

7. 直接法と反復法

7-1. 概説

直接法 (Direct-Method) : 係数行列の形は刻々と変化し、ゼロ要素もノンゼロ対象として逐次計算する為、中小規模の密行列の解法に適し、計算の安定性と厳密解が期待出来るが、計算時間がかかる。一定の形をした帯行列にも適している。ベクトル計算機の発達と共に各種の直接法改良版が提唱されている。古典的な直接法は収束過程において、行列の条件数(Condition number)に依存しない特長がある。

反復法 (Iterative-Method) : もとの行列の特徴を生かしながら、ある種の反復行列に変形し、初期値から出発して、反復計算によって求める。大規模な疎行列に適しており、反復方法によってかなりの高速性が期待出来るが、条件数が大きい場合など収束出来ず、不安定要因も混在している。定常型の反復法を陽解法(Explicit method)と呼び、もとの行列を分離して行う反復を陰解法(Implicit method)と呼ぶ。行列の前処理(Preconditioning)の考えの発達から、最近では各種の反復法が提唱されている。収束過程は行列の条件数に依存し、反復方法によっては収束が困難な場合がある。

収束 (Convergence) : 数値計算としての収束とは、数学的に定義すると行列 A の絶対値の最大固有値におけるスカラー半径(Spectral radius)が 1 未満に収まる事を意味する。

(行列 A における固有値 λ の特性多項式は A の要素から構成されるベクトル空間上で、互いに相異なる又は同一な n 個の零点を有する。この n 個の零点の集合体を A のスカラーと呼ぶ。ここで λ に付随する固有ベクトルより生成される部分空間(Sub-space)がスカラー半径である。)

収束には定常の打ち切りによる強収束と、ノルム(norm)の特徴を生かした弱収束がある。

(ノルムには Euclid norm, Spectral norm, Frobenius norm, Hilbert-Schmidt norm 等があるが、ここでは Euclid norm を意味する。)

行列の収束は、一般にその行列の持つ固有値の分布状況に密接に関連している。一般論では固有値分布がスムーズである場合、強収束の考えで良く、一方固有値分布が悪化傾向を示す場合、弱収束が望まれる。悪化傾向とは固有値の大小が極めて差がある場合を言う。例えば熱伝導率の媒体の係数が場所によって大幅に異なった場合や、物質の寸法や差分メッシュの細かさを部位によって大きく変化させる時、この数式行列における行列では固有値分布は悪化傾向を示す。これが極端である数式行列を作ると、数値解法そのものが困難となる。つまりここでは、あまりにも極端である数式行列の場合は考えず、程々の行列について考える。

強収束では長所として、考えや計算方法がシンプルである。打ち切り計算としての考え方の的確性の面で優れている。一方、短所として行列が良好でないケースであれば収束が困難な傾向がある。

弱収束では長所として計算過程が穏やかで、安定性に優れている。

Ill-condition な数式行列にも適用できる。相対残差ノルム形式にみる方式では、数

式モデル全体のバランス性を保ちながら、整合性を保って収束して行く事が出来る。条件数は比較的大きい場合でも収束が期待出来る。一方、短所として収束ステップの繰り返しが増える傾向がある。

これら収束に関する問題は、数式モデルとして作られる方程式の強表現の場合や、弱表現の場合における数値計算とは直接的な関係は見られない。

誤差 : 誤差には4つの型がある。(von Neumann , Goldstein による定義)

(Error) 一つの計算をする上で、以下の4つの誤差が常につきまとう。

モデル化誤差(Modelling error)、測定誤差(Measurement error)、

打ち切り誤差(Truncation error)、丸め誤差(Round off error)である。

モデル化誤差と測定誤差に関しては、現象を定式化したり測定したりする過程で吟味すべき事であり、現実と合わせた形へ如何に数式モデルを立てるか、実測するかによる問題である。打ち切り誤差と丸め誤差に関しては、数値アルゴリズムに含まれる数式の繋ぎから数値データへの変換と計算過程における計算方式での誤差であり、これらはハードウェアに多く依存し、数値データ形式表現によるその時の有限のビットにより左右されている。

つまり数値シミュレーションとは理論・実験で得られた解や現象に対して、近似式・近似値によって求めている事になる。理論解・厳密解に対してどこまで近似出来るかという問いに対して、幾つもの数式やそのアルゴリズムが研究され提案されている。

7-2. 連立一次方程式 (Linear Systems)

次の連立一次方程式を考える。

$$(1) \quad \begin{aligned} 2x_1 + x_2 + x_3 &= 9 \\ 2x_1 + 3x_2 + 5x_3 &= 17 \\ x_1 + x_2 + 3x_3 &= 8 \end{aligned}$$

式(1)の行列式表現を

$$(2) \quad \mathbf{Ax} = \mathbf{b}$$

とおく。

式(1)において

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 \\ 2 & 3 & 5 \\ 1 & 1 & 3 \end{pmatrix} \quad \text{の場合は非対称行列 : 流体解析、化学、原子力他}$$

$$\mathbf{A} = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 3 & 5 \\ 1 & 5 & 3 \end{pmatrix} \quad \text{の場合は対称行列 : 物理、構造解析、基礎化学、統計学重回帰他}$$

