$$\approx \frac{k}{h} \left[\left(-U(x_i + h, y_j) + U(x_i, y_j) \right) \times P(x_i + \frac{h}{2}, y_j) + \left(U(x_i, y_j) - U(x_i - h, y_j) \right) \times P(x_i - \frac{h}{2}, y_j) \right]$$

同様に

$$(2.6) \quad - \int \int_{w_{ij}} \frac{\partial}{\partial y} \left(P(x, y) \frac{\partial}{\partial y} U(x, y) \right) dx dy \approx \frac{h}{k} \left\{ \left(-U(x_i, y_j + k) + U(x_i, y_j) \right) \times P(x_i, y_j + \frac{k}{2}) \right. \\ \left. + \left(U(x_i, y_j) - U(x_i, y_j - k) \right) \times P(x_i, y_j - \frac{k}{2}) \right\}$$

$$(2.7) \quad \int \int_{w_{i,j}} q(x, y) U(x, y) dx dy \approx hk U(x_i, y_j) q(x_i, y_j)$$

$$(2.8) \quad \int \int_{w_{i,j}} S(x, y) dx dy \approx hk S(x_i, y_j)$$

$$u_{i,j} = U(x_i, y_j) \quad , \quad s_{i,j} = S(x_i, y_j) \quad , \quad q_{i,j} = q(x_i, y_j)$$

$$p_{i+1/2,j} = P(x_i + h/2, y_j) \quad , \quad p_{i,j+1/2} = P(x_i, y_j + k/2)$$

$$p_{i-1/2,j} = P(x_i - h/2, y_j) \quad , \quad p_{i,j-1/2} = P(x_i, y_j - k/2)$$

式 (2.1) に対する差分近似式が $I \times J$ 元連立一次方程式として得られる。

ここで境界条件 $u_{0,j} = u_{i+1,j} = 0 \quad (0 \leq j \leq I+1) \quad , \quad u_{i,0} = u_{i,j+1} = 0 \quad (0 \leq i \leq I+1)$

まとめると

$$(2.9) \quad d_{i,j} u_{i,j} - l_{i,j} u_{i-1,j} - r_{i,j} u_{i+1,j} - b_{i,j} u_{i,j-1} - t_{i,j} u_{i,j+1} = \bar{s}_{i,j} \quad , \quad (1 \leq i \leq I, 1 \leq j \leq J)$$

但し、

$$(2.10) \quad l_{i,j} = \frac{k}{h} p_{i-1/2,j} \quad , \quad r_{i,j} = \frac{k}{h} p_{i+1,j} \quad , \quad b_{i,j} = \frac{h}{k} p_{i,j-1/2} \quad , \quad t_{i,j} = \frac{h}{k} p_{i,j+1/2} \quad ,$$

$$d_{i,j} = l_{i,j} + r_{i,j} + b_{i,j} + t_{i,j} + \bar{g}_{i,j} \quad , \quad \bar{g}_{i,j} = hkg_{i,j} \quad , \quad \bar{s}_{i,j} = hks_{i,j}$$

一般的な境界条件

$$(2.11) \quad \alpha(x, y) u(x, y) + \beta(x, y) \frac{\partial u(x, y)}{\partial n} = \gamma(x, y) \quad , \quad (x, y) \in \Gamma$$

$\frac{\partial u}{\partial n}$ は法線に添った外向きの微係数を示し、 $\alpha(x, y) \geq 0$,

$$\beta(x, y) \geq 0$$

$$\alpha(x, y) + \beta(x, y) > 0 \quad \text{とする。}$$

積分記号内の偏微分を

$$(2 \cdot 12) \quad \frac{\partial u(x, y)}{\partial n} = \frac{1}{\beta(x, y)} (\gamma(x, y) - \alpha(x, y)u(x, y))$$

で置き換えれば、同じく連立一次方程式を得る。

定義領域が長方形でない場合、 x 方向、 y 方向の分割を $h_i = x_{i+1} - x_i$, $k_j = y_{j+1} - y_j$

として、境界 Γ_1 に対して分割点を含んだ多角形 Γ_p で置き換え、境界線上の分割点を 1 つの

頂点とする三角形 Δ 上での式 (2・1) の積分を次の様に行う。

ガウスの定理を用いて

$$(2 \cdot 13) \quad - \iint_{\Delta} ((Pu_x)_x + (Pu_y)_y) dx dy = - \int_{\Delta c} (Pu_x dy - Pu_y dx)$$

Δc は Δ の境界で、積分の方向は正の向きとする。 $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$ の順に積分する。

X 軸、 Y 軸に平行な $p_1 \rightarrow p_2$ の積分は上記の通りであるが、 $p_2 \rightarrow p_3$ の積分は、次の形とする。
 t をパラメータとして、

$$x = x_{i+1/2} - t \sin \theta \quad , \quad y = y_{j-1/2} + t \cos \theta \quad , \quad dx = -\sin \theta dt \quad , \quad dy = \cos \theta dt$$

$$- \int_{p_2}^{p_3} (Pu_x dy - Pu_y dx) = - \int_0^l P(u_x \cos \theta + u_y \sin \theta) dt$$

$$= - \int_0^l P \frac{\partial u}{\partial n} dt$$

$$= - \int_0^l P \left(\frac{\gamma(t) - \alpha(t)u(t)}{\beta(t)} \right) dt$$

$$= -P(x_i, y_j) \cdot \left(\frac{\gamma_{i,j} - \alpha_{i,j}u_{i,j}}{\beta_{i,j}} \right) \cdot l$$

$$l \text{ は } \overrightarrow{p_2 p_3} \text{ の長さで } ((h_i/2)^2 + (k_j/2)^2)^{1/2}$$

$P(x, y), q(x, y), S(x, y)$ は連続であると仮定して式を導いたが、 $P(x, y), q(x, y), S(x, y)$ は

区分的に連続である時、 $P(x, y)$ の不連続点で $P(x, y) \frac{\partial}{\partial x} \cdot u(x, y)$ 、 $P(x, y) \frac{\partial}{\partial y} \cdot u(x, y)$ は

連続であるとして、分割点の集合 $\{ (x_i, y_j) ; 0 \leq i \leq I+1, 0 \leq j \leq J+1 \}$ は、

$P(x, y), q(x, y), S(x, y)$ の不連続点を全て含む様にしておく。

式 (2・10) に対応して

$$(2 \cdot 14) \quad \left\{ \begin{aligned} l_{i,j} &= \frac{1}{2h_{i-1}} (P(-,+)k_j + P(-,-)k_{j-1}) \quad , \quad r_{i,j} = \frac{1}{2h_i} (P(+,+)k_j + P(+,-)k_{j-1}) \\ b_{i,j} &= \frac{1}{2k_{j-1}} (P(+,-)h_i + P(-,-)h_{i-1}) \quad , \quad t_{i,j} = \frac{1}{2k_j} (P(+,+)h_i + P(-,+)h_{i-1}) \\ d_{i,j} &= l_{i,j} + r_{i,j} + b_{i,j} + t_{i,j} + \bar{g}_{i,j} \quad (1 \leq i \leq I, 1 \leq j \leq J) \\ \bar{g}_{i,j} &= \frac{1}{4} \{ h_i k_j q(+,+) + h_{i-1} k_j q(-,+) + h_i k_{j-1} P(+,-) + h_{i-1} k_{j-1} P(-,-) \} \\ \bar{s}_{i,j} &= \frac{1}{4} \{ h_i k_j s(+,+) + h_{i-1} k_j s(-,+) + h_i k_{j-1} s(+,-) + h_{i-1} k_{j-1} s(-,-) \} \end{aligned} \right.$$

ここで

$$P(+,+) = P(x_i + h_i/2, y_i + k_j/2) \quad , \quad P(+,-) = P(x_i + h_i/2, y_i - k_{j-1}/2)$$

$$l_{0,j}, l_{1,j}, l_{i,0}, l_{i,j}, r_{0,j}, \dots \quad \text{は与えられた境界条件から定まる。}$$

u_{xx} と u_{yy} の係数が異なる時、式 (2・2) の代わりに

$$(2 \cdot 15) \quad -\frac{\partial}{\partial x} \left(P(x, y) \frac{\partial}{\partial x} u(x, y) \right) - \frac{\partial}{\partial y} \left(Q(x, y) \frac{\partial}{\partial y} u(x, y) \right) + q(x, y) u(x, y) = S(x, y)$$

$$\text{但し、} \quad (x, y) \in \Omega_1 \quad , \quad \text{ここで} \quad P(x, y) > 0 \quad , \quad Q(x, y) > 0 \quad , \quad q(x, y) > 0$$

$$\text{式 (2・10) において} \quad b_{i,j} = h/2 \cdot k \cdot Q_{i,j-1/2} \quad , \quad t_{i,j} = h/2 \cdot k \cdot Q_{i,j+1/2}$$

式 (2・14) において

$$b_{i,j} = \frac{1}{2k_{j-1}} \cdot (Q(+,-)h_i + Q(-,-)h_{i-1}) \quad , \quad t_{i,j} = \frac{1}{2k_j} \cdot (Q(+,+)h_i + Q(-,+)h_{i-1})$$

境界上で $\beta(x, y) = 0$ の時、あるいは境界がすべての Γ_p の様に X 軸、Y 軸のどちらにも平行でない部分があるとき、式 (2・11) の境界条件では解けない。この場合、次式の線積分が求められる境界条件でなければならない。

$$(2 \cdot 16) \quad \int \left(P(x, y) \frac{\partial}{\partial x} u(x, y) \cos \theta + Q(x, y) \frac{\partial}{\partial y} u(x, y) \sin \theta \right) dt$$

