デュアルピボットを行うマルチレベル ILU 分解*

渡邉慶太郎 † 野寺隆 ‡

† 慶應義塾大学大学院理工学研究科 ‡ 慶應義塾大学理工学部

1 はじめに

大規模で非対称な係数行列 A を持つ連立 1 次方程式 $Ax = b$ の近似解法には，クリロフ部分空間に基づく様々な反復解法が存在する．通常，このような反復法の収束を向上させるには，方程式の前処理を利用することになる．不完全 LU 分解（ILU 分解）は反復法の前処理として有効な算法の 1 つである．その中でも，フィルインを 2 つの閾値によってドロッピングを行う ILUT 分解 [1] が，多様なフィルインに適応できる分解として知られている．本発表では，分解過程を分割して行う Crout 版による ILU 分解 (ILUC 分解) [2] に関して，下三角行列 L の列に対するピットと上三角行列 U の行に対する置換の双方，いわゆるデュアルピボットを行う ILDUCDP 分解 [3] を扱うことにする．この手法は，小さな値でのピボットの回避と行列のスパース性の維持をすることによって，反復ソルバの性能の向上させることができる．

2 不完全 LU 分解

不完全 LU 分解は，係数行列 A を上三角行列 U と下三角行列 L に分解する手法である．

$$A = LU \tag{1}$$

この分解を完全に行わない場合を不完全 LU 分解 (Incomplete LU factorization) といい，行列の前処理として広く使われている．

2.1 Crout 版による不完全 LU 分解

ILUC 分解では，係数行列 A から前処理行列を作る際の更新処理が，上三角行列 U を作る際には上から下へ分解を行い，下三角行列 L を作るは左から右へ分解を行う．この ILUC 分解の算法を図 1 に示す．ここでのドロッピングは 2 つの閾値を持ったプロセスで行われる．最初に，行に関して，大きさが閾値 τ 未満のフィルインを棄却，すなわち 0 とする．次に，列の全非ゼロ要素の数が閾値 p 以上であれば，非ゼロ要素の数が閾値になるまで，フィルインの大きさが小さいものから順に棄却を行う．

for $k = 1, \ldots, n$
  $z_{1:k-1} = 0, z_{k:n} = a_{k,k:n}$
  for $i = 1, \ldots, k-1$ and $l_{ki} \neq 0$
    $z_{k:n} = z_{k:n} - l_{ki} * u_{i,k:n}$
  end for i
  Apply a dropping rule to row z
  $w_{1:k} = 0, w_{k+1:n} = a_{k+1:n,k}$
  for $i = 1, \ldots, k-1$ and $u_{ik} \neq 0$
    $w_{k+1:n} = w_{k+1:n} - u_{ik} * l_{k+1:n,i}$
  end for i
  Apply a dropping rule to column w
  $u_{k,:} = z$
  $l_{:,k} = w/u_{kk}, l_{kk} = 1$
end for k

図 1: ILUC 分解の算法

2.2 デュアルピボットを用いた不完全 LU 分解

ILUC 分解では対角要素の大きさが小さい場合があり，対角優位性を損ねる問題が生じる．このときに列に対するピボット，つまり列の中で最大の要素を選択し置き換えることにより，対角優位性を保つ [4]．ピボットはパラメータ c によって選択的に行い，列中の要素で最大の大きさを P_{max}，対角要素の大きさを P_D と置き，$P_{max} > c \cdot P_D$ を満たすときのみピボットを実行する．さらに，分解後の上三角行列 U に対して，k 番目のステップにおいて，k 番目までの行に対して非ゼロ要素 $M(k)$ が 1 番少ない行に対して消去を行う．この行の入れ替えの操作は，行列を疎に保ちフィルインを少なくすることを目的とする．これら 2 つの行列に対する操作，すなわち列に対するピボットと行の入れ替えをデュアルピボッティングという．また，利便性のために，係数行列 A を下三角行列 L と上三角行列 U に分解する LU 分解ではなく，対角行列 D を含

