

波回折問題のための境界要素法への ILUC の適用

Application of the Crout version of the incomplete LU factorization (ILUC) to the Boundary Element Analysis for Wave Diffraction Problems

巻幡憲俊*・宇都宮智昭**・渡邊英一***

Noritoshi MAKIHATA, Tomoaki UTSUNOMIYA and Eiichi WATANABE

*学生員 京都大学大学院 工学研究科社会基盤工学専攻 (〒 606-8501 京都市左京区吉田本町)

**正会員 工博 京都大学助教授 大学院工学研究科社会基盤工学専攻 (〒 606-8501 京都市左京区吉田本町)

***フェロー Ph.D., 工博 京都大学教授 大学院工学研究科社会基盤工学専攻 (〒 606-8501 京都市左京区吉田本町)

Incomplete LU factorizations (ILU), combined with the generalized minimal residual method (GMRES), are often regarded as the best 'general purpose' iterative solvers. The dual threshold incomplete LU factorization (ILUT) is a well-known ILU preconditioner. It is effective for general sparse linear systems. However, for dense matrices, the cost for the construction of this preconditioner is high. We thus consider the Crout version of the ILU preconditioner (ILUC) in this study. It is known that ILUC can compute much faster than ILUT for sparse matrices. We apply the ILUC preconditioner for the analysis of the boundary value problem related to the diffraction wave field around a Very Large Floating Structure (VLFS) and show that this preconditioner is, in some cases, effective for dense linear systems.

Key Words : preconditioning, incomplete LU factorization, ILUT, ILUC, non-Hermitian systems, dense linear systems, VLFS, boundary element method, fast multi-pole method

1. はじめに

超大型浮体まわりの回折波浪場に関する境界値問題を境界要素法 (BEM) により離散化すると、非 Hermite かつ密行列を係数とする大規模連立一次方程式 $Ax = b$ が現れる¹⁾。このような連立一次方程式の反復解法として、the generalized minimal residual method (GMRES)²⁾ がよく知られている。GMRES は収束に到るまでの反復回数の増加とともに、必要とする記憶容量も増加する。記憶容量が制限された環境においては、このような問題に対する連立一次方程式の解法として the implicitly restarted GMRES (GMRES-IR)³⁾ や the GMRES with deflated restarting (GMRES-DR)⁴⁾ が有効であることを実証した^{5), 6)}。

しかし記憶容量が十分にある環境においては、GMRES を適用することができる。また前処理を用いることにより、より高速にかつ安定に連立一次方程式を解くことが可能になる場合がある。前処理としてよく知られているものに the dual threshold incomplete LU factorization (ILUT)⁷⁾ がある。現在では、ILUT よりも収束性をより高めることのできる前処理行列を作成する the Crout version of the incomplete LU factorization (ILUC)⁸⁾ が開発されている。

本研究では、超大型浮体まわりの回折波浪場に関する境界値問題の解析に ILUC を適用し、そのベンチマーク計算の結果から本手法の有効性について検討する。

2. 前処理について

反復解法に前処理を施すことにより、より効率的に近似解を求められることがある。ここでは、前処理として不完全 LU 分解 (ILU)^{7), 8)} を考え、この方法が直接法と反復法の間隔的な手法であるという観点で記述する。

2.1 反復改良

連立一次方程式を消去法 (LU 分解)⁹⁾ により数値的に解くことを考える。通常の LU 分解を実行したとき、その計算過程で非常に小さい数や非常に大きい数が現れることがある。また、計算過程で現れる数が厳密計算では 0 であるが丸め誤差の影響をうけて非常に小さい数が現れたりすることもある。そのような数に対して演算を実行すると解の精度に悪い影響を与える。解を安定に求める方法として、計算過程で現れる数が倍精度計算において例えば 10^{-12} より小さいとき、それを 0 とみなして LU 分解を実行することにより近似解を求め、その後その近似解に対して GMRES などの反復法を用いて解を求める。これをここでは反復改良と呼ぶことにする^{9), 10)}。計算時間においても、通常の LU 分解より速く解が求まることが多い。

