数値計算セミナー資料
2002.11.18.

最近の数値解法のご紹介

NEC 曽谷勝義

前処理ＩＬU法に伴う考察

連立一次方程式

$$\mathbf{Ax}=\mathbf{b} \qquad (1)$$

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

式(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正値対称行列 $\mathbf{A}$ 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。正値対称行列 $\mathbf{A}$ に対して、完全 $\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ 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 法を取り上げる。正値対称行列 $\mathbf{A}$ 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、$\mathbf{A}=\mathbf{U}^{\mathrm{T}}\mathbf{U}+\mathbf{N}$　又は$\mathbf{A}=\mathbf{U}^{\mathrm{T}}\mathbf{DU}+\mathbf{N}$ の不完全コレスキー分解をする。

計算手順として、

$\mathbf{r}_0=\mathbf{b}-\mathbf{Ax}_0$ ， $\mathbf{q}=\left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)^{-1}\mathbf{r}_0$，　$\mathbf{p}_0=\mathbf{q}$

となる。

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

―＜ＣＲ法との相違＞―

行列 $\mathbf{A}$ の不完全 LU 分解　$\mathbf{A}=\mathbf{LU}+\mathbf{R}$ 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　$\mathbf{A}=\mathbf{LU}+\mathbf{R}$ した後、共役残差法を適用する。　$\mathbf{Ax}=\mathbf{b}$ の両辺に$\left(\mathbf{LU}\right)^{-1}$を乗じて、$\left(\mathbf{LU}\right)^{-1}\mathbf{Ax}=\left(\mathbf{LU}\right)^{-1}\mathbf{b}$　を得る。

この$\left(\mathbf{LU}\right)^{-1}\mathbf{A}$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 $\mathbf{S}=(\mathbf{A}+\mathbf{A}^{\mathrm{T}})/2$が正定値になる必要がある。

BCG 法の計算ステップが、

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

$\alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / (\mathbf{A}\mathbf{p}_k, \mathbf{p}_k^*)$　と進めるのに対して、

ILUBCG 法は計算ステップが、

$\mathbf{r}_0 = (\mathbf{LU})^{-1}\mathbf{b} - \mathbf{A}\mathbf{x}_0$

$\alpha_k = (\mathbf{r}_k, \mathbf{r}_k^*) / ((\mathbf{LU})^{-1}\mathbf{A}\mathbf{p}_k, \mathbf{p}_k^*)$　とおく事で求められる。

不完全ＬＵ分解のアルゴリズム

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

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。

$k = 1,2,3,\cdots\cdots\cdots$

$j = 1,2,3,\cdots\cdots, k-1$

if　$(k, j) \in G$　then

$l_{kj} = (a_{kj} - \sum_{l=1}^{j-1} l_{ki} u_{ij}) / u_{jj}$

$l_{kk} = 1$

$u_{kk} = a_{kk} - \sum_{l=1}^{k-1} l_{ki} u_{ik}$

$j = k+1, n$

if　$(k, j) \in G$　then

$u_{kj} = a_{kj} - \sum_{l=1}^{k-1} l_{ki} u_{ij}$

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

式(1)と(33)より

ＢＣＧ法アルゴリズム

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

$\mathbf{r}_0^* = \mathbf{b}^* - \mathbf{A}^T\mathbf{x}_0^*$

$\mathbf{p}_0 = \mathbf{r}_0$

$\mathbf{p}_0^* = \mathbf{r}_0^*$

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

$\alpha_i = \dfrac{(\mathbf{r}_i, \mathbf{r}_i^*)}{(\mathbf{A}\mathbf{p}_i, \mathbf{p}_i^*)}$

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

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

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

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

$\beta_i = \dfrac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1}^*)}{(\mathbf{r}_i, \mathbf{r}_i^*)}$

$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_i \mathbf{p}_i$

$\mathbf{p}_{i+1}^* = \mathbf{r}_{i+1}^* + \beta_i \mathbf{p}_{i+1}^*$

i=i+1

continue

前処理に対する代表的な流儀、並びに最新の解法を含めて８つ紹介したい。

１） 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$\mathbf{M}$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、$\mathbf{M}=\mathbf{U}^{\mathrm{T}}\mathbf{U}$ とした$\mathbf{U}^{\mathrm{T}}\mathbf{U}$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$\mathbf{U}^{-\mathrm{T}}$を掛けると