[2] 偏微分方程式から導かれる差分式の表現

偏微分方程式を差分系に置き換えて、連立一次方程式で解くにおいて、差分では一定の幅を持たせた近似計算となるため、数式理論とは僅かであるが誤差が考えられる。区分けした幅を極力小さく、

0（ゼロ）に近づけた方がより正確な計算になるが、この場合には計算時間が大きく増えてしまう。従って現実の問題としては、連立一次方程式の規模による計算環境のメモリの問題と、計算時間との兼ね合いになっている。

楕円型偏微分方程式を差分近似する場合、式（2・2）に対して近似式を式（1・1）として、その解は分割幅 $h^2 + k^2$ を 0 に近づけることで、式（2・2）の解となっている。熱伝導方程式の場合、時間変数 t と空間変数 x に関して $(\Delta t)^2 + (\Delta x)^2 \rightarrow 0$ であっても、もとの方程式とはわずかの誤差を持つ。

Δt , Δx の選び方に左右される。

ラプラシアン $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ を差分近似する。

これは $\Delta x = \Delta y = h$ としたとき、 $O(h^2)$ となる。

差分式 $u(x+h, y)$ のテーラー展開を行うと、

$$u(x+h, y) = u(x, y) + hu_x(x, y) + \frac{h^2}{2 \cdot 1} u_{xx}(x, y) + \frac{h^3}{3 \cdot 2 \cdot 1} u_{xxx}(x, y) + \frac{h^4}{4 \cdot 3 \cdot 2 \cdot 1} u_{xxxx}(\bar{x}, y) \\ (x < \bar{x} < x+h)$$

次に $u(x-h, y)$, $u(x, y+h)$, $u(x, y-h)$ についても展開し、それらを加え合わせる。

$$u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) = 4u(x, y) + h^2(u_{xx}(x, y) + u_{yy}(x, y)) + \frac{h^4}{6} M\theta$$

$$M = \max_{\substack{x-h < \bar{x} < x+h \\ y-h < \bar{y} < y+h}} \left(|u_{xxxx}(\bar{x}, y)|, |u_{yyyy}(x, \bar{y})| \right) \quad (-1 \leq \theta \leq 1)$$

上記の差分近似式における一般式は、次式で示される。

$$(2 \cdot 17) \quad \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u(x, y) = \frac{u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h)}{h^2} + \frac{h^2}{6} M\theta$$

2・4．偏微分方程式の差分式の係数からなる行列

この節では、偏微分方程式から導かれた差分式の係数からなる行列について述べる。

[1] 係数からなる行列の性質

I=J=3 の時、式（2・9）の係数からなる行列は次の形になる。

$$\mathbf{A} = \begin{pmatrix} \mathbf{d}_{11} & -\mathbf{r}_{11} & 0 & -\mathbf{t}_{11} & & & & \mathbf{0} \\ -\mathbf{l}_{21} & \mathbf{d}_{21} & -\mathbf{r}_{21} & \cdot & -\mathbf{t}_{21} & & & \\ 0 & -\mathbf{l}_{31} & \mathbf{d}_{31} & 0 & \cdot & -\mathbf{t}_{31} & & \\ -\mathbf{b}_{12} & \cdot & 0 & \mathbf{d}_{12} & -\mathbf{r}_{12} & \cdot & -\mathbf{t}_{12} & \\ & -\mathbf{b}_{22} & \cdot & -\mathbf{l}_{22} & \mathbf{d}_{22} & -\mathbf{r}_{22} & \cdot & -\mathbf{t}_{22} \\ & & -\mathbf{b}_{32} & \cdot & -\mathbf{l}_{32} & \mathbf{d}_{32} & 0 & \cdot & -\mathbf{t}_{32} \\ & & & -\mathbf{b}_{13} & \cdot & 0 & \mathbf{d}_{13} & -\mathbf{r}_{12} & 0 \\ & & & & -\mathbf{b}_{23} & \cdot & -\mathbf{l}_{23} & \mathbf{d}_{23} & -\mathbf{r}_{23} \\ \mathbf{0} & & & & & -\mathbf{b}_{33} & 0 & -\mathbf{l}_{33} & \mathbf{d}_{33} \end{pmatrix}$$

式 (2・14) の係数行列を $I = J = 3$ とおくと

$$\mathbf{A} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{F}_1 & & \mathbf{0} \\ \mathbf{E}_2 & \mathbf{D}_2 & \cdot & \\ & \cdot & \cdot & \mathbf{F}_{j-1} \\ \mathbf{0} & & \mathbf{E}_j & \mathbf{D}_j \end{pmatrix}, \quad \mathbf{D}_j = \begin{pmatrix} d_{1j} & -r_{1j} & \cdot & \mathbf{0} \\ -l_{2j} & d_{2j} & \cdot & \cdot \\ \cdot & \cdot & \cdot & -r_{I-1,j} \\ \mathbf{0} & \cdot & -l_{I,j} & d_{I,j} \end{pmatrix},$$

$$\mathbf{E}_j = \begin{pmatrix} -b_{1j} & & & \mathbf{0} \\ & -b_{2j} & & \\ & & \cdot & \\ \mathbf{0} & & & -b_{I,j} \end{pmatrix}, \quad \mathbf{F}_j = \begin{pmatrix} -t_{1j} & & & \mathbf{0} \\ & -t_{2j} & & \\ & & \cdot & \\ \mathbf{0} & & & -t_{I,j} \end{pmatrix}$$

この時 $\mathbf{D}_j (1 \leq j \leq J)$ は三重対角行列 (Tridiagonal Matrix) で、 $\mathbf{E}_j (2 \leq j \leq J)$,

$\mathbf{F}_j (1 \leq j \leq J-1)$ は対角行列である。

式 (2・10) 、式 (2・14) から分かる様に

$$r_{i,j} = l_{i+1,j} (1 \leq i \leq I-1, 1 \leq j \leq J), \quad t_{i,j} = b_{i,j+1} (1 \leq i \leq I, 1 \leq j \leq J-1)$$

であり、 $P(x,y) > 0$, $q(x,y) > 0$, $\alpha(x,y) \geq 0$, $\beta(x,y) \geq 0$

から $r_{i,j}, l_{i,j}, t_{i,j}, b_{i,j}$ はすべて正である。

この時、以下のことが言える。

- ① $\mathbf{A}$ は対称行列である
- ② $\mathbf{A}$ の対角要素は正であり、その他の要素は負または0である
- ③ $\mathbf{A}$ の対角要素はそれを含む行又は列の残り要素の絶対値の和よりも大である。

$$d_{i,j} > |-b_{i,j}| + |-l_{i,j}| + |-r_{i,j}| + |-t_{i,j}|$$

- ④ $\mathbf{A}$ の逆行列の各要素は正である。
- ⑤ $\mathbf{A}$ は既約行列である。

n 次順序行列 $\mathbf{P}$ を適当に選んで $\mathbf{PBP}^{-1} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{0} & \mathbf{B}_{22} \end{pmatrix}$, $\mathbf{B}_{11}, \mathbf{B}_{12}$ は正方行列の形に表わ

される様な n 次行列 $\mathbf{B}$ は可約であるという。ここで n 次順列行列 $\mathbf{P}$ をどの様にも選んでも

$$\mathbf{PBP}^{-1} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{0} & \mathbf{B}_{22} \end{pmatrix} \text{ の形にならない時、行列 } \mathbf{B} \text{ は既約であるという。}$$

- ⑥ $\mathbf{A}$ は正の定符号である。つまり $\mathbf{A}$ の固有値は全て正である。このことは、解の安定性や収束の割合に重要な意味を持つ。

[2] 繰り返し計算の方法の説明

$\mathbf{A} = (a_{ij})$ を n 次の正則行列として、式 (1.1) より

$$\mathbf{Ax} = \mathbf{b}$$

を考える。

$\mathbf{x}$, $\mathbf{b}$ を n 次のベクトルとする。行列 $\mathbf{A}$ について、次の3つの形に分離する

$$\mathbf{D}_p = \begin{pmatrix} a_{11} & & \mathbf{0} \\ & a_{22} & \\ & & \ddots \\ \mathbf{0} & & & a_{nn} \end{pmatrix}, \quad \mathbf{L}_p = - \begin{pmatrix} 0 & & \mathbf{0} \\ a_{21} & 0 & \\ \cdot & & \cdot \\ a_{n1} & \cdot & a_{n,n-1} & 0 \end{pmatrix}, \quad \mathbf{U}_p = - \begin{pmatrix} 0 & a_{12} & \cdot & a_{1n} \\ & 0 & & \cdot \\ & & \cdot & a_{n-1,n} \\ \mathbf{0} & & & 0 \end{pmatrix}$$

とおくと

$$(2.18) \quad \mathbf{A} = \mathbf{D}_p - \mathbf{L}_p - \mathbf{U}_p$$

この形で表現できる。

但し、対角行列 $\mathbf{D}_p$ の要素、 $a_{11}, a_{22}, \dots, a_{nn}$ はいずれも 0 でないとする。

(1) ヤコビ法の反復行列

式 (2.18) を次の様に変形する。

$$(2.19) \quad a_{ii}x_i = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j + k_i \quad (1 \leq i \leq n)$$

方程式 (1.1) の解に対する初期値推定ベクトルを $\mathbf{x}_i^{(0)}$ として、式 (2.19) に代入する。得られた結果を a_{ii} で割れば、新しく $\mathbf{x}_i^{(1)}$ を得る。