2.2 ILU

連立一次方程式 $Ax = b$ を GMRES などの反復法により数値的に解くことを考える。反復回数が非常に多くなる問題もあり、この場合制限された時間内に解が求められないことがある。また、その残差が適当な停止基準を時間内に満たすことができたとしても、計算量が多いため丸め誤差の影響を受けて求めたい精度の

解が得られないときもある。そこで、反復回数を大幅に減らす可能性がある方法の一つとして、前処理がある。ここでは前処理として ILU を考える。

計算過程で現れる数が適当に与えた正の数 τ より小さいとき、それを 0 とみなして LU 分解を実行し、 $M = LU$ とおく。 M は前処理行列と呼ばれている。左前処理と呼ばれる、

$$M^{-1}Ax = M^{-1}b \quad (1)$$

または右前処理と呼ばれる、

$$AM^{-1}y = b; x = M^{-1}y \quad (2)$$

として、 $M^{-1}A$ または AM^{-1} に対して GMRES を実行することにより解を求める。 τ を小さくしていけば、 $M^{-1}A$ または AM^{-1} のすべての固有値は 1 の近くに集まり、次の不等式¹¹⁾から反復回数が減少すると期待される。

$$\frac{\|r_m\|_2}{\|r_0\|_2} \leq \frac{L}{2\pi\epsilon} \inf_{p_m \in P_m, p_m(0)=1} \|p_m\|_{\Lambda_\epsilon} \quad (3)$$

ただし、 ϵ は適当に与えた正数であり、 Λ_ϵ は $M^{-1}A$ または AM^{-1} の ϵ -pseudospectrum¹¹⁾ 全体、 L は Λ_ϵ の境界の長さ、 P_m は m 次多項式全体、 r_m は GMRES による近似解 x_m の残差である。

しかし、 M の作成時間や、その保存のためのメモリは増加する。したがって、前処理行列の作成時間があまりかからないように、同時に反復回数が減少するように τ を適切に選ぶ必要がある。また、適切な τ が存在しない場合もある。

ここで、 τ が 10^{-12} のときは前処理と反復改良は同じものとなる。

前処理は、例えば有限要素法 (FEM)¹²⁾ などにより得られた疎行列を係数とする連立一次方程式に対する適用がほとんどであり、この場合 τ を小さく選んでも前処理行列の作成時間はほとんどかからないことが多い。

2.3 密行列への適用

連立一次方程式の係数行列が密である場合、前処理の適用は困難となる。

ここでは ILUC を密行列に対して適用し、その有効性を検討する。

3. ILUT とその問題点

前処理において一般的な手法として、the dual threshold incomplete LU factorization (ILUT)⁷⁾ がよく知られている。ここでは ILUT のアルゴリズムを記述することにより、ILUT の問題点を明らかにする。

3.1 LU 分解

LU 分解の実行には数種類の変形がある。break down を避けるために、また数値的に安定な解を求めるために、LU 分解において枢軸選択⁹⁾が必要であるが、ILU において枢軸選択を行うと前処理行列作成に必要な計

算時間が増加するため効果が現れない時がある。ここでは枢軸選択を考えないことにする。

係数行列が狭義優対角行列や正定値行列のときは枢軸選択を行わなくても、break down は起こらない。

通常の LU 分解は、次の KIJ version である。この version の ILU への適用に関する利点は、 L の成分と U の成分を対称に生成することである。これは、 L と U の 0 でない成分を少なくしても、すなわち τ が大きくても、 $M = LU$ は A をよく近似することにつながる。(このことは、あとでもう少し詳しく記述する。) しかし、 S の保存に多くのメモリを必要とするため ILU への適用は困難である。

ここで、 $\hat{d}_k$ は $\hat{U}$ の k 番目の対角成分である。

KIJ version

```

S = A
do k = 1, ..., n
   $\hat{d}_k = s_{kk}$ 
  do i = k + 1, ..., n
     $\hat{u}_{ki} = s_{ki}; \hat{l}_{ik} = s_{ik}/\hat{d}_k$ 
    do j = k + 1, ..., n
       $s_{ij} = s_{ij} - \hat{l}_{ik}s_{kj}$ 
    end do
  end do
end do

```

KIJ version において、K-loop と I-loop を入れ替えると、次の IKJ version が得られる。この version から ILUT が導かれる。