$\mathbf{U}^{-\mathrm{T}}\mathbf{A}\mathbf{x}=\mathbf{U}^{-\mathrm{T}}\mathbf{b}$, $\tilde{\mathbf{x}}=\mathbf{U}\mathbf{x}$とおけば、

$\mathbf{U}^{-\mathrm{T}}\mathbf{A}\mathbf{U}^{-1}\tilde{\mathbf{x}}=\mathbf{U}^{-\mathrm{T}}\mathbf{b}$ $\mathbf{U}^{-\mathrm{T}}\mathbf{A}\mathbf{U}^{-1}=\tilde{\mathbf{A}}$

この時、$\mathbf{U}^{-\mathrm{T}}\mathbf{A}\mathbf{U}^{-1}$ $(=\tilde{\mathbf{A}})$ の固有値は $\mathbf{A}$ に比較して１に近くなる。

Meijerink は$\mathbf{U}^{\mathrm{T}}\mathbf{U}$ 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$\mathbf{U}^{\mathrm{T}}\mathbf{D}\mathbf{U}$ 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$\mathbf{U}^{\mathrm{T}}\mathbf{U}$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

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

$d_i=u_{ii}^{-1}$ つまり、$u_{ii}d_i=1$ となる値にした。

$G\supseteq\{(i,j)\ ;\ a_{ij}\neq 0\}$

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。

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

$a_{ij}=u_{ii}+\sum_{k=1}^{i=1}u_{ki}d_k u_{kj}$ 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、 $\mathbf{A}$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、

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

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。

２） 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、$\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ 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。$\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ 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 $\mathbf{U}^{\mathrm{T}}\mathbf{DU}$ 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)
$$\tilde{d}_i^{-1} = (1+\varepsilon)d_i - a_i g_{i-mn}\tilde{d}_{i-mn} - b_i f_{i-m}\tilde{d}_{i-m} - c_i e_{i-1}\tilde{d}_{i-1} - u\left\{a_i\tilde{d}_{i-mn}(e_{i-mn}+f_{i-mn}) + b_i\tilde{d}_{i-m}(e_{i-m}+g_{i-m}) + c_i\tilde{d}_{i-1}(f_{i-1}+g_{i-1})\right\}$$

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 $\mathbf{M}$ 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(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 $\mathbf{M}$ 行列を用いる事で数値解法として安定性にが向上し、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Mを付けたＭＩＬＵＢＣＧ法やＭＩＬＵＣＲ法と名付けられ、その効果の故普及している。

３） 速水流：

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 $(\mathbf{LDL}^{-\mathrm{T}})^{-1}\mathbf{r}$ の計算がベクトル計算機の高速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、**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。対角行列を着目し、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リングと定義する。**A** をスケー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 $\mathbf{D}^{-\frac{1}{2}}$ を定義し、$\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \quad (=\tilde{\mathbf{A}})$ の形に変形する。この時、$\tilde{\mathbf{A}}$ も正値対称行列となる。 $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{D}^{\frac{1}{2}}\mathbf{x}=\mathbf{D}^{-\frac{1}{2}}\mathbf{b}$ で、$\tilde{\mathbf{x}}=\mathbf{D}^{\frac{1}{2}}\mathbf{x}$ ， $\tilde{\mathbf{b}}=\mathbf{D}^{-\frac{1}{2}}\mathbf{b}$ とおく事で、$\tilde{\mathbf{A}}\tilde{\mathbf{x}}=\tilde{\mathbf{b}}$ この時の $\tilde{\mathbf{A}}$ 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、**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 **A** が非対称行列の場合、**A** の転値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。

ＳＣＧ法アルゴリズム

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

ここで理論的には **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 **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,·····

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

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

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

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

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

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

i=i+1

continue

４） 原田流：

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。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。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 $\mathbf{v}_{i-1,j}, \mathbf{v}_{i,j}$ 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。

$$\mathbf{A} = \mathbf{D} + \mathbf{A}_x + \mathbf{A}_y$$

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 (Tri-diagonal Approximate Factorization Method)と名付けた。

ベクトル計算機の高速性に向けて、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。

$$\mathbf{M}_{TF} = \mathbf{L}_x \mathbf{U}_x \mathbf{D}^{-1} \mathbf{L}_y \mathbf{U}_y$$

これにより逆行列計算が容易に行える。

②近似性

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

$$\mathbf{R}_{TF} = \mathbf{M}_{TF} - \mathbf{A}$$
$$= \mathbf{A}_x \mathbf{D}^{-1} \mathbf{A}_y$$