(数式モデル説明)

ここで、解析要求としての立場から式(2)の形式に導く為、例えば移流項のない熱拡散方程式等を取り上げて考えた場合、これはエネルギーを最小にする形の変分原理を満たしている。この時、差分システムを直角座標で定式化すれば、行列 \mathbf{A} は対称化出来るが、座標を円筒座標や曲線座標とすると、回転による微小微分角や交互微分項が現れ、行列 \mathbf{A} の対称性が崩れ、緩やかな非対称行列(Non-symmetric matrix)となる。

次に最近の行列における数値計算の研究概論を少し述べる。以下の事が言える。
行列 A の数値解法を効率的に解く為、次の M 行列 (M matrix) を定義する。

M 行列 $\{ M = (a_{ij}) ; a_{ii} > 0, a_{ij} \leq 0, A^{-1} \geq 0 \}$
(M 行列について) :

数値計算において、求めるべき行列が優対角行列(Diagonal Dominant Matrix)であれば、数値解が求まり易い事が知られている。行列 A が M 行列の性質を有する時、次の3点が言える。

対角優位な行列でなくても、数値解の安定性(stability)が得られる。

正則分離(Non-Singular separation)が可能な行列であれば、絶対値最大固有値のスペクトル半径が1未満となり、行列 A の正則分離に対応する反復解法は必ず収束する。

不完全LU分解が常に可能となり、大規模行列における計算の高速化が得られると共に、この時の反復解法は必ず収束する。

特に流体解析(二次元、三次元移流拡散問題の離散化等)に置いては、効率的な M 行列を求める為、セル数(あみ目幅、移流の強さと拡散係数の関係)を小さくしたり、対応するメッシュを細かくする傾向がある。

行列 A が M 行列であれば、数値解法が比較的容易である。数式から離散化を行う為に差分形式を考える時、この M 行列に少しでも速く近づける為、中心差分法より、風上差分法が優遇される傾向にある。

従来からの中心差分法(一次、二次以上を含めて)は、計算精度の面では良好な場合が多いが、反面安定収束の面で一考の余地が考えられる。

これに対して風上差分法(一次、二次以上を含めて)は、計算過程がより安定収束させる事が出来る傾向にあるが、反面計算精度の面では幾つかの問題点を含ませている。現実の数値解法には、解法アルゴリズム以外に、プログラミング技法、ハードウェアに依存する記憶容量、処理速度も問題になる。計算の高精度化を追求するあまり、差分法を二次にして離散化すると、近似式に使われる差分の点数が増えて、効率的な計算が困難となる。スーパーコンピュータで計算する上においては、ベクトル化率向上の立場から、又最近の並列化技術においては、なおさら計算に至るまでの準備に検討を要する。

一般論として、更新された新しいデータを順次使う計算方式では、収束値への反復回数は少なく済むが、それに引き替え計算のステップは再帰的となり、スーパーコンピュータの特長を發揮させるベクトル化が困難となる。一方で古いデータを使う計算方式では、スーパーコンピュータの特長を活かせるベクトル化に向いており、計算の効率性を追求し、ベクトル化率向上と共に、その場での見かけの高速計算を実現させる事が出来るが、それに引き替え収束の面から考えると、収束値迄の計算ステップがどうしても長くなり、数値シミュレーション全体における高速計算を追求して行く上で、必ずしもすべてが効率的とは言えず、この方式をとる事が最良の方法ではない。

幾つもの数値計算アルゴリズムは、いずれもこのような長短を持ち合わせている。

つまり、計算対象とする行列の大きさ、行列の規則不規則性(regulations, non-regulations)、スパース性(sparse)、行列の対称非対称(symmetric, non-symmetric)や条件数(Condition number)等も考慮した上で、それに応じた最適な解法を研究し、その上で可能な限りのベクトル化の技術が要求される。

特に共有メモリ統合型スーパーコンピュータ(SX4)を利用する場合においては、計算過程におけるアルゴリズムの収束速度と、ベクトルパイプラインの効率的利用、及び並列性の効用の両面を考え、数学的安定性を保ちながら、効率的に収束させて行く事が重要な課題となる。こうしたことから数値計算では、

計算対象とする数式行列や、計算対象とする行列自体の性質を見極めて、いろいろと使い分けて行く必要がある。

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、ガウスの消去法、反復法の代表、ヤコビ法を始めとする基礎的な解法に触れる。

以下、基礎的な数値解法の幾つかを、具体式と数値計算例、又は概略にて簡単に紹介する。

直接法

ガウスの消去法、ガウス・ジョルダン法、LU分解法、コレスキー法、ブロック化法

反復法

ヤコビ法、ガウス・ザイデル法、SOR法、CG法、ADI法、チェビシェフ反復法、ランチョス法、ICCG法、QR法

7-3. 数値解法

7-3-1. 直接法 計算の概要

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。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「オーダーリング」の考えが提案されている。

1) ガウスの消去法 (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ブニッツ(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)はライブニッツの「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思考(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。

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。ガウスの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。計算方式の並列性に特長があり、現在では並列計算の面で研究されている。