IKJ version

```

do i = 1, ..., n
  do k = 1, ..., i - 1
     $\hat{l}_{ik} = a_{ik}/\hat{d}_k$ 
    do j = k + 1, ..., n
       $a_{ij} = a_{ij} - \hat{l}_{ik}\hat{u}_{kj}$ 
    end do
  end do
   $\hat{d}_i = a_{ii}$ 
  do j = i + 1, ..., n
     $\hat{u}_{ij} = a_{ij}$ 
  end do
end do

```

また、loop を適当に入れ替えることによって、JKI version も得られる。この version から ILUT が得られる。

3.2 ILUT のアルゴリズム

LU 分解の計算過程において、適当に与えられた正の数 τ より小さい数を捨てることにより、ILUT が得られる。実行の容易さから、IKJ version または JKI version が選ばれる。疎行列 L および U は IKJ version のときは共に圧縮行格納法 (CRS)¹³⁾ により、JKI version のときは共に圧縮列格納法 (CCS)¹³⁾ により格納される。

ここで、 d_k は U の k 番目の対角成分である。

```

ILUT (IKJ version)
do i = 1, ..., n
  do j = 1, ..., n
     $w_j = a_{ij}$ 
  end do
  do k = 1, ..., i - 1
     $w_k = w_k / d_k$ 
    if ( $|w_k| \geq \tau$ ) then
       $l_{ik} = w_k$ 
    end if
    do j = k + 1, ..., n;  $u_{kj} \neq 0$ 
       $w_j = w_j - w_k u_{kj}$ 
    end do
  end do
   $d_i = w_i$ 
  do j = i + 1, ..., n
    if ( $|w_j| \geq \tau$ ) then
       $u_{ij} = w_j$ 
    end if
  end do
end do

```

3.3 ILUTの問題点

ILUTのIKJ versionは上の行から下の行へと各行ごとに L の成分と U の成分が生成される。したがって、上の方の行から生成された成分の近似はよいが、下の方の行から生成された成分の近似はあまりよくない。すなわち、ILUTにより得られた疎行列 U はLU分解により得られる上三角行列 $\hat{U}$ をよく近似するが、 L は $\hat{L}$ をあまり近似しないことを意味する。このため、 L の成分による影響が、 U のそれと比較して、あまり出ないように、 L の成分の生成に関する“if ($|w_j| \geq \tau$) then”の w_j において、 d_j で割る前の w_j に対してではなく d_j で割った後の w_j に対して、ifにより判断するよう工夫されている。 L が $\hat{L}$ を近似するためには、 τ を十分小さく選ぶ必要がある。このため L 、 U の0でない成分の数は共に大きくなり、係数行列 A が密のときは、前処理行列の作成に時間がかかる。同様にILUTのJKI versionは L は $\hat{L}$ を近似するが、 U は $\hat{U}$ をあまり近似しないことが起こる。

4. ILUCとその実行

LU分解におけるIKJ versionとJKI versionを組み合わせることで、Crout versionが得られる。このversionをILUへ適用したものは、the Crout version of the incomplete LU factorization (ILUC)⁸⁾と呼ばれている。ILUCは、 U は $\hat{U}$ を L は $\hat{L}$ を、ILUTと比較して共によく近似するため、疎行列に対してはもちろんのこと、密行列に対しても効果的となる可能性がある。

4.1 ILUCのアルゴリズム

ILUCのアルゴリズムは以下のとおりである。Crout versionによる L の成分と U の成分の生成順序は、KIJ versionと同じであるので、ILUTのように L 部分また