一般に

$$(2.20) \quad \mathbf{x}_i^{(m+1)} = \frac{1}{a_{ii}} \left(- \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j^{(m)} + k_i \right) \quad (1 \leq i \leq n, 0 \leq m)$$

により $\mathbf{x}_i^{(m+1)}$ 、 $\mathbf{x}_i^{(m)}$ とがあらかじめ定められた基準以下になるまで繰り返す。

式 (2.20) は次の形となる。

$$(2.21) \quad \mathbf{X}^{(m+1)} = \mathbf{D}_p^{-1} (\mathbf{L}_p + \mathbf{U}_p) \mathbf{X}^{(m)} + \mathbf{D}_p^{-1} \mathbf{K} \quad (m \geq 0)$$

$$(2.22) \quad \mathbf{M}_{J,p} = \mathbf{D}_p^{-1} (\mathbf{L}_p + \mathbf{U}_p)$$

この $\mathbf{M}_{J,p}$ が反復行列となる。

(2) ガウス・ザイデル法の反復行列

ヤコビ方法では $\mathbf{x}_i^{(m+1)}$ を計算する時、 $\mathbf{x}_j^{(m)}$ ($1 \leq j \leq n$) だけを用いるが、

$\mathbf{x}_1^{(m+1)}, \mathbf{x}_2^{(m+1)}, \dots, \mathbf{x}_{i-1}^{(m+1)}, \mathbf{x}_i^{(m+1)}, \dots, \mathbf{x}_n^{(m+1)}$ と計算する時、 $\mathbf{x}_i^{(m+1)}$ の計算に $(\mathbf{x}_j^{(m+1)})$ ($j < i$) を用いる事が出来る。

式で表現すると

$$(2.23) \quad \begin{aligned} a_{ii}\mathbf{x}_i^{(m+1)} &= -\sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j^{(m+1)} - \sum_{j=i+1}^n a_{ij}\mathbf{x}_j^{(m)} + k_i \\ a_{ii}\mathbf{x}_i^{(m+1)} + \sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j^{(m+1)} &= -\sum_{j=i+1}^n a_{ij}\mathbf{x}_j^{(m)} + k_i \end{aligned}$$

となる。

行列表現すると

$$(2.24) \quad \begin{aligned} (\mathbf{D}_p - \mathbf{L}_p)\mathbf{X}^{(m+1)} &= \mathbf{U}_p\mathbf{X}^{(m)} + \mathbf{K} \\ \mathbf{X}^{(m+1)} &= (\mathbf{D}_p - \mathbf{L}_p)^{-1} \mathbf{U}_p\mathbf{X}^{(m)} + (\mathbf{D}_p - \mathbf{L}_p)^{-1} \mathbf{K} \end{aligned}$$

この時の反復行列は

$$(2.25) \quad \mathbf{M}_{G,P} = (\mathbf{D}_p - \mathbf{L}_p)^{-1} \mathbf{U}_p$$

となる。

(3) SOR 法の反復行列

ガウス・ザイデル法の $\mathbf{x}_i^{(m+1)}$ と $\mathbf{x}_j^{(m)}$ との誤差を $\mathbf{R}_i^{(m)}$ とする。すなわち $\mathbf{x}_i^{(m+1)} = \mathbf{x}_i^{(m)} + \mathbf{R}_i^{(m)}$

$\mathbf{x}_1^{(0)}, \mathbf{x}_1^{(1)}, \dots, \mathbf{x}_1^{(m)}, \mathbf{x}_1^{(m+1)}, \dots$ が、式 (1.1) の解 $\mathbf{x}_i$ に近づいている時、角ステップでの誤差

$\mathbf{R}_i^{(m)}$ にパラメータ ω を加えて $\mathbf{x}_i^{(m+1)} = \mathbf{x}_i^{(m)} + \omega\mathbf{R}_i^{(m)}$ により計算する方法が SOR 法である。

ガウス・ザイデル法の補助ベクトル $\mathbf{Y}^{(m+1)}$ を

$$(2.26) \quad a_{ii}\mathbf{Y}_i^{(m+1)} = -\sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j^{(m+1)} - \sum_{j=i+1}^n a_{ij}\mathbf{x}_j^{(m)} + k_i \quad (1 \leq i \leq n, m \geq 0)$$

から計算し、この $\mathbf{Y}^{(m+1)}$ と ω とから $\mathbf{X}^{(m+1)}$ を

$$(2.27) \quad \mathbf{x}_i^{(m+1)} = \mathbf{x}_i^{(m)} + \omega(\mathbf{y}_i^{(m+1)} - \mathbf{x}_i^{(m)}) = (1-\omega)\mathbf{x}_i^{(m)} + \mathbf{y}_i^{(m+1)}$$

より求める。

式 (2.27) より

$$(2.28) \quad a_{ii}\mathbf{x}_i^{(m+1)} = a_{ii}\mathbf{x}_i^{(m)} + \omega \left(-\sum_{j=1}^{i-1} a_{ij}\mathbf{x}_j^{(m)} - \sum_{j=i+1}^n a_{ij}\mathbf{x}_j^{(m)} + k_i - a_{ii}\mathbf{x}_i^{(m)} \right) \quad (1 \leq i \leq n, m \geq 0)$$

SOR 法の行列表現

$$\begin{aligned}
(2\cdot 29) \quad & (\mathbf{D}_p - \omega \mathbf{L}_p) \mathbf{X}^{(m+1)} = (1 - \omega) \mathbf{D}_p \mathbf{X}^{(m)} + \omega (\mathbf{U}_p \mathbf{X}^{(m)} + \mathbf{K}) \\
& \mathbf{X}^{(m+1)} = (\mathbf{D}_p - \omega \mathbf{L}_p)^{-1} ((1 - \omega) \mathbf{D}_p + \omega \mathbf{U}_p) \mathbf{X}^{(m)} + (\mathbf{D}_p - \omega \mathbf{L}_p)^{-1} \mathbf{K} \quad (m \geq 0)
\end{aligned}$$

SOR 法の反復行列

$$(2\cdot 30) \quad \mathbf{M}_{s.p.w} = (\mathbf{D}_p - \omega \mathbf{L}_p)^{-1} ((1 - \omega) \mathbf{D}_p + \omega \mathbf{U}_p)$$

で表現できる。

(4) チェビシェフ法の漸化式

$\mathbf{M}$ を反復行列として、

$$(2\cdot 31) \quad \mathbf{X}^{(m+1)} = \mathbf{M} \mathbf{X}^{(m)} + \mathbf{h}$$

で与えられる反復法を考える。

m 回の繰り返し計算で得られた $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)}$ を用いて

$$(2\cdot 32) \quad \mathbf{Y}^{(m+1)} = \sum_{j=0}^m P_j(m) \mathbf{X}^{(j)} \quad (m \geq 0)$$

から $\mathbf{Y}^{(m)}$ を計算し、この $\mathbf{Y}^{(m)}$ が式 (2・31) の解 $\mathbf{X}$ に速く収束する様に、m 次の多項式

$P_j(m)$ を選ぶ考えから導かれた。

各ステップ毎に、それまでの $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)}$ を保存して、式 (2・32) から $\mathbf{Y}^{(m)}$ を計算する。

多項式 P がチェビシェフ多項式と関係づけられ、チェビシェフ多項式が満たす漸化式を用いることで、 $\mathbf{X}^{(m)}$ と $\mathbf{X}^{(m+1)}$ だけを保存しておけば良い。

チェビシェフ法は次式で与えられる。

$$\begin{aligned}
(2\cdot 33) \quad & \mathbf{X}^{(m+1)} = \omega_{m+1} (\mathbf{M} \mathbf{X}^{(m)} + \mathbf{h}) (1 - \omega_{m+1}) \mathbf{X}^{(m-1)} \\
& \omega_{m+1} = 1 + \frac{C_{m-1}(1/\rho)}{C_{m+1}(1/\rho)}
\end{aligned}$$

ここで C_m は、m 次のチェビシェフ多項式であり、 ρ は定数で $0 < \rho < 1$ に対して収束し、

反復行列 $\mathbf{M}$ のスペクトル半径を ρ として選んだとき、最も速く収束する。

初回の計算は $\mathbf{X}^{(1)} = \mathbf{M} \mathbf{X}^{(0)} + \mathbf{h}$ とする。ここでチェビシェフ多項式は、次の漸化式を満たす。

$$\begin{aligned}
(2\cdot 34) \quad & C_0(x) = 1, C_1(x) = x \\
& C_{m+1}(x) = 2x C_m(x) - C_{m-1}(x) \quad (m \geq 1)
\end{aligned}$$

2・5．偏微分方程式の数値解法、収束する為の条件

この節では、偏微分方程式から導かれた各種の数値解法が、収束する為の条件を始め、収束の速さや、収束の割合について述べる。

[1] 基礎的反復法の収束の割合

方程式 (1・1) を解く繰り返し法を、次式で定義する。

$$(2・35) \quad \mathbf{x}^{(m)} = \mathbf{M}\mathbf{x}^{(m-1)} + \mathbf{g} \quad (m > 0)$$