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。

$$(4) \quad \begin{array}{rcl} a_{11}x_1 & & = b_1 \\ & a_{22}x_2 & = b_2 \\ & & a_{33}x_3 & = b_3 \\ & & & \dots \end{array}$$

$$\begin{array}{rcl}
 a_{n-1,n-1}x_{n-1} & & = b_{n-1} \\
 \dots\dots\dots & & \\
 a_{nn}x_n & & = b_n
 \end{array}$$

式(4)は
$$\begin{cases} k = 1, 2, \dots, n \\ x_k = b_k \times d_k \end{cases}$$

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け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、ガウスの消去法が約 $\frac{1}{3}n^3$ 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、ガウスの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の消去法より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。

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

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

A が $x \neq 0$ なるすべてのn次元ベクトルに対して、

$$x^T A x > 0$$

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

$$a_{ii} > 0$$

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。

A を下三角行列と上三角行列の形をそれぞれ L と U で表現する。

A の対角行列を D で表現する。

$$D^{1/2} = \begin{pmatrix} \sqrt{a_{11}} & & & \\ & \sqrt{a_{22}} & & \\ & & \ddots & \\ & & & \sqrt{a_{nn}} \end{pmatrix}$$

$\tilde{L} = L D^{1/2}$ と置くことで

$A = \tilde{L} \tilde{L}^T$ で表現する事が出来る。

A が正定値でない場合、

(L の対角行列を 1 とおく分解としての表現をする)

$$\mathbf{U} = \mathbf{D}\mathbf{V}$$

で表現できる。

$$\text{但し、}\mathbf{V} = \begin{pmatrix} 1 & v_{12} & v_{13} & \cdot & v_{1n} \\ & 1 & v_{23} & \cdot & v_{2n} \\ & & 1 & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & 1 \end{pmatrix}$$

この時

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{V}$$

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

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

となる。

式(2)により

$$\mathbf{A}\mathbf{x} = \mathbf{b} \text{ から}$$

$$\mathbf{L}\mathbf{D}\mathbf{L}^T\mathbf{x} = \mathbf{b}$$

の形となる。

$$(5) \quad \mathbf{D}\mathbf{L}^T\mathbf{x} = \mathbf{y}$$

$$(6) \quad \mathbf{L}\mathbf{y} = \mathbf{b}$$

と置くことが出来、式(6)を解いて \mathbf{y} を求め、次に式(5)を解くことで数値解が求められる。

4) LU分解法 (LU factorization 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。ユース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。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。

\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。

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

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。

5) ブロック化法

代表的な解法に Stewart 法がある。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(1965)。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。大規模な構造解析等に活用が見いだされる。

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Aのブロック毎のLU分解を行う。

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

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

ここで \mathbf{I}_i は各単位行列である。

式(2)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。

ここで、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。

7-3-2. 反復法 計算プロセス概要

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。

$$\text{式(2)より} \quad \mathbf{Ax} = \mathbf{b}$$

これと同値な形を次に変形する。

$$(9) \quad \mathbf{x} = \mathbf{Mx} + \mathbf{c}$$

式(9)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。

$$(10) \quad \mathbf{x}^{(k+1)} = \mathbf{Mx}^{(k)} + \mathbf{c}$$

なる反復代入の計算ステップを構成させる。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。

6) 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。

ヤコビ法一般形

$$(11) \quad \mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

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

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

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

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。

7) 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。

ガウス・ザイデル法一般形

$$(13) \quad \mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{x}^{(k+1)} - \mathbf{F}\mathbf{x}^{(k)})$$

式(13)を変形すると

$$(14) \quad \mathbf{x}^{(k+1)} = -(\mathbf{D} + \mathbf{E})^{-1}\mathbf{F}\mathbf{x}^{(k)} + (\mathbf{D} + \mathbf{E})^{-1}\mathbf{b}$$

普通は式(14)が用いられる。

8) SOR法 (Successive Over Relaxation method) 加速緩和法

ガウス・ザイデル法の計算修正量に着目した Young と Frankel は厳密解に少しでも速く収束させる目的で、ガウス・ザイデル法を少し変形し、加速パラメータ ω を乗じて次の近似解への修正量を拡大し、収束を速めた方法が SOR 法である(1952)。応用事例分野として、時間積分を伴ったディバイズシミュレーションは巡回解法となる。この時は SOR 法としての解法が効果があり、多く用いられている。現在、一般には計算物理の分野に根強く人気がある。この SOR 法は枝葉となる解法がいろいろ研究され、SOR 法系(SLOR, SSOR, USSOR, SBOR, Odd-Even SOR 法 他)を構成している。

代表的な SOR 法は次の形となる。

SOR 法反復行列 \mathbf{M} は次式で表現出来る。

$$(15) \quad \mathbf{M} = (\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{E})^{-1}((1 - \omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{F})$$

SOR 法の一般形

$$(16) \quad \mathbf{x}^{(k+1)} = (\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{E})^{-1}((1 - \omega)\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{F})\mathbf{x}^{(k)} + \omega(\mathbf{D} + \omega\mathbf{E})^{-1}\mathbf{b}$$

加速パラメータ ω は一般に $0 < \omega < 2$ である。

収束する場合とは、反復行列 \mathbf{M} の固有値を λ とすると $\lambda^2 < 1$ である。

9) 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。現在、前処理法(Preconditioning Method)を伴った各種のCG法として研究され広く普及している。構造解析、流体解解析等に幅広く適用されている。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。