は U 部分のどちらかに偏ることなく、 L は $\hat{L}$ を U は $\hat{U}$ を共に近似する。ここではbreak downを避けるために、 U の対角成分 d_k がマシンイプシロン ϵ_{eps} より小さいときは 10^{-3} で置き換えることにする。

```

ILUC
do k = 1, ..., n
  do j = k, ..., n
     $z_j = a_{kj}$ 
  end do
  do i = 1, ..., k - 1;  $l_{ki} \neq 0$ 
    do j = k, ..., n;  $u_{ij} \neq 0$ 
       $z_j = z_j - l_{ki} u_{ij}$ 
    end do
  end do
  do j = k + 1, ..., n
     $w_j = a_{jk}$ 
  end do
  do i = 1, ..., k - 1;  $u_{ik} \neq 0$ 
    do j = k + 1, ..., n;  $l_{ji} \neq 0$ 
       $w_j = w_j - l_{ji} u_{ik}$ 
    end do
  end do
  if ( $|z_k| < \epsilon_{\text{eps}}$ ) then
     $z_k = 10^{-3}$ 
  end if
   $d_k = z_k$ 
  do j = k + 1, ..., n
    if ( $|z_j| \geq \tau$ ) then
       $u_{kj} = z_j$ 
    end if
  end do
  do j = k + 1, ..., n
    if ( $|w_j| \geq \tau$ ) then
       $l_{jk} = w_j / d_k$ 
    end if
  end do
end do

```

4.2 L と U の格納について

U は疎行列でありまたその生成順序から、 U の行中の0でない要素を連続したメモリ上に配置するCRS¹³⁾により格納する。 U の0でない成分の数をnzuとすると、 U の0でない成分を格納する複素数配列U(nzu)、その列番号を格納する整数配列JU(nzu)および各行の開始位置を格納する整数配列IU(n+1)を用いると、k stepにおいて次のようになる。ただし、IU(1)=1、IU(n+1)=nzu+1とする。

```

q = IU(k)
do j = k + 1, ..., n
  if ( $|z_j| \geq \text{tol}$ ) then
    U(q) =  $z_j$ 
    JU(q) = j
    q = q + 1
  end if
end do

```

$$IU(k+1) = q$$

同様に、 L は疎行列でありまたその生成順序から、 L の列中の 0 でない要素を連続したメモリ上に配置する CCS¹³⁾により格納する。 L の 0 でない成分の数を nzl とすると、複素数配列 $L(nzl)$ 、整数配列 $IL(nzl)$ および整数配列 $JL(n+1)$ が用いられる。

4.3 ILUC の詳細

疎行列 L 、 U の格納方法から次の 2 点が問題になる。

1. "do $j = k, \dots, n; u_{ij} \neq 0$ "
2. "do $i = 1, \dots, k-1; l_{ki} \neq 0$ "

1 に関して、 k step において、整数 $Ufirst(i)$ ($i = 1, \dots, k-1$) に、 i 行の k 番目以降である $u_{ik}, \dots, u_{in}$ の配列 $U(*)$ における開始位置を格納する。 $Ufirst$ は各 step ごとに更新される。

2 に関して、linked list⁸⁾により、 $l_{k1}, \dots, l_{k,k-1}$ の列番号を指定する。すなわち、 $Llist(k)$ には、 $l_{k1}, \dots, l_{k,k-1}$ において 0 でないもののうち列番号が最も小さいものが格納され、 $Llist(Llist(k))$ にはその次に小さい列番号が格納される。

詳しくは、 k step において次のようになる。ここで、 $Lend$ は $l_{k1}, \dots, l_{k,k-1}$ のうち 0 でないものの個数である。

```

m = k
Lend = 0
do i = 1, ..., k-1
  p = Lfirst(i)
  if ((p ≤ JL(i+1) - 1) ∧ (IL(p) = k)) then
    Llist(m) = i
    m = i
    Lend = Lend + 1
  end if
end do
i = k
do m = 1, ..., Lend
  i = Llist(i)
  p = Lfirst(i)
  do q = Ufirst(i), ..., IU(i+1) - 1
    j = JU(q)
    z_j = z_j - L(p) * U(q)
  end do
  Lfirst(i) = p + 1
end do
Lfirst(k) = JL(k)

```

5. ベンチマーク計算結果

ここでは超大型浮体まわりの回折波浪場に関する境界値問題の解析^{1), 14), 15)}に ILUC-GMRES を適用し、そのベンチマーク計算結果を載せる。

解析モデルは、一定水深の無限海域に設置される箱形ポンツーン浮体である。また、その主要諸元を表-1

表-1 主要諸元

VLFS の型	pontoon
長さ (m)	2000
幅 (m)	400
喫水 (m)	1.5
水深 (m)	20.0

に表す。平均自由水面上に xy 平面をとり、 z 軸は鉛直上向きが正の向きとなるように定義する。これに、波振幅 A 、角周波数 ω の微小振幅波が入射するものとする。入射波方向 β を、 $\beta = 0$ の時に x 軸の正の方向から負の方向へ進行する波となり、 $\beta = \pi/2$ の時に y 軸の正の方向から負の方向へ進行する波となるような方向として定義する。