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、$\mathbf{M}_{TF}^{-1}$ 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、

$$(\mathbf{D} + \omega \mathbf{A}_x)\mathbf{v} = \mathbf{g}$$

を定義する。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、効果をもったらす事が顕著である。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

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

$\mathbf{p}_0 = \mathbf{M}^{-1}\mathbf{r}_0$

$\mathbf{p}_0^* = \mathbf{r}_0^* = \mathbf{r}_0$

i = 0,1,2,·····

$$\alpha_{i+1} = \frac{(\mathbf{r}_i, \mathbf{r}_i^*)}{(\mathbf{A}\mathbf{p}_i, \mathbf{p}_i^*)}$$

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

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

$\mathbf{r}_{i+1}^* = \mathbf{r}_i^* - \alpha_{i+1}(\mathbf{M}^{-1})^T \mathbf{A}^T \mathbf{p}_i^*$

$$\beta_{i+1} = \frac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1}^*)}{(\mathbf{r}_i, \mathbf{r}_i^*)}$$

$\mathbf{p}_{i+1} = \mathbf{M}^{-1}\mathbf{r}_{i+1} + \beta_{i+1}\mathbf{p}_i$

$\mathbf{p}_{i+1}^* = \mathbf{r}_{i+1}^* + \beta_{i+1}\mathbf{p}_{i+1}^*$

i=i+1

continue

4-1. 前処理付き反復法を持つ三項対角近似因子分解法　(Tri-diagonal Approximate Factorization Method)

解くべき方程式を次に定義する。　　$\mathbf{A}$　は　n×n

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}$$

三項対角近似因子分解の計算上における並列性に関して、参考文献[5]に詳細が記載されているが、後述の提案への理解を促すためここで簡単に紹介したい。

$\mathbf{M}_{ILU}^{-1}$ 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ﾊﾟ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、ﾍﾞｸﾄﾙ計算機はﾘｽﾄﾍﾞｸﾄﾙを用いて強制的にﾍﾞｸﾄﾙ化する事が行われる。しかしﾘｽﾄﾍﾞ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、ﾍﾞ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、計算ﾉｰﾄﾞのｸﾞﾘｯﾄﾞﾎﾟｲﾝﾄが斜め方向並列計算として進む。 (Fig.3.参照)

不完全コレスキー分解法

1950 年 Young と Frankel はｶﾞｳｽｻﾞｲﾃﾞﾙ法に加速ﾊﾟﾗﾒｰﾀを加味し漸近収束率を高め、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、枝葉として SLOR 法、SSOR 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[16]属するが、計算物理の分野に見受けられ、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性化傾向があったり、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。

５） 土肥流： 1998/7/17

Ｏｄｄ－Ｅｖｅｎ ＳＯＲ法にみる多色オーダー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。
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 $\mathbf{S}_1,\mathbf{S}_2,\cdots\cdots,\mathbf{S}_{nc}$ を $\mathbf{S}_l=\{(i,j,k)\mid \mathrm{mod}(i+j+k-3,nc)+1=l\}$ とおき、格子点を $\mathbf{S}_1$ に含まれるものから順に $\mathbf{S}_{nc}$ をかけて順序付けする。
この Ordering に従い係数行列 $\mathbf{A}$ を構成すると、下記ブロック対角部
$\mathbf{D}l\ (l=1,2,.....,nc)$ 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

$$\left[\begin{array}{cccccccc|c}
D_1 & U_{1,2} & 0 & 0 & 0 & \cdot & \cdot & U_{1,nc} & \langle S_1 \\
L_{2,1} & D_2 & U_{2,3} & 0 & 0 & \cdot & \cdot & 0 & \langle S_2 \\
0 & L_{3,2} & D_3 & & & \cdot & \cdot & 0 & \langle S_3 \\
0 & 0 & \cdot & & & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & & & & \cdot & \cdot & \cdot \\
0 & 0 & 0 & \cdot & & & \cdot & 0 & \cdot \\
0 & 0 & 0 & 0 & \cdot & & D_{nc-1} & U_{nc-1,nc} & \langle S_{nc-1} \\
L_{nc,1} & 0 & 0 & 0 & \cdot & \cdot & U_{nc,nc-1} & D_{nc} & \langle S_{nc}
\end{array}\right]$$

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 $\mathbf{A}$ 初期値 $\mathbf{u}_0$ 右辺 $\mathbf{b}$ をサブルーチン内部で多色 Ordering の順に並び替え $\mathbf{A}$, $\mathbf{b}$ 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 $(u_i)_{1 \le i \le n}$ も作業配列 $(w_i)_{1 \le i \le n}$ を用いて次のように並び替える