式(2)において、もし \mathbf{A} が非対称であるならば、式(2)の両辺に \mathbf{A} の転置行列 \mathbf{A}^T を乗じて

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

(正定値：固有値がすべて正であること)

式(2)の解を $\mathbf{z} (= \mathbf{A}^{-1}\mathbf{b})$ とする。

つぎの関数を定義する。

$$(17) \quad f(\mathbf{x}) = (\mathbf{x} - \mathbf{z}, \mathbf{A}(\mathbf{x} - \mathbf{z}))$$

式(17)において \mathbf{A} は正定値であるから、 $\mathbf{x} \neq \mathbf{z}$ の時

$$(\mathbf{x} - \mathbf{z})^T \mathbf{A}(\mathbf{x} - \mathbf{z}) > 0 \quad \text{つまり} \quad f(\mathbf{x}) > 0, \quad \mathbf{x} = \mathbf{z} \text{ の時} \quad f(\mathbf{x}) = 0$$

$f(\mathbf{x})$ 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。

$$(18) \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{P}^{(k)}$$

n次元空間でそれぞれ直交し、式(18)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。 $\alpha^{(k)}$ 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。

10) 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、核融合シミュレーションのフル解法等に使用されている。

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

$$(19) \quad \mathbf{A} = \mathbf{D} + \mathbf{H} + \mathbf{V}$$

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(2)を次の二つの式へと導く。

$$(\mathbf{H} + \frac{1}{2}\mathbf{D} + \omega\mathbf{I})\mathbf{x} = (\omega\mathbf{I} - \mathbf{V} - \frac{1}{2}\mathbf{D})\mathbf{x} + \mathbf{b}$$

(20)

$$(\mathbf{V} + \frac{1}{2}\mathbf{D} + \omega\mathbf{I})\mathbf{x} = (\omega\mathbf{I} - \mathbf{H} - \frac{1}{2}\mathbf{D})\mathbf{x} + \mathbf{b}$$

但し、 ω は加速パラメータ

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

とおいて、

式(20)は次の形で定義される。

$$(\mathbf{H}_1 + \omega_{i+1}\mathbf{I})\mathbf{x}^{i+\frac{1}{2}} = (\omega_{i+1}\mathbf{I} - \mathbf{V}_1)\mathbf{x}^i + \mathbf{b}$$

(21)

$$(\mathbf{V}_1 + \omega_{i+1}\mathbf{I})\mathbf{x}^{i+1} = (\omega_{i+1}\mathbf{I} - \mathbf{H}_1)\mathbf{x}^{i+\frac{1}{2}} + \mathbf{b}$$

式(21)は陰反復解法になる。ここで式(21)の左辺はガウシ消去法に基づき、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。フ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。

11) チェビシエフ反復法 (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エフ反復として提案した(1960)。チェビシエフはチェバロフ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。一般的な行列では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見られる。

M は $n \times n$ 行列収束行列 (Element Convergent Matrix $M^* = M$,
 M^* は M の複素共役転置行列)
 とする。 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。
 M のスペクトル半径 $\rho(M) < 1$ であるとする。

チェビシフ反復の基本的考え方

一般的な反復法を次に定義する。

$$(22) \quad \mathbf{x}^{(m+1)} = M\mathbf{x}^{(m)} + \mathbf{g} \quad , \quad (m \geq 1)$$

式(22)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。

$$(23) \quad \mathbf{y}^{(m)} = \sum_{k=0}^m v_k(m) \mathbf{x}^{(k)} \quad , \quad (m \geq 0)$$

式(23)を併用させて解く。

式(22)の反復行列を式(23)で代数的に結合させる為、準反復法と呼ばれる。

チェビシフ反復法は次の形をとる。

$$(24) \quad \mathbf{y}^{(m+1)} = \omega_{m+1} (M\mathbf{y}^{(m)} + \mathbf{g} - \mathbf{y}^{(m-1)}) + \mathbf{y}^{(m-1)} \quad , \quad m \geq 0$$

加速パラメータ ω_{m+1} は

$$(25) \quad \omega_{m+1} = 1 + \frac{C_{m-1}(1/\rho)}{C_{m+1}(1/\rho)} \quad , \quad m \geq 1 \quad , \quad \omega_1 = 1$$

である。

ここで $C_m(x)$ は次式により定義される。

$$(26) \quad C_m(x) = \begin{cases} \cos(m \cos^{-1} x) & , \quad -1 \leq x \leq 1 \quad , \quad m \geq 0 \\ \cosh(m \cosh^{-1} x) & , \quad x \geq 1 \quad , \quad m \geq 0 \end{cases}$$

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。

1.2) ランチョス法 (Lanczos Method)

ガウス行列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。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で効力を発揮する事が出来る。原子力工学の輸送方程式の解法などに応用事例が見つかる。

A は $n \times n$ 対称行列 直交行列 P によって B は3重対角行列とする。

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$$

この場合、A が対称であるから、B も対称となる。

$$(27) \quad \mathbf{B} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \alpha_{n-1} & \beta_{n-1} \\ \mathbf{0} & & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