$\mathbf{M}$ は $I \times J$ 次の反復行列である

初期ベクトル $\mathbf{x}^{(0)}$ と解ベクトル $\mathbf{x}$ との残差を表すベクトルを

$$(2・36) \quad \boldsymbol{\varepsilon}^{(0)} = \mathbf{x}^{(0)} - \mathbf{x}$$

として、 m 回の繰り返し計算で得られた $\mathbf{x}^{(m)}$ に対する残差ベクトルを $\boldsymbol{\varepsilon}^{(m)}$ とすると

$$(2・37) \quad \boldsymbol{\varepsilon}^{(m)} = \mathbf{M}\boldsymbol{\varepsilon}^{(m-1)} = \mathbf{M}^2\boldsymbol{\varepsilon}^{(m-2)} = \dots = \mathbf{M}^m\boldsymbol{\varepsilon}^{(0)}$$

$\mathbf{M}$ の固有値を λ_k 、その固有ベクトルを $\mathbf{x}_k$ とおくと

$$(2・38) \quad \begin{aligned} \mathbf{M}\mathbf{x}_k &= \lambda_k \mathbf{x}_k \\ \mathbf{M}^m\mathbf{x}_k &= (\lambda_k)^m \mathbf{x}_k \end{aligned}$$

初期ベクトル $\mathbf{x}^{(0)}$ を固有ベクトル $\mathbf{x}_k$ を用いて展開する。

C_k を展開係数とすると次式で表現できる。

$$(2・39) \quad \mathbf{x}^{(0)} = \sum_{k=1}^{I \times J} C_k \mathbf{x}_k$$

式 (2・37) 式 (2・38) 式 (2・39) より

$$\boldsymbol{\varepsilon}^{(m)} = \sum_{k=1}^{I \times J} (\lambda_k)^m C_k \mathbf{x}_k$$

$m \rightarrow \infty$ の時 $\lambda_k^m \rightarrow 0$ であれば、 $\boldsymbol{\varepsilon}^{(m)} \rightarrow 0$ となり、 $\mathbf{x}^{(m)}$ は解ベクトル $\mathbf{x}$ に収束

するから、初期ベクトル $\mathbf{x}^{(0)}$ に対して式 (2・34) が収束する為には

$$\lambda_{\max} = \max_k |\lambda_k| < 1 \quad \text{となる。}$$

$\lambda_{\max}$ を行列 $\mathbf{M}$ のスペクトル半径といい、 $\rho(\mathbf{M})$ で表現する。

反復行列 $\mathbf{M}$ に対して $\rho(\mathbf{M})$ が小さいほど速く収束する。

式 (2・37) から

$$(2\cdot40) \quad \|\boldsymbol{\varepsilon}^{(m)}\| = \|\mathbf{M}^{(m)}\boldsymbol{\varepsilon}^{(0)}\| \leq \|\mathbf{M}^{(m)}\| \|\boldsymbol{\varepsilon}^{(0)}\|, \quad \frac{\|\boldsymbol{\varepsilon}^{(m)}\|}{\|\boldsymbol{\varepsilon}^{(0)}\|} \leq \|\mathbf{M}^{(m)}\|$$

となる。

$\|\boldsymbol{\varepsilon}^{(m)}\|$ はベクトルに対するユークリッドノルムを表し、 $\|\mathbf{M}\|$ は行列 $\mathbf{M}$ に対するスペクトル

ノルムであり、 $\mathbf{M}$ が実対称行列であるとき、 $\|\mathbf{M}\| = \rho(\mathbf{M})$ である。

m 回の繰り返し計算によって減少した誤差の1回の繰り返し計算あたりの平均値は、

$$\left(\frac{\|\boldsymbol{\varepsilon}^{(m)}\|}{\|\boldsymbol{\varepsilon}^{(0)}\|} \right)^{1/m} \text{ で与えられる。 (2\cdot37) 式から } \left(\frac{\|\boldsymbol{\varepsilon}^{(m)}\|}{\|\boldsymbol{\varepsilon}^{(0)}\|} \right)^{1/m} \leq \|\mathbf{M}^{(m)}\|^{1/m} \text{ であり、}$$

$R_m(\mathbf{M})$ を次に定義して反復法の式 (2\cdot35) の収束の割合に対する目安とする。

$$(2\cdot41) \quad R_m(\mathbf{M}) = \frac{-\ln \|\mathbf{M}^{(m)}\|}{m}$$

$m \rightarrow \infty$ の時は $R_\infty(\mathbf{M})$ で表現する。

$$(2\cdot42) \quad R_\infty(\mathbf{M}) = \lim_{m \rightarrow \infty} R_m(\mathbf{M}^{(m)}) = \lim_{m \rightarrow \infty} -\ln \frac{\|\mathbf{M}^{(m)}\|}{m}$$

$\mathbf{M}$ が実対称行列であれば、 $\|\mathbf{M}\| = \rho(\mathbf{M})$ であるから、

$$(2\cdot43) \quad R_\infty(\mathbf{M}) = R_m(\mathbf{M}) = -\ln \rho(\mathbf{M})$$

m を繰り返しの回数として、 l を整数とすると、 $\rho^m(\mathbf{M}) \leq e^{-l}$ を満たす m は

$$m \geq \frac{-l}{\ln \rho(\mathbf{M})} \text{ となる。} \quad \text{つまり } \rho(\mathbf{M}) < 1 \text{ となる。}$$

収束判定の基準を定める l に対して、繰り返し回数 m は $\frac{1}{R_\infty}$ に比例する。

$R_m(\mathbf{M})$ を平均的な収束の割合(average rate of convergence)とし、 $R_\infty(\mathbf{M})$ を漸近的な収束の割合(asymptatic rate of convergence)とする。式 (2\cdot34) の繰り返し計算で $\rho(\mathbf{M})$ が小さいほど速く収束する。従って $R_\infty(\mathbf{M})$ が大きいほど速く収束する。

(1) ヤコビ法、ガウス・ザイディル法、SOR法の収束性

式 (2\cdot35) の行列 $\mathbf{M}$ が、前節「係数からなる行列の性質」で述べた性質を持っている時、ヤコビ法、ガウスザイディル法 SOR 法の収束性、及び収束の速さについて、次の様な関係が証明されている。

① $0 < \rho(\mathbf{M}_{J,G}) < 1$ すなわちヤコビ法は必ず収束する。

② $0 < \rho(\mathbf{M}_{G,P}) < \rho(\mathbf{M}_{J,P}) < 1$ ガウス・ザイディル法の方が、ヤコビ法より速く収束する。

③ $0 < \omega < 2$ に対して $|\omega - 1| \leq \rho(\mathbf{M}_{S,P,\omega}) < 1$

④ $\mathbf{M}_{J,G}$ の固有値を μ_i 、 $\mathbf{M}_{G,P}$ の固有値を λ_j とするとき $1 \leq i, j < n$ が成り立つ。

⑤ SOR 法の加速パラメータ ω の最適値 ω_b は次式で与えられる。

$$\omega_b = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{M}_{J,P})}}$$

⑥ ⑤で与えられた ω_b に対して、 $\mathbf{M}_{S,P,\omega_b}$ のスペクトル半径は次式で与えられる。

$$\rho(\mathbf{M}_{S,P,\omega_b}) = \omega_b - 1$$

ヤコビ法の漸近的収束の割合 $-\ln \rho(\mathbf{M}_{J,P})$

⑦ $R_\infty(\mathbf{M}_{G,P}) = 2R_\infty(\mathbf{M}_{J,P})$ ④で $\omega = 1$ とおけば $\lambda_j = \mu_j^2$

⑧ $R_\infty(\mathbf{M}_{G,P,\omega_b}) = -\ln(\omega_b - 1)$ これは⑥と式 (2.42) から明らかである。

⑨ $R_\infty(\mathbf{M}_{S,P,\omega_b}) = 2\sqrt{2} \left(R_\infty(\mathbf{M}_{J,P}) \right)^{\frac{1}{2}}$

以上の事から、ヤコビ法、ガウスザイデル法、SOR 法の漸近的な収束の役割の大小関係は明確となる。実際の値を知る為には、 $\rho(\mathbf{M}_{J,P})$ を求める必要がある。

$\rho(\mathbf{M}_{J,P})$ の推定は次の近似式となる。 $\mathbf{M}_{J,P} = (m_{ij})$ とする時

$$(2.44) \quad \min_{1 \leq i \leq n} \left(\sum_{j=1}^n |m_{ij}| \right) \leq \rho(\mathbf{M}_{J,P}) \leq \max_{1 \leq i \leq n} \left(\sum_{j=1}^n |m_{ij}| \right)$$

の関係式が成立する。これはすべての規約行列に対して成り立つ。

(2) チェビシェフ法の収束性

チェビシェフ法の収束の割合を定義する。式 (2.31) から $\mathbf{Y}^{(m)} = \sum_{j=0}^m P_j(\mathbf{m}) \mathbf{X}^{(j)}$ における多項式 $P_j(\mathbf{m})$ は $\mathbf{Y}^{(m)}$ と $\mathbf{X}^{(j)}$ に式 (2.30) の解 $\mathbf{X}$ を入れて、 $\sum_{j=0}^m P_j(\mathbf{m}) = 1$ でなければならない。式 (2.36) 式 (2.37) と同様に $\boldsymbol{\varepsilon}^{(m)} = \mathbf{X}^{(m)} - \mathbf{X}$ 、 $\boldsymbol{\varepsilon}^{(m)} = \mathbf{Y}^{(m)} - \mathbf{X}$ ($m \geq 0$) とおくと $\boldsymbol{\varepsilon}^{(m)} = \mathbf{M}^{(m)} \boldsymbol{\varepsilon}^{(m)}$ が成り立つ。

$$\begin{aligned}\varepsilon^{(m)} &= \sum_{j=0}^m P_j(m) \mathbf{X}^{(j)} - \mathbf{X} = \sum_{j=0}^n P_j(m) \mathbf{X}^{(j)} - \sum_{j=0}^n P_j(m) \mathbf{X} \\ &= \sum_{j=0}^n P_j(m) (\mathbf{X}^{(j)} - \mathbf{X}) = \sum_{j=0}^n P_j(m) \varepsilon^{(j)}\end{aligned}$$