流体を非粘性、非圧縮性の完全流体とし、流体運動を非回転的とするとき、全ての流体運動は速度ポテンシャル Φ で表現することができる。また、流体運動、浮体運動いずれも定常的な角周波数 ω の調和振動を行っているものとする。このとき、全ての時間依存項について、以下のように表す。

$$\Phi(x, y, z, t) = \text{Re}[\phi(x, y, z)e^{i\omega t}] \quad (4)$$

ただし、 i は虚数単位を、 t は時間を表す。

流体運動は微小であるとし、微小振幅波理論が適用できるものとする。この前提のもとで、本研究で扱う問題を、速度ポテンシャル ϕ に関する境界値問題として定式化する。

$$\nabla^2 \phi = 0 \quad \text{in } \Omega \quad (5)$$

$$\frac{\partial \phi}{\partial z} = K\phi \quad \text{on } S_F \quad (6)$$

$$\frac{\partial \phi}{\partial z} = 0 \quad \text{on } B_0 \quad (7)$$

$$\frac{\partial \phi}{\partial n} = 0 \quad \text{on } S_H \quad (8)$$

$$\lim_{r \rightarrow \infty} \sqrt{r} \left(\frac{\partial(\phi - \phi_I)}{\partial r} - ik(\phi - \phi_I) \right) = 0 \quad \text{on } S_\infty \quad (9)$$

$$\phi_I = i \frac{gA}{\omega} \frac{\cosh k(z+h)}{\cosh kh} e^{ik(x \cos \beta + y \sin \beta)} \quad (10)$$

ここで、 ϕ_I は入射波ポテンシャルである。また、 Ω は流体領域を、 S_F は自由水面を、 B_0 はベースとなる $z = -h$ の一定水深面を、 S_H は浮体の没水表面を、 S_∞ は無限遠境界を表す。 K は無限水深時波数 ($= \omega^2/g$; g は重力加速度) を、 k は波数を表し、次式の分散関係式を満たす。

$$k \tanh kh = K \quad (11)$$

また、 n は単位法線ベクトル (流体から外向き方向を正とする) を表す。 r は原点からの水平距離である。

速度ポテンシャル ϕ に関する境界条件式を積分方程式に代入すると次式が得られる。

$$\begin{aligned} 4\pi\phi(x) + \int_{S_H} \left\{ \phi(\xi) \frac{\partial G(x, \xi)}{\partial n} - \phi(x) \frac{\partial G_2(x, \xi)}{\partial n} \right\} dS \\ = 4\pi\phi_I(x) \end{aligned} \quad (12)$$

ここで、積分および法線微分は ξ に関して計算し、 $G(x, \xi)$ は水波 Green 関数である。要素としては 8 節点 2 次要素および 4 節点 1 次要素を用いる。上式は、 S_H の表面上に配置される N 個の節点における未知ポテンシャル $\phi_i (i = 1, \dots, N)$ を未知数とする連立一次方程式に帰着される^{14),15)}。要素分割はポンツーン型浮体の没水表面に関して行う。8 節点 2 次要素の場合は、底面を $10\text{m} \times 10\text{m}$ のパネルで、側面を $10\text{m} \times 1.5\text{m}$ のパネルで分割し、要素数 8480、節点数 25921 である。4 節点 1 次要素の場合は、底面を $5\text{m} \times 5\text{m}$ のパネルで、側面を $5\text{m} \times 1.5\text{m}$ のパネルで分割し、要素数 32960、節点数 33441 とした場合 (分割 A) と底面を $4\text{m} \times 4\text{m}$ のパネルで、側面を $4\text{m} \times 1.5\text{m}$ のパネルで分割し、要素数 51200、節点数 51801 とした場合 (分割 B) の 2 種類とした。

それぞれの解析