$i_z = 1$ **から** n_z **に対して**
　$i = 1$ **から** n_{xy}
　　$w_{(nxy+inc)(iz-1)+i} = v_{nxy}(i_z - 1) + 1$
$i_z = 1$ **から** $n_x - 1$ **に対して**
　$j = 1$ **から** mc **に対して**
　　$v_{mc(ix-1)+1} = w_{ix+(ix-1)(nx-1)}$

ここで $n_{xy} = n_x \cdot n_y$ であり、 inc は $\mathrm{mod}(n_x + inc - 1, n_x - 1) = 0$ 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図（１）で示すように各色の成分が $n_x - 1$ 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 $\mathbf{S}_{nx-1}$ 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 $n = n_x(n_y + 1)n_z$ としておけば十分である。

６） Axelsson 流：

アクレッソンは行列 $\mathbf{A}$ 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。 $\mathbf{A}$ 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、上三角行列を取り上げる。

SSOR 前処理行列 $\mathbf{M}_{SSOR}$ は加速パラメータ ω （$0 \leqq \omega \leqq 2$）を用いて
$\mathbf{M}_{SSOR} = (\mathbf{A}_D + \omega \mathbf{A}_L)\mathbf{A}_D^{-1}(\mathbf{A}_D + \omega \mathbf{A}_U)$ で定義される。

特長： $\mathbf{A}_D+\omega\mathbf{A}_L$、 $\mathbf{A}_D^{-1}$ 、 $\mathbf{A}_D+\omega\mathbf{A}_U$ 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。

Axelsson ,O.," Solution of Linear Systems of Equations", Lecture Notes in Mathematics,

572,Springer-Verlag pp.1-51(1977).

７） 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 $\mathrm{Span}\{\mathbf{r}_0,\mathbf{A}\mathbf{r}_0,\cdots\cdots,\mathbf{A}^{n-1}\mathbf{r}_0\}$ を定義し、その中で Hermite 行列に対して、ガレルキン条件 (Galerkin condition) $\mathbf{r}_{n+1}\perp \mathrm{K}_{n+1}(\mathbf{A}:\mathbf{r}_0)$ 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次の形の３項漸化式が導かれる。

$$\mathbf{v}_{n+1}=\alpha_n\left\{\frac{(\mathbf{A}\mathbf{v}_n,\mathbf{v}_n)}{(\mathbf{v}_n,\mathbf{v}_n)}\mathbf{v}_n+\frac{(\mathbf{A}\mathbf{v}_{n-1},\mathbf{v}_n)}{(\mathbf{v}_{n-1},\mathbf{v}_{n-1})}\mathbf{v}_{n-1}-\mathbf{A}\mathbf{v}_n\right\}$$

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 $\mathbf{r}_n\cong \mathrm{R}_n(\mathbf{A})\mathrm{R}_n(\mathbf{A})\mathbf{r}_0$ 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ｽﾞ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。

特長

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の高速性が実現する。

８）Jacobi-Davidson 流

大規模疎行列列の固有値計算は 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 u_k の直交補空間 $\mathrm{u}_k^{\perp}$ を定義する。行列 $\mathbf{A}$ の $\mathrm{u}_k^{\perp}$ への直交射影を

$\mathbf{A}_p = \left(\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*\right)\mathbf{A}\left(\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*\right)$ 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、上式は $\left(\mathbf{A}_p - \lambda \mathbf{I}\right)\mathbf{z} = -\mathbf{r}$ 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近似演算子 $\tilde{\mathbf{M}}_k = \left(\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*\right)\mathbf{M}_k\left(\mathbf{I} - \mathbf{u}_k \mathbf{u}_k^*\right)$ 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 形を正規直交基定から射影行列を定義する事で、固有値を昇順に並ばせる事が容易に出来、複数個の固有値計算を高速で求解する事が出来る。

特長

固有値解法として著名な QR 法に対して、並列性の面から一歩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。

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-ザイデイル法の方が速く収束する事が知られており、このガウス-ザイデイル法より、ガウス-ザイデイ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。

＜　はより速い処理速度を表す。

ガウス法	＜ヤコビ法	＜ガウスザイデイル法	＜ＳＯＲ法	＜チェビシェフ法	＜ＣＧ法	＜前処理付ＣＧ法	＜今後の研究
	〃		＜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 $\mathbf{A}$ 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。