従って $\varepsilon^{(m)} = \left(\sum_{j=0}^m P_j(m) \right) \mathbf{M}^j$ となる。

そこでチェビシェフの反復行列を

$$(2.45) \quad \mathbf{M}_{C,P}(m) = \sum_{j=0}^m P_j(m) \mathbf{M}^j$$

とにおいて、式 (2.41) 式 (2.43) と同じように

$$(2.46) \quad R_m(\mathbf{M}_{C,P}(m)) = -\ln \frac{\|\mathbf{M}_{C,P}(m)\|}{m}$$

$$(2.47) \quad R_\infty(\mathbf{M}_{C,P}) = \lim_{m \rightarrow \infty} R_m(\mathbf{M}_{C,P}(m))$$

でチェビシェフ法の平均的な収束の割合及び漸近的な収束の割合を定義する。

$\rho = \rho(\mathbf{M})$ の時

ヤコビ等の収束性 (①～⑨) の続きの項目として、

$$(10) \quad R_\infty(\mathbf{M}_{C,P}) = -\frac{1}{2} \ln(\omega_b - 1)$$

この式と⑧から、チェビシェフ法の式 (2.33) と加速パラメータとして最適値 ω_b を用いた SOR 法では、SOR 法の収束が速い場合がある。

ρ を $\mathbf{M}$ のスペクトル半径とした時

$$(11) \quad \lim_{m \rightarrow \infty} \omega_m = \omega_b$$

となる。

[2] 計算繰り返し方法の代表例

式 (2.9) にみる $u_{i,j}$ に隣接した4点 $u_{i-1,j}, u_{i+1,j}, u_{i,j-1}, u_{i,j+1}$ の関係から1点を求めるものをポイント法と称し、一直線に並ぶ点を纏めて解く方法をブロック法と称する。このブロック法で数値解を求める。

(1) ブロック法

式 (1.1) より $\mathbf{Ax} = \mathbf{b}$

行列 $\mathbf{A}$ を次に分解する。

$$(2.48) \quad \mathbf{A} = \mathbf{D} + \mathbf{E} + \mathbf{F}$$

ここで

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & & \mathbf{0} \\ & \mathbf{D}_2 & \\ & & \ddots \\ \mathbf{0} & & & \mathbf{D}_J \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} \mathbf{0} & & \mathbf{0} \\ \mathbf{E}_2 & \cdot & \\ & \cdot & \cdot \\ \mathbf{0} & & \mathbf{E}_J & \mathbf{0} \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \mathbf{0} & \mathbf{F}_1 & & \mathbf{0} \\ & \cdot & \cdot & \\ & & \cdot & \mathbf{F}_{J-1} \\ \mathbf{0} & & & \mathbf{0} \end{pmatrix}$$

小行列

$$\mathbf{D}_j (1 \leq j \leq J), \quad \mathbf{E}_j (2 \leq j \leq J), \quad \mathbf{F}_j (1 \leq j \leq J-1)$$

は 3・4. [1] 係数からなる行列の性質、で定義されたものである。これは I 次の行列である。

$$\text{次に} \quad \mathbf{x} = \text{col.}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J), \quad \mathbf{b} = \text{col.}(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_J)$$

$\mathbf{x}_j$, $\mathbf{b}_j$ は I 次のベクトルを表す。

$$\mathbf{E}_i \mathbf{x}_{i-1} + \mathbf{D}_i \mathbf{x}_i + \mathbf{F}_i \mathbf{x}_{i+1} = \mathbf{b}_i \quad (1 \leq i \leq J, \mathbf{E}_1 = \mathbf{F}_J = \mathbf{0})$$

繰り返し添え字への展開

$$(2 \cdot 49) \quad \mathbf{D}_i \mathbf{x}_i^{(m+1)} = -(\mathbf{E}_i \mathbf{x}_{i-1}^{(m)} + \mathbf{F}_i \mathbf{x}_{i+1}^{(m)}) + \mathbf{b}_i$$

$$(2 \cdot 50) \quad \mathbf{x}_i^{(m+1)} = -\mathbf{D}_i^{-1}(\mathbf{E}_i \mathbf{x}_{i-1}^{(m)} + \mathbf{F}_i \mathbf{x}_{i+1}^{(m)}) + \mathbf{D}_i^{-1} \mathbf{b}_i, \quad \mathbf{x}_i^{(m+1)} \quad (1 \leq i \leq J)$$

を求めて行く方法がブロックヤコビ法である。

$$(2 \cdot 51) \quad \mathbf{D} \mathbf{x}^{(m+1)} = -(\mathbf{E} + \mathbf{F}) \mathbf{x}^{(m)} + \mathbf{b}$$

$$(2 \cdot 52) \quad \mathbf{x}^{(m+1)} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) \mathbf{x}^{(m)} + \mathbf{D}^{-1} \mathbf{b}$$

この時、反復行列を $\mathbf{M}$ とすると

$$(2 \cdot 53) \quad \mathbf{M}_{J,B} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})$$

である。

(2) ブロック SOR 法

ブロック型の SOR 法は SLOR 法と呼ばれる。

$$(2 \cdot 54) \quad \mathbf{D}_i \mathbf{y}_i^{(m+1)} = -(\mathbf{E}_i \mathbf{x}_{i-1}^{(m)} + \mathbf{F}_i \mathbf{x}_{i+1}^{(m)}) + \mathbf{b}_i$$

$$(2 \cdot 55) \quad \mathbf{x}_i^{(m+1)} = (1 - \omega) \mathbf{x}_i^{(m)} + \omega \mathbf{y}_i^{(m+1)}$$

から

$$(2 \cdot 56) \quad \mathbf{x}_i^{(m+1)} = (1 - \omega) \mathbf{x}_i^{(m)} - \omega \mathbf{D}_i^{-1}(\mathbf{E}_i \mathbf{x}_{i-1}^{(m+1)} + \mathbf{F}_i \mathbf{x}_{i+1}^{(m)}) + \omega \mathbf{D}_i^{-1} \mathbf{b}_i$$

すなわち

$$(2 \cdot 57) \quad \mathbf{x}^{(m+1)} = (\mathbf{D} - \omega \mathbf{E})^{-1}((1 - \omega) \mathbf{D} + \omega \mathbf{F}) \mathbf{x}^{(m)} - \omega (\mathbf{D} - \omega \mathbf{E})^{-1} \mathbf{b}$$

この時の反復行列

$$(2.58) \quad \mathbf{M}_{J,B,\omega} = (\mathbf{D} - \omega\mathbf{E})^{-1}((1-\omega)\mathbf{D} + \omega\mathbf{F})$$

で表す事が出来る。 $\omega = 1$ の時、ブロック・ガウスザイデル法である。

[3] ポイント法とブロック法の比較

偏微分方程式を差分式に展開して、数値解を求めるにおいて、2つの方法がある。1つは個々の式を逐次に計算ステップを進める方法であり、これをポイント法と呼んでいる。これに対して、差分式を格子状に配列したとして、同一直線上に並んだ式を同時に修正を加えて、次のステップへ導く方法をブロック法と呼んでいる。この2つの内どちらが速く収束するかについて、以下の形で展開する。

式 (1.1) の行列 $\mathbf{A}$ を次式の形に分解する。

$$(2.59) \quad \mathbf{A} = \mathbf{B} - \mathbf{C}$$

$\mathbf{B}, \mathbf{C}$ は $\mathbf{A}$ と同じ次数であり、正則行列とする。

ブロックヤコビ法では $\mathbf{B} = \mathbf{D}$, $\mathbf{C} = \mathbf{E} + \mathbf{F}$

ブロック型の SOR 法は SLOR 法と呼ばれる。

SLOR 法では $\mathbf{B} = \frac{1}{\omega}(\mathbf{D} - \omega\mathbf{E})$, $\mathbf{C} = \frac{1}{\omega}((1-\omega)\mathbf{D} + \omega\mathbf{F})$ であるから

$$(2.60) \quad \mathbf{X}^{(m+1)} = \mathbf{B}^{-1}\mathbf{C}\mathbf{X}^{(m)} + \mathbf{B}^{-1}\mathbf{b}$$

と表現できる。

$$\mathbf{B}^{-1} \geq 0 \quad \text{であれば、} \quad \rho(\mathbf{B}^{-1}\mathbf{C}) = \frac{\rho(\mathbf{A}^{-1}\mathbf{C})}{1 + \rho(\mathbf{A}^{-1}\mathbf{C})} < 1$$

が成立し、繰り返し法として式 (2.60) は収束する。

$$\mathbf{A}^{-1} > 0 \quad \text{であるとし、} \quad \mathbf{A} = \mathbf{B}_1 - \mathbf{C}_1 = \mathbf{B}_2 - \mathbf{C}_2 \quad , \quad \mathbf{C}_2 \geq \mathbf{C}_1 \geq 0 \quad , \quad (\mathbf{C}_2 \neq \mathbf{C}_1)$$

$$(2.61) \quad \begin{aligned} 0 < \rho(\mathbf{B}_1^{-1}\mathbf{C}_1) < \rho(\mathbf{B}_2^{-1}\mathbf{C}_2) < 1 \\ 0 < R_\infty(\mathbf{B}_2^{-1}\mathbf{C}_2) < R_\infty(\mathbf{B}_1^{-1}\mathbf{C}_1) \end{aligned}$$