P の第 k 列ベクトルを \mathbf{u}_k とすると、その直交性から次式が成立する。

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 0 & ; i \neq j \\ 1 & ; i = j \end{cases}$$

又直交変換であるため次式が成立する。

$$(28) \quad \mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{B}$$

式(28)の関係から、ベクトル \mathbf{u}_k を定めながら、同時に三重対角化を行って行く。

式(28)の右辺 B に式(27)を代入して各列を比較すると、

$$(29) \quad \begin{aligned} \mathbf{A}\mathbf{u}_1 &= \alpha_1 \mathbf{u}_1 + \beta_1 \mathbf{u}_2 \\ \mathbf{A}\mathbf{u}_2 &= \beta_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \beta_2 \mathbf{u}_3 \\ &\dots\dots\dots \\ \mathbf{A}\mathbf{u}_k &= \beta_{k-1} \mathbf{u}_{k-1} + \alpha_k \mathbf{u}_k + \beta_k \mathbf{u}_{k+1} \\ &\dots\dots\dots \\ \mathbf{A}\mathbf{u}_n &= \beta_{n-1} \mathbf{u}_{n-1} + \alpha_n \mathbf{u}_n \end{aligned}$$

式(29)において第 k 行目に左から \mathbf{u}_k^T を乗ざると直交性から次式が求められる。

$$\alpha_k = \mathbf{u}_k^T \mathbf{A}\mathbf{u}_k$$

ここで $\mathbf{u}_{k-1}, \mathbf{u}_k$ が既に求められていると、 \mathbf{u}_{k+1} は次の形から求められる。

$$\mathbf{v}_{k+1} = \mathbf{A}\mathbf{u}_k - \beta_{k-1} \mathbf{u}_{k-1} - \alpha_k \mathbf{u}_k$$

と置く事により

\mathbf{u}_{k+1} に正規化条件(Normalize condition) $\mathbf{u}_{k+1}^T \mathbf{u}_{k+1} = 1$ を満足させるため β_k を

$$\beta_k = \|\mathbf{v}_{k+1}\|_2 \quad \text{と定める。そして}$$

$$\mathbf{u}_{k+1} = \frac{1}{\beta_k} \mathbf{v}_{k+1}$$

とする。

この様にして $\|\mathbf{u}_1\|_2 = 1$ なる任意の初期ベクトル \mathbf{u}_1 から始めて順次 α_k, β_k を計算する事により三重対角行列を作る事が出来る。つまりもとの行列 A は変形を受けず、A とベクトルの積だけの計算が行われる。3項のみからなる漸化関係により、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{u}_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検討すべき余地が考えられる。

13) 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。Golub の死後、後の時代 Golub 研究者が提案した方法、対角行列を組み入れた修正 Cholesky 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。これは不完全 Cholesky 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 Cholesky 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。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。応用事例として、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多く見受けられる。

9) 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(2)において、行列 A の不完全コレスキー分解を行う。

U : 上三角行列、 U^T : 上三角行列の転置行列、 D : 対角行列

$A \approx U^T D U$ で表現される。

式(2)について次の形にする。

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

ここで、 $x' = (D^{1/2}U)x$ 、 $b' = (D^{1/2}U)^{-T}b$

初期近似ベクトル $x^{(0)}$ を選ぶ。

$$r^{(0)} = b - Ax^{(0)}$$

$$P^{(0)} = (U^T D U)^{-1} r^{(0)}$$

$k = 0, 1, 2, \dots$ 、 1) から 5) の繰り返し

$$1) \quad \alpha^{(k)} = \frac{(r^{(k)}, (U^T D U)^{-1} r^{(k)})}{(P^{(k)}, A P^{(k)})}$$

$$2) \quad x^{(k+1)} = x^{(k)} + \alpha^{(k)} P^{(k)}$$

$$3) \quad r^{(k+1)} = r^{(k)} - \alpha^{(k)} A P^{(k)}$$

$$4) \quad \beta^{(k)} = \frac{(r^{(k+1)}, (U^T D U)^{-1} r^{(k+1)})}{(r^{(k)}, (U^T D U)^{-1} r^{(k)})}$$

$$5) \quad P^{(k+1)} = (U^T D U)^{-1} r^{(k+1)} + \beta^{(k)} P^{(k)}$$

収束判定 定数 $\varepsilon > 0$ として、 $\sqrt{(r^{(k)}, r^{(k)})} < \varepsilon$ で収束

この時の $x^{(k)}$ が数値解となる。

14) 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ス(Francis)は、コレスキ-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 A を、直交行列 Q と右上三角行列 R に分解する方法を提唱した。ここで 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 A の 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、 A の左下半分は0に収束し、最後に対角線上に A の固有値が並ぶ。これをQR法として提案した(Computer Journal)(1961)。フランス(Francis)と仲の良い研究者グラボラス加(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。代表的な応用例として、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。

行列 A を直交行列 Q と右上三角行列 R の積に分解する。

$$(30) \quad A = QR$$

$k=1,2,\dots$ として繰り返し

$$A_k = Q_k R_k$$

$$A_{k+1} = R_k Q_k$$

$$(32) \quad A_k = P_{k-1}^{-1} A P_{k-1}$$

式(32)の十分大きな n に対して、 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 A が非対称であってもQR法は収束する。