*A Multilevel Imcomplete LU factorization with Dual Pivoting

Keitaro WATANABE† and Takashi NODERA‡
†Graduate School of Science and Technology , Keio University
‡Faculty of Science and Technology, Keio University
ketaro@z5.keio.jp

for $k = 1, \ldots, n$
  $\hat{k} = permrows(k)$
  $ur(\hat{k}) = false$
  $z_{\overline{np}} = 0, z_{np} = a_{\hat{k},np}$
  for $i = 1, \ldots, k-1$ and $l_{ki} \neq 0$
    $z_{np} = z_{np} - l_{\hat{k}i} * d_{ii} * u_{i,np}$
  end for i
  choose z_{k^*}
  $d_{kk} = z_{k^*}, z_{np} = z_{np}/d_{kk}$
  Update $perm$
  $np_{k^*} = false$
  $w_{\overline{ur}} = 0, w_{ur} = a_{ur,perm(k)}$
  for $i = 1, \ldots, k-1$ and $u_{i,perm(k)} \neq 0$
    $w_{ur} = w_{ur} - l_{ur,i} * d_{ii} * u_{i,perm(k)}$
  end for i
  Apply a dropping rule to row z_{np}
  Apply a dropping rule to column w_{ur}
  $l_{ur,k} = w_{ur}/u_{k,perm(k)}, l_{\hat{k},k} = 1$
  $u_{k,np} = z_{np}, u_{k,perm(k)} = 1$
 Update the permuting rows and $permrows$
end for k

図 2: ILUCDP 分解の算法

めた LDU 分解を使用する．このデュアルピボッティングを ILUC 分解に組み込んだ ILDUCDP 分解の算法を図 2 に示す．ただし，$permrows$ は次に消去を行う行，$perm$ は次にピボットを行う列を示し，np は列がピボットを未実行かを示すブーリアン変数，ur は行が未消去かを示すブーリアン変数である．

2.3 マルチレベル化

係数行列 $A = A_1$ が以下のブロック行列で分割できるとする．

$$A_1 = A = \begin{pmatrix} B & F \\ E & C \end{pmatrix}$$

ここで，$B = L_B D_B U_B$ となる LDU 分解を行う．

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} L_B & 0 \\ E_B & I \end{pmatrix} \begin{pmatrix} D_B & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} U_B & F_B \\ 0 & I \end{pmatrix}$$

ただし，$B = L_B D_B U_B$, $E_B = E {U_B}^{-1} {D_B}^{-1}$, $F_B = {D_B}^{-1} {L_B}^{-1} F$ である．ここで，$S = C - E_B D_B F_B$ はシュールコンプリメントである．そして，S を次のレベルで用いる係数行列 $A_2 = S$ と置く．この分解を幾度か繰り返し，シュールコンプリメントの分解を終了する．分解の終了条件は，係数行列 A_l が閾値 β を用いて，

$$M(k) > \beta \frac{nnz(A_l)}{n} \tag{2}$$

を満たした時である．ここで，$nnz(A)$ は係数行列 A の非ゼロ要素を意味する．

2.4 プリプロセス

レベル l においての行列 A_l を処理することによって，より効果的な前処理をすることができる．代表的な例として，行と列に対するスケーリング，PQ リオーダリング，MND リオーダリング，最大重み付けマッチングオーダリングなどが存在する．これらのプリプロセスを組み合わせることによって，さらに効果的な前処理をすることが可能になる．

3 まとめ

本稿では，列に対するピボットと行に対する置換を併せたデュアルピボットを用いた不完全 LU 分解，ILDUCDP 分解について述べた．この手法は ILUC 分解などに比べて，反復法に対してよい前処理行列を作ることができると考えられる．この手法の詳細，数値実験の結果と他手法との比較，考察は当日発表する．

参考文献

[1] Saad, Y.: ILUT: a dual threshold incomplete LU factorization, *Numer. Linear. Algebra. Appl.*, Vol. 1, No. 4, pp. 387–402 (1994).

[2] Li, N., Saad, Y., and Chow, E.: Crout versions of ILU preconditioner for general sparse matrices, *SIAM J. Sci. Comput.*, Vol. 25, No. 2, pp. 715–728 (2003).

[3] Mayer, J.: A multilevel preconditioner with pivoting and row permutation, *Numer. Linear. Algebra. Appl.*, Vol. 14, No. 10, pp. 771–789 (2007).

[4] Mayer, J.: ILUCP: a Crout ILU preconditioner with pivoting, *Numer. Linear. Algebra. Appl.*, Vol. 12, No. 9, pp. 941–955 (2005).

[5] 藤原，藤野，古田：フィルインに対する対角要素の補填による Crout 版 ILU 前処理の改良，日本計算工学会論文集，論文番号 20060022 (2006).