ここで $\mathbf{B}$ が正則であること、そして $\mathbf{A}$ の逆行列の各要素が正である。

つまりブロック法の方がポイント法より速く収束する。

(1) ブロック法としての S2LOR 法の収束性

ブロック法は同一直線上の点をひとまとめにして同時に修正して行く方法であるが、更に二直線あるいはそれ以上の直線を一括して、それらの直線上の点をまとめて修正して行く方法もあり、二直線上の点を一括して扱う SOR 法を S2LOR (Successive 2-Line Overrelaxation Iterative Method) と呼んでいる。スプリッティングの考えからは、 $\mathbf{B}$ が必要な条件を満たす限り、SOR 法より S2LOR 法の方が収束するまでに要する繰り返し計算の回数は少なく済む。

今係数行列を次の形にする。

$$\mathbf{A} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{F}_1 \\ \mathbf{E}_2 & \mathbf{D}_2 \end{pmatrix}$$

$$\mathbf{D}_j = \begin{pmatrix} d_{1,j} & -b_{1,j} & -r_{1,j} & & & \mathbf{0} \\ -t_{2,j} & d_{2,j} & 0 & -r_{2,j} & & \\ -l_{3,j} & 0 & d_{3,j} & -b_{3,j} & -r_{3,j} & \\ & -l_{4,j} & -t_{4,j} & d_{4,j} & 0 & -r_{4,j} \\ & & -l_{5,j} & 0 & d_{5,j} & -b_{5,j} \\ \mathbf{0} & & & -l_{6,j} & -t_{6,j} & d_{6,j} \end{pmatrix} \quad (j=1,2)$$

$$\mathbf{F}_1 = \begin{pmatrix} 0 & & & & & \mathbf{0} \\ -b_{2,1} & 0 & & & & \\ & 0 & 0 & & & \\ & & -b_{4,1} & 0 & & \\ & & & 0 & 0 & \\ \mathbf{0} & & & & -b_{6,1} & 0 \end{pmatrix}$$

$$\mathbf{E}_2 = \begin{pmatrix} 0 & -t_{1,2} & & & & \mathbf{0} \\ & 0 & 0 & & & \\ & & 0 & -t_{3,2} & & \\ & & & 0 & 0 & \\ & & & & 0 & -t_{5,2} \\ \mathbf{0} & & & & & 0 \end{pmatrix}$$

この時 S2LOR 法は次の様に計算する。

$$\begin{aligned} \mathbf{X}_1^{(m+1)} &= (1-\omega)\mathbf{X}_1^{(m)} - \omega\mathbf{D}_1^{-1}\mathbf{F}_1\mathbf{X}_2^{(m)} + \omega\mathbf{D}_1^{-1}\mathbf{K}_1 \\ \mathbf{X}_2^{(m+1)} &= (1-\omega)\mathbf{X}_2^{(m)} - \omega\mathbf{D}_2^{-1}\mathbf{E}_2\mathbf{X}_1^{(m)} + \omega\mathbf{D}_2^{-1}\mathbf{K}_2 \end{aligned}$$

この計算式に現れる $\mathbf{D}_1^{-1}, \mathbf{D}_2^{-1}$ の計算が SOR 法より煩雑になっている。

これに対して、ブロック法では、逆行列を必要とする行列は全て三重対角行列であり、以下の手順で容易に解く事が出来る。

$$(2 \cdot 62) \quad -l_k y_{k-1} + d_k y_k - r_k y_{k+1} = g_k \quad (1 \leq k \leq I, l_1 = r_1 = 0)$$

式 (2・62) を解くのに $y_k = \mathbf{P}_k + \mathbf{Q}_k y_{k+1}$ とおき、 y_{k-1} を消去して係数を比較すると、次の公式が得られる。

$$\begin{aligned}
(2.63) \quad & \mathbf{P}_k = \frac{y_k + \mathbf{l}_k \mathbf{P}_{k-1}}{d_k - \mathbf{l}_k \mathbf{Q}_{k-1}}, \mathbf{P}_1 = \frac{y_1}{d_1} \\
& \mathbf{Q}_k = \frac{r_k}{d_k - \mathbf{l}_k \mathbf{Q}_{k-1}}, \mathbf{Q}_1 = \frac{1}{d_1} \\
& y_k = \mathbf{P}_k + \mathbf{Q}_k y_{k+1}
\end{aligned}$$

式 (2.63) において、まず $\mathbf{P}_k, \mathbf{Q}_k$ を $k=1 \sim I$ まで求め、次に $y_k = \mathbf{P}_k + \mathbf{Q}_k y_{k+1}$ により、 y_k を $k=1 \sim I$ まで求める。ポイント法に比べてブロック法の長所は、収束が速いほかに、逆行列が簡単にできることにある。

式 (2.63) をブロック法に適用すると、

$$(2.64) \quad -\mathbf{E}_k \mathbf{X}_{k-1} + \mathbf{D}_k \mathbf{X}_k - \mathbf{F}_k \mathbf{X}_{k+1} = \mathbf{K}_k \quad (1 \leq k \leq J)$$

を次の式から直接解く。

$$\begin{aligned}
(2.65) \quad & \mathbf{P}_k = (\mathbf{D}_k - \mathbf{E}_k \mathbf{Q}_{k-1})^{-1} (\mathbf{K}_k + \mathbf{E}_k \mathbf{P}_{k-1}) \\
& \mathbf{P}_1 = \mathbf{D}_1^{-1} \mathbf{K}_1 \\
& \mathbf{Q}_k = (\mathbf{D}_k - \mathbf{E}_k \mathbf{Q}_{k-1})^{-1} \mathbf{F}_k \\
& \mathbf{Q}_1 = \mathbf{D}_1^{-1} \mathbf{E}_1 \\
& \mathbf{X}_k = \mathbf{P}_k + \mathbf{Q}_k \mathbf{X}_{k+1} \quad (1 \leq k \leq J)
\end{aligned}$$

この方法をファクトリゼーション法 (factorization method) と呼んでいる。

(2) ブロック法としての ADI 法の収束性

ADI 法は、1955 年にピースマンとラチフォードにより提唱された。ここではブロック法の考え方で進める。

式 (2.2) で

$$P(x, y) = c_1, \quad q(x, y) = c_2$$

(c_1, c_2) は定数、分割幅は任意とする。境界条件は式 (2.11) で示される。

差分式から得られる連立一次方程式において、式 (1.1) から

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

n ($= I \times J$) 次行列 $\mathbf{A}$ を次式の形に分解する。

$$(2.66) \quad \mathbf{A} = \mathbf{H} + \mathbf{V}$$

$$(2.67) \quad d_{i,j}^{(x)} = l_{i,j} + r_{i,j} + \frac{1}{2} q_{i,j}, \quad d_{i,j}^{(y)} = b_{i,j} + t_{i,j} + \frac{1}{2} q_{i,j}$$

とおくことで

$$(2\cdot68) \quad \mathbf{H} = \begin{pmatrix} d_{1,1}^{(x)} & -r_{1,1} & & & & & & \mathbf{0} \\ -l_{2,1} & d_{2,1}^{(x)} & \cdot & & & & & \\ & \cdot & \cdot & -r_{I-1,1} & & & & \\ & & -l_{I,1} & d_{I,1}^{(x)} & \cdot & & & \\ & & & \cdot & \cdot & -r_{1,J} & & \\ & & & & -l_{2,J} & d_{2,J}^{(x)} & \cdot & \\ & & & & & \cdot & \cdot & -r_{I-1,J} \\ \mathbf{0} & & & & & & -l_{I,J} & d_{I,J}^{(x)} \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} d_{1,1}^{(y)} & 0 & -t_{1,1} & & & & & \mathbf{0} \\ 0 & d_{2,1}^{(y)} & 0 & -t_{2,1} & & & & \\ -b_{1,2} & 0 & \cdot & \cdot & & & & \\ & -b_{2,2} & \cdot & d_{I,1}^{(y)} & & & & \\ & & \cdot & & d_{1,2}^{(y)} & & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & \cdot & d_{1,J}^{(y)} & \cdot \\ & & & & & -b_{2,J} & \cdot & d_{2,J}^{(y)} & \cdot & -t_{I,J-1} \\ & & & & & & \cdot & 0 & \cdot & 0 \\ \mathbf{0} & & & & & & & -b_{I,J} & 0 & d_{J,J}^{(y)} \end{pmatrix}$$

$\mathbf{H}$, $\mathbf{V}$ はそれぞれ対称行列である。

式 (1・1) を解くためには、補助ベクトル $\mathbf{Y}^{(m+1)}$ を第一式から求めて、第二式に近似の解

$\mathbf{X}^{(m+1)}$ を求める。

$$(2\cdot69) \quad \begin{aligned} (\mathbf{H} + \omega_{m+1}\mathbf{I})\mathbf{Y}^{(m+1)} &= (\omega_{m+1}\mathbf{I} - \mathbf{V})\mathbf{X}^{(m)} + \mathbf{b} \quad (m \geq 0) \\ (\mathbf{V} + \omega_{m+1}\mathbf{I})\mathbf{X}^{(m+1)} &= (\omega_{m+1}\mathbf{I} - \mathbf{H})\mathbf{Y}^{(m+1)} + \mathbf{b} \quad (m \geq 0) \end{aligned}$$