15) 前処理ILU法に伴う考察

1) ILUBCG法 (Incomplete LU Biconjugate Gradient Method)の見解

式(2)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正値対称行列 A に対して、前処理として不完全なコレスキ-分解したものをICCG法と呼ばれる。正値対称行列 A に対して、完全 $U^T D U$ 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-分解を行う考えである一般的なPCG法を取り上げる。正値対称行列 A に対して、前処理として不完全コレスキ-分解したものを考える。

A に対して、 $A = U^T U + N$ 又は $A = U^T D U + N$ の不完全コレスキ-分解をする。計算手順として、

$$r_0 = b - Ax_0, \quad q = (U^T U)^{-1} r_0, \quad p_0 = q$$

となる。

ICCG法は $(U^T U)^{-1}$ であるが、この時の $(U^T U)^{-1}$ の代わりに $(U^T D U)^{-1}$ とおけばMICCG法になる。

<CR法との相違>

行列 A の不完全LU分解 $A = LU + R$ 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 $\mathbf{A} = \mathbf{LU} + \mathbf{R}$ した後、共役残差法を適用する。 $\mathbf{Ax} = \mathbf{b}$ の両辺に $(\mathbf{LU})^{-1}$ を乗じて、 $(\mathbf{LU})^{-1}\mathbf{Ax} = (\mathbf{LU})^{-1}\mathbf{b}$ を得る。この $(\mathbf{LU})^{-1}\mathbf{A}$ 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 CR 法が収束するためには \mathbf{A} の対称部 $\mathbf{S} = (\mathbf{A} + \mathbf{A}^T) / 2$ が正定値になる必要がある。

BCG 法の計算ステップが、

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{b} - \mathbf{Ax}_0 \\ \alpha_k &= (\mathbf{r}_k, \mathbf{r}_k^*) / (\mathbf{Ap}_k, \mathbf{p}_k^*) \quad \text{と進めるのに対して、} \end{aligned}$$

ILUBCG 法は計算ステップが、

$$\begin{aligned} \mathbf{r}_0 &= (\mathbf{LU})^{-1}\mathbf{b} - \mathbf{Ax}_0 \\ \alpha_k &= (\mathbf{r}_k, \mathbf{r}_k^*) / ((\mathbf{LU})^{-1}\mathbf{Ap}_k, \mathbf{p}_k^*) \quad \text{とおく事で求められる。} \end{aligned}$$

不完全 LU 分解のアルゴリズム

$$G_A = \{(i, j); a_{i,j} \neq 0\}$$

として、 $G \supseteq G_A$ なる集合 G を決めておき、 \mathbf{A} を LU 分解するとき、 L 、 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。

$$\begin{aligned} k &= 1, 2, 3, \dots \\ j &= 1, 2, 3, \dots, k-1 \\ \text{if } (k, j) \in G & \text{ then} \\ l_{kj} &= (a_{kj} - \sum_{l=1}^{j-1} l_{kl} u_{lj}) / u_{jj} \\ l_{kk} &= 1 \\ u_{kk} &= a_{kk} - \sum_{l=1}^{k-1} l_{kl} u_{lk} \\ j &= k+1, n \\ \text{if } (k, j) \in G & \text{ then} \\ u_{kj} &= a_{kj} - \sum_{l=1}^{k-1} l_{kl} u_{lj} \end{aligned}$$

continue

双対共役勾配法は式(2)に対して、次の双対な式を組み合わせる

$$(33) \quad \mathbf{A}^T \mathbf{x}^* = \mathbf{b}^*$$

式(2)と(33)より

BCG 法アルゴリズム

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{b} - \mathbf{Ax}_0 \\ \mathbf{r}_0^* &= \mathbf{b}^* - \mathbf{A}^T \mathbf{x}_0^* \\ \mathbf{p}_0 &= \mathbf{r}_0 \\ \mathbf{p}_0^* &= \mathbf{r}_0^* \\ i &= 0, 1, 2, \dots \end{aligned}$$

$$\alpha_i = \frac{(\mathbf{r}_i, \mathbf{r}_i^*)}{(\mathbf{Ap}_i, \mathbf{p}_i^*)}$$

$$\begin{aligned}
\mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i \mathbf{p}_i \\
\mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_{i+1} \mathbf{A} \mathbf{p}_i \\
\mathbf{r}_{i+1}^* &= \mathbf{r}_i^* - \alpha_i \mathbf{A} \mathbf{p}_i \\
\mathbf{r}_{i+1}^* &= \mathbf{r}_i^* - \alpha_i \mathbf{A}^T \mathbf{p}_i^* \\
\beta_i &= \frac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1}^*)}{(\mathbf{r}_i, \mathbf{r}_i^*)} \\
\mathbf{p}_{i+1} &= \mathbf{r}_{i+1} + \beta_i \mathbf{p}_i \\
\mathbf{p}_{i+1}^* &= \mathbf{r}_{i+1}^* + \beta_i \mathbf{p}_{i+1} \\
i &= i+1 \\
\text{continue}
\end{aligned}$$