次式で表現できる。

$$(2\cdot70) \quad \mathbf{X}^{(m+1)} = \mathbf{M}_{\omega_{m+1}}\mathbf{X}^{(m)} + \mathbf{h}_{\omega_{m+1}}(\mathbf{b})$$

$$(2\cdot71) \quad \mathbf{M}_{\omega_{m+1}} = (\mathbf{V} + \omega_{m+1}\mathbf{I})(\mathbf{I} - \mathbf{H})(\mathbf{H} + \omega_{m+1}\mathbf{I})^{-1}(\omega_{m+1}\mathbf{I} - \mathbf{V})$$

$$(2\cdot72) \quad \mathbf{h}_{\omega_{m+1}}(\mathbf{b}) = (\mathbf{V} + \omega_{m+1}\mathbf{I})^{-1} \left\{ (\omega_{m+1}\mathbf{I} - \mathbf{H})(\mathbf{H} + \omega_{m+1}\mathbf{I})^{-1} + \mathbf{I} \right\} \mathbf{b}$$

式 (2・69) では水平方向に $(j = 1 \sim J)$ 同一直線上の点を纏めて解き、そこで得た新しい

$\mathbf{Y}^{(m+1)}$ を用いて垂直方向 $(i = 1 \sim I)$ に同一直線上の点を纏めて解くことになる。

ADI 法の収束性

式 (1・1) の解を $\mathbf{x}$, 誤差を $\boldsymbol{\varepsilon}^{(m)} = \mathbf{x}^{(m)} - \mathbf{x}$ とする。式 (2・70) から

$$\begin{aligned}
 \boldsymbol{\varepsilon}^{(m)} &= \mathbf{M}_{\omega m} \boldsymbol{\varepsilon}^{(m-1)} = \mathbf{M}_{\omega m} \mathbf{M}_{\omega m-1} \boldsymbol{\varepsilon}^{(m-2)} = \cdots = \mathbf{M}_{\omega m} \mathbf{M}_{\omega m-1} \cdots \mathbf{M}_{\omega 2} \mathbf{M}_{\omega 1} \\
 (2 \cdot 73) \quad &= \left(\prod_{j=1}^m \mathbf{M}_{\omega j} \right) \boldsymbol{\varepsilon}^{(0)} \\
 &\frac{\|\boldsymbol{\varepsilon}^{(m)}\|}{\|\boldsymbol{\varepsilon}^{(0)}\|} \leq \left\| \prod_{j=1}^m \mathbf{M}_{\omega j} \right\|
 \end{aligned}$$

$\mathbf{x}^{(m)}$ が $m \rightarrow \infty$ の時、解 $\mathbf{x}$ に収束する為には

$$(2 \cdot 74) \quad \left\| \prod_{j=1}^m \mathbf{M}_{\omega j} \right\| < 1 \quad (m > 0)$$

式 (3・74) より ω を選び $\mathbf{H}$, $\mathbf{V}$ の固有値を λ_j, μ_j ($1 \leq j \leq n$) とする。

$$\begin{aligned}
 (2 \cdot 75) \quad \|\mathbf{M}_{\omega}\| &= \|(\mathbf{V} + \omega \mathbf{I})^{-1}(\omega \mathbf{I} - \mathbf{H})(\mathbf{H} + \omega \mathbf{I})^{-1}(\omega \mathbf{I} - \mathbf{V})\| \\
 &\leq \|(\omega \mathbf{I} - \mathbf{H})(\mathbf{H} + \omega \mathbf{I})^{-1}\| \cdot \|(\omega \mathbf{I} - \mathbf{V})(\mathbf{V} + \omega \mathbf{I})^{-1}\| \\
 &= \left(\max_{1 \leq j \leq n} \left| \frac{\omega - \lambda_j}{\omega + \lambda_j} \right| \right) \left(\max_{1 \leq j \leq n} \left| \frac{\omega - \mu_j}{\omega + \mu_j} \right| \right)
 \end{aligned}$$

$\omega > 1$ $\|\mathbf{M}_{\omega}\| < 1$ で、反復行列 $\mathbf{M}_{\omega}$ を中心とする計算は収束する。

式 (2・70) による反復計算は $\omega_m > 0$ として用いると必ず収束する。

$\mathbf{H}$, $\mathbf{V}$ は対称行列であるが、 $\mathbf{H}$, $\mathbf{V}$ が交換可能であると、 $\mathbf{H}\mathbf{V} = \mathbf{V}\mathbf{H}$ で、正の実数の集合 $\{\omega_j\}$ ($1 \leq j \leq m$) に対して、式 (2・74) が満たされる。

$\mathbf{H}\mathbf{V} = \mathbf{V}\mathbf{H}$ の時、 $\mathbf{H}$, $\mathbf{V}$ は正規直交系をなす固有ベクトル $\{e_i\}$ ($1 \leq i \leq n$) を共有する。

[4] 加速パラメータ、収束の割合

式 (2・69) で示された ADI 法以外にも幾つかの ADI 法があり、いずれの方法においても加速パラメータ ω_m の基準が重要な問題となる。

ω_m を決める簡単な方法として、 $\mathbf{H}, \mathbf{V}$ の固有値 λ, μ ($1 \leq i \leq n$) に対して、

α, β を次の関係式から定めておく。

$$① \quad 0 < \alpha \leq \min_{1 \leq i \leq n} (|\lambda_i|, |\mu_i|) \leq \max_{1 \leq i \leq n} (|\lambda_i|, |\mu_i|) \leq \beta$$

$$② \quad (\log \beta - \log \alpha) / 0.764 \text{ に最も近い整数 } 1 \text{ を求める。}$$

$$\textcircled{3} \quad \gamma = (\beta/\alpha)^{\frac{1}{2l}} \quad \text{として} \quad \omega_k = \alpha \gamma^{2k-1} \quad (1 \leq k \leq l)$$

から l 個のパラメータを求めて、反復計算が終了するまで繰り返して使用する。このとき平均的な収束の割合は、次の関係式が成立する。

$$R_m \left(\prod_{j=1}^m M_{\omega_j} \right) \geq \frac{2}{m} \sum_{k=1}^n \ln \left(\frac{\gamma^{2k-1} + 1}{\gamma^{2k-1} - 1} \right)$$

ADI 法は係数行列の次数が高く、分割幅が小さいとき、有効とされており、3変数の場合にも適用される。

ここで β はおよその数字で良いが、 α は $(\lambda_i), (\mu_i)$ の最小値の近い値を用いた方が良い。

[5] 最大固有値、最小固有値の計算

大きな行列計算では、加速パラメータの少しの違いが、計算時間に多大な影響を与える場合がある。

$\rho(\mathbf{M}_J)$ の計算方法を示す。

SOR 法では、最大固有値を求めるから

$$(2.76) \quad \mathbf{x}^{(m+1)} = \mathbf{M}_J \mathbf{x}^{(m)} \quad (m > 0)$$

を繰り返し、 $\frac{x_i^{(m+1)}}{x_i^{(m)}} \quad (1 \leq i \leq n)$ の最大固有値と最小固有値が一致するまで繰り返して計算する。

基本的な反復行列では、 $\mathbf{A}$ から導かれる反復行列（ヤコビ等）には、 $\rho^2(\mathbf{M}_J) = \rho(\mathbf{M}_C)$ が成立する。

$$(2.77) \quad \mathbf{x}^{(m+1)} = \mathbf{M}_C \mathbf{x}^{(m)} \quad (m > 0)$$

による反復計算の方が、収束が速い。

$\mathbf{H}$ の最小固有値を求めるには、 $\mathbf{H}^{-1}$ の最大固有値を計算することで良い。

式 (2.77) の代わりに

$$(2.78) \quad \mathbf{x}^{(m+1)} = \mathbf{H}^{-1} \mathbf{x}^{(m)} \quad (m > 0)$$

$$(2.79) \quad \mathbf{H} \mathbf{x}^{(m+1)} = \mathbf{x}^{(m)}$$

の式を解くことになる。得られた最大固有値の逆数が $\mathbf{H}$ の最小固有値となる。

収束を高める時、 $\frac{x_i^{(m+1)}}{x_i^{(m)}} \quad (1 \leq i \leq n)$ のうち最小のものを λ_m として

$$(2.80) \quad (\mathbf{H} - \lambda_m \mathbf{I}) \mathbf{x}^{(m+1)} = \mathbf{x}^{(m)}$$