前処理に対する代表的な流儀を4つ紹介したい。

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) を取り上げ、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。いま \mathbf{A} に近い正値対称行列 $\tilde{\mathbf{M}}$ 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 $\tilde{\mathbf{M}}$ には \mathbf{A} の \mathbf{U} を用いて、 $\tilde{\mathbf{M}} = \mathbf{U}^T \mathbf{U}$ とした $\mathbf{U}^T \mathbf{U}$ のコレスキー分解を施す点である。もとの方程式に両辺から \mathbf{U}^{-T} を掛けると

$$\begin{aligned}
\mathbf{U}^{-T} \mathbf{A} \mathbf{x} &= \mathbf{U}^{-T} \mathbf{b} \quad , \quad \tilde{\mathbf{x}} = \mathbf{U} \mathbf{x} \text{ とおけば、} \\
\mathbf{U}^{-T} \mathbf{A} \mathbf{U}^{-1} \tilde{\mathbf{x}} &= \mathbf{U}^{-T} \mathbf{b} \quad \mathbf{U}^{-T} \mathbf{A} \mathbf{U}^{-1} = \tilde{\mathbf{A}}
\end{aligned}$$

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

$$\begin{aligned}
a_{ij} &= u_{ii} + \sum_{k=1}^{i-1} u_{ki} d_k u_{kj} \quad \text{で計算され、} \\
d_i &= u_{ii}^{-1} \quad \text{つまり、} u_{ii} d_i = 1 \quad \text{となる値にした。} \\
G &\supseteq \{(i, j) ; a_{ij} \neq 0\}
\end{aligned}$$

のような格子点集合 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。

$$d_i^{-1} = a_i - b_{i-1}^2 d_{i-1} - c_{i-1}^2 d_{i-n} - e_{i-mn}^2 d_{i-mn}$$

$$a_{ij} = u_{ii} + \sum_{k=1}^{i-1} u_{ki} d_k u_{kj} \quad \text{で計算され、}$$

\mathbf{A} の不完全ガウスを次式で定義する。(Meijerink 流の不完全ガウスの方法による)

$$\mathbf{A} = \mathbf{L} \tilde{\mathbf{D}} \mathbf{U} - \mathbf{R}$$

この \mathbf{LDU} を使用して、不完全 \mathbf{LDU} の逆を両辺から掛ける。

$$(\mathbf{LDU})^{-1} \mathbf{A} \mathbf{x} = (\mathbf{LDU})^{-1} \mathbf{b}$$

Meijerink 流の考え方では、例えば7点差分の行列の場合、 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、

$$(34) \quad \tilde{d}_i^{-1} = d_i - a_i g_{i-mn} \tilde{d}_{i-mn} - b_i f_{i-m} \tilde{d}_{i-m} - c_i e_{i-1} \tilde{d}_{i-1}$$

とする形である。これに対して、BCG法やCR法を適用する。

上式に対して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(2)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、 $U^T D U$ 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 A にはならない事が分かっている。 $U^T D U$ 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。(34)式の d_i^{-1} の右辺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 $U^T D U$ を作る。こうした方法で Meijerink の数値計算を改良した。Gustafsson は式(34)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。

$$(35) \quad \tilde{d}_i^{-1} = (1 + \varepsilon) d_i - a_i g_{i-mn} \tilde{d}_{i-mn} - b_i f_{i-m} \tilde{d}_{i-m} - c_i e_{i-1} \tilde{d}_{i-1} \\ - u \left\{ a_i \tilde{d}_{i-mn} (e_{i-mn} + f_{i-mn}) + b_i \tilde{d}_{i-m} (e_{i-m} + g_{i-m}) + c_i \tilde{d}_{i-1} (f_{i-1} + g_{i-1}) \right\}$$

式(35)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 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(2)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

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 I L U B C G 法や I L U C R 法に対して、Gustafsson 流は M を付けた M I L U B C G 法や M I L U C R 法と名付けられ、その効果の故普及している。

4) 速水流 :

C G 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 $(LDL^{-T})^{-1}r$ の計算がベクトル計算機の高速度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 I C C G 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。I C C G 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、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。

A を正値対称行列とする。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。A の左右から 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ルリングと定義する。A をスケールリングするにおいて、A の対角行列に着目する。対角行列を D とおくと、ここで $D^{-\frac{1}{2}}$ を定義し、 $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ ($=\tilde{A}$) の形に変形する。この時、 \tilde{A} も正値対称行列となる。

$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}D^{\frac{1}{2}}x = D^{-\frac{1}{2}}b \quad \text{で、} \quad \tilde{x} = D^{\frac{1}{2}}x \quad , \quad \tilde{b} = D^{-\frac{1}{2}}b \quad \text{とおく事で、} \quad \tilde{A}\tilde{x} = \tilde{b}$$

この時の \tilde{A} はもとの 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 S C G 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、A の両辺から掛ける事で (スケールリングを行う)、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られる事が出来る。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 S C G 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。S C G 法は対称行列を取り扱うが、活用方法として A が非対称行列の場合、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 S C G 法を使う事が出来る。

S C G 法アルゴリズム

$$\mathbf{D}^{-1} = \begin{pmatrix} 1/a_{11} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & 1/a_{nn} \end{pmatrix} \quad \text{とおく。}$$