の式を解くことにより求める事が出来る。

数値解法の歴史的発展、分類と特徴

3. 数値解法の歴史的発展

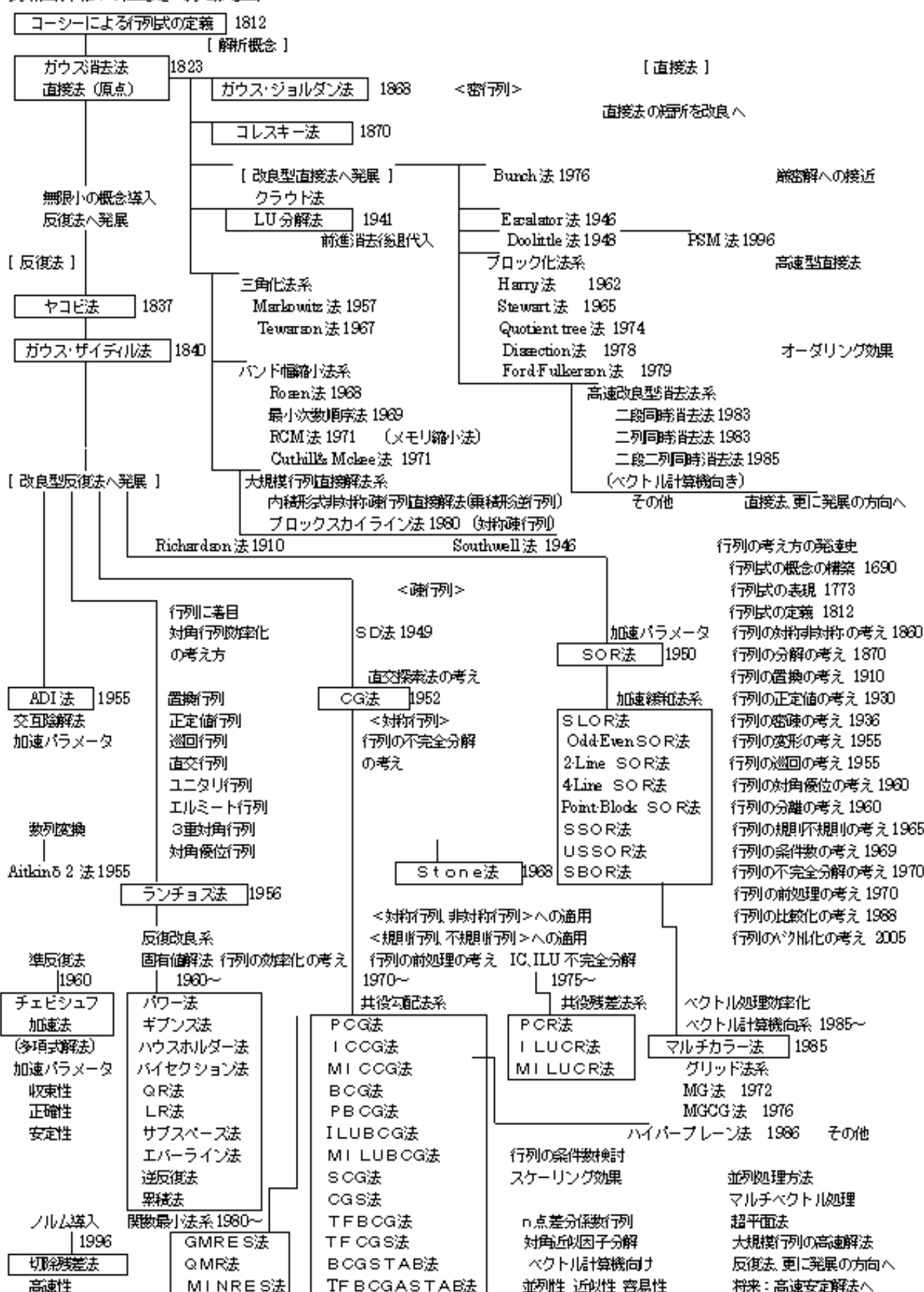
数値解法の歴史と分類と特徴において、数値計算の歴史的意義から将来へ発展して行く方向を図・表を用いて紹介したい。

行列の歴史的発展は、ニュートンの方程式を複数個並べて考えたライプニッツによる行列式の概念の表現である「行列式の原案の考察」(1690)に始まる。後にラグランジュは、ライプニッツの行列式を研究し、行列式の基礎的な考え方である「行列式の表現」(1773)を発表している。更に後の時代、コーシーは、ラグランジュ研究に始まり、行列の和、差、積、除（逆）を論じ、行列式体系である「行列式」(1812)を総合的に確立させた。コーシーと同時代のガウスは、数値計算としての方法論の考えを独立させた。そして「ガウス消去法」(1823)と呼ばれる比類なき大革命的な数値解法を提案した。これは、その後の行列におけるあらゆる数値計算の原点となっている。この「ガウス消去法」に、行列の数値解法におけるその源流を見いだす事が出来る。これは現在においても数値計算として、教育にも研究にも産業界にも広く使われている解法である。消去法と言う考え方の簡便性と、その数値解の厳密性は、現在に至っても他に例をみない。このガウス消去法は、後に直接法と定義され、ガウス・ジョルダン法(1868)、コレスキー法(1870)、LU 分解法(1941)等があり、近年においては構造力学の立場から発展したバンド幅縮小法系が幾つか提案され、そして高速解法直接法として Stewart 法(1965)、Dissection 分割法(1978)等を代表とするブロック化法系が提案された。更に最近では高速改良型消去法系として、二列同時消去法(1983)等の新しい直接法が次々と提案されて行く。

一方で、ガウスと同世代のヤコビは、フィボナッチが定義した無限小の概念を取り込み、連続的反復事象が極限迄続くと言う考え方を具体化させた。これが「ヤコビ法」(1837)として提案され、反復法として分類されて、後の時代に大きく発展して行く。ヤコビ法を発展させたガウス・ザイディル法(1840)を始めとして、このガウス・ザイディル法に加速パラメータを採り入れた SOR 法(1950)、そしてその SOR 法の改良や枝葉に位置づけられる SOR 法各種、直交探索の考えから発展した CG 法(1952)、その派生であり残差最小から研究された PCR 法(1975)を始めとする CR 法各種、交互陰解法としての ADI 法(1955)等が次々提案された。この反復法の発達と共に、固有値問題解法におけるランチョス法(1956)、ハウスホルダー法(1960)、パワー法(1960)等次々に新しい固有値解法が提案された。

近年では、前処理としての考え方が提唱され、CG 法に入る前に不完全な LU 分解等により、特に大型行列に対しては、疎な特徴を生かしたまま数値解を進めるとした考えである。これは PCG 法(1970)を始めとする CG 法系として発展し、その後多くの解法を生み出して行く。更にスーパーコンピュータが普及する時代になり、これに適した解法として、ベクトル処理に適したマルチカラー法(1985)や、ハイパープレーン法(1986)等の高速解法が提案された。最近ではスーパーコンピュータの並列化に発展方向があり、それに適したマルチベクトル処理関係の各種解法がいろいろ研究されている。これら数値解法の研究と発展を歴史的に整理した「数値解法の歴史的発展図」としてまとめている。更に、行列の解法には直接法と反復法があり、行列の性質を分類すると対称型と非対称型があり、これをクロスさせた表として、その数値解法の特徴を一言で表現した表を作り、「数値解法分類と特徴」をまとめた。

数値解法の歴史的発展図



数値解法分類と特徴

解法	対称型	特徴	非対称型	特徴
直接法	コレスキー法 改訂 コレスキー法 ブロック・スカイライン法 バンド幅縮小法 RCM 法 最小次数順序法 Dissection 法	密行列正確 LLt 分解 LDLt 分解 計算速度改良 大規模疎行列 帯行列 メモリ縮小型 疎行列 効率良 効率良	ガウス消去法 ガウス・ジョルダン法 Doolittle 法 二段同時消去法 二列同時消去法 LU 分解法 ブロック化法 Stewart 法 Ford-Fulkerson 法 Tewarson 法 乗積形逆行列法 P S M 法	密行列正確 対称行列も可 並列 三角分解、誤差影響小 計算速度速 正確性普通 クラフト分解 ブロック分割 反復より遅い 収束性工夫 疎行列 適量 安定性保証 小行列解法正確
反復法	Eberlein 法 QL 法 ベキ乗法 逆ベキ乗法 ハウスホルダー法 ランチョス法 バイセクション法 (2列法、2分法) ダニレフスキー法 レイ・リット同時反復法 ブロック反復法 Bathe 法 村田 Rutishauser 法 Determinant Search 法 サブスペース法 SD 法 CG 法系 PCG 法 ICCG 法 MICCG 法 SCG 法 MINRES 法 SYMMLQ 法	実対称行列 ユニタリ行列 LU 分解固有値解 安定解法 対角疎行列適正 3重対角行列 3重対角行列 特性方程式 3重対角行列 直交化行列 3重対角行列 コレスキー分解 直交化行列 小規模固有値解法 大規模固有値解法 最急降下法 大規模対称型 疎行列 正値対称行列 高速解法 3重対角 Meijerink Gustafsson U _t DU 不完全 U _t DU 非正定値 最小残差 非正定値 対称 LQ (巡回行列) (7重対角行列) (7重対角行列)	ガウスサイデル法 ヤコビ法、2列法、2行法 ギブンス法 多分法 逆反復法 累積法 QR 法 ダブル QR 法 LR 法 変形 LR 法 Stone 法 EVP 法 ADI 法 ADE 法 リチャードソン法 リーブマン法 サウスウエル法 CGNE 法 CGNR 法 SOR 法系 SSOR 法 USSOR 法 SLOR 法 Odd-Even SOR 法 Point-Block SOR 法 2-Line 4-Line SOR 法 CR 法系 PCR 法 ILUCR 法 MILUCR 法 BCG 法系 PBCG 法 QMR 法 TFBCG 法 ILUBCG 法 MILUBCG 法 GPBCG 法 BCGSTAB 法 CGS 法系 PCGS 法 CGSTAB 法 MRTR 法 GMRES 法 BILU-2列法 3重対角法 MG 法 Reduction 法 3重対角加速反復法 Sheldon 準反復法 超平面法	収束安定 条件数依存無 3重対角行列 符号条件法 原点移動法 固有値効率解法 相似変換 3重対角行列 収束効率化影響係数 陽解法、陰解法 楕円方程式 改良リチャードソン 残差緩和 非正定値 逐次過緩和法 加速パラメータにより 高速性が左右される LU 分解法付残差法 正値行列 双対共役勾配法 非対称 正値行列 大型不規則3重対角 CG 型三項漸化式 4重対角計算機向き 3重対角、高速解法 多項式展開収束 平均収束 並列処理方式 (1重対角行列)