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、 \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。

$$\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

$$\tilde{\mathbf{x}} = \mathbf{D}^{\frac{1}{2}} \mathbf{x}$$

$$\tilde{\mathbf{b}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{b}$$

$$\tilde{\mathbf{r}}_1 = \tilde{\mathbf{b}} - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_1$$

$$\tilde{\mathbf{p}}_1 = \mathbf{D}^{-1} \tilde{\mathbf{r}}_1$$

$i = 1, 2, 3, \dots$

$$\alpha_i = \frac{(\tilde{\mathbf{r}}_i, \mathbf{D}^{-1} \tilde{\mathbf{r}}_i)}{(\tilde{\mathbf{p}}_i, \tilde{\mathbf{A}} \tilde{\mathbf{p}}_i)}$$

$$\tilde{\mathbf{x}}_{i+1} = \tilde{\mathbf{x}}_i + \alpha_i \tilde{\mathbf{p}}_i$$

$$\tilde{\mathbf{r}}_{i+1} = \tilde{\mathbf{r}}_i - \alpha_i \tilde{\mathbf{A}} \tilde{\mathbf{p}}_i$$

$$\tilde{\mathbf{r}}_i = \tilde{\mathbf{b}} - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_i$$

$$\beta_i = \frac{(\tilde{\mathbf{r}}_{i+1}, \mathbf{D}^{-1} \tilde{\mathbf{r}}_{i+1})}{(\tilde{\mathbf{r}}_i, \mathbf{D}^{-1} \tilde{\mathbf{r}}_i)}$$

$$\tilde{\mathbf{p}}_{i+1} = \mathbf{D}^{-1} \tilde{\mathbf{r}}_{i+1} + \beta_i \tilde{\mathbf{p}}_i$$

$i = i + 1$

continue

5) 原田・土肥流 :

C G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。原田・土肥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P B C G法や前処理付き共役残差法系P C R法では不完全L U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L U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变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L U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。例えば2次元5点差分の場合、前進代入として

$$\mathbf{v}_{i,j} = (\mathbf{g}_{i,j} - \mathbf{l}_{i,j-1} \cdot \mathbf{v}_{i,j-1} - \mathbf{l}_{i-1,j} \cdot \mathbf{v}_{i-1,j}) / \mathbf{l}_{i,j}$$

を定義すると、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 $v_{i-1,j}, v_{i,j}$ の間に参照関係から並列処理が出来ない。しかし $i+j = \text{一定}$ となる $v_{i,j}$ 、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、 A を対角行列 D と x 方向微分に関する非対角要素からなる 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 A_y を定義した。

$$A = D + A_x + A_y$$

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。

ベクトル計算機の高速度性に向けて、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。

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。

$$M_{TF} = L_x U_x D^{-1} L_y U_y$$

これにより逆行列計算が容易に行える。

近似性

近似誤差行列 R_{TF} を定義して、前処理行列 M_{TF} とおく

$$\begin{aligned} R_{TF} &= M_{TF} - A \\ &= A_x D^{-1} A_y \end{aligned}$$

つまり前処理行列 M_{TF} は A に似た行列が望ましい点。

並列性

並列計算機を特徴とする計算機では、 M_{TF}^{-1} が計算機に適した並列性を持つことが重要となる。 x 方向因子の近似計算として、

$$(D + \omega A_x) v = g$$

を定義する。これは A_x と 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、大きな効果をもたらす事が出来、大規模な行列に対しても、効果をもたらず事が顕著である。収束の面では ILUBCG 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。

TF 法アルゴリズム

$$r_0 = b - Ax_0$$

$$p_0 = M^{-1} r_0$$

$$p_0^* = r_0^* = r_0$$

$$i = 0, 1, 2, \dots$$

$$\alpha_{i+1} = \frac{(r_i, r_i^*)}{(Ap_i, p_i^*)}$$

$$x_{i+1} = x_i + \alpha_{i+1} p_i$$

$$\begin{aligned}
\mathbf{r}_{i+1} &= \mathbf{r}_i - \alpha_{i+1} \mathbf{A} \mathbf{p}_i \\
\mathbf{r}_{i+1}^* &= \mathbf{r}_i^* - \alpha_{i+1} (\mathbf{M}^{-1})^T \mathbf{A}^T \mathbf{p}_i \\
\beta_{i+1} &= \frac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1}^*)}{(\mathbf{r}_i, \mathbf{r}_i^*)} \\
\mathbf{p}_{i+1} &= \mathbf{M}^{-1} \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{p}_i \\
\mathbf{p}_{i+1}^* &= \mathbf{r}_{i+1}^* + \beta_{i+1} \mathbf{p}_{i+1} \\
i &= i+1 \\
\text{continue}
\end{aligned}$$

6) 展望 :

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が）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法を取り上げて、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。

< はより速い処理速度を表す。

ガウス	<	ヤコビ	<	ガウスザイデル	<	SOR	<	チェビシェフ	<	CG	<	前処理付CG	<	今後の研究
法		法		法		法		法		法		法		
		"				< ADI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(2)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 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、ステイ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(2)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(2)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(詳細省略)。

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、変化して行くため、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できない点も判明している。これらの点からいろいろな改良方法が研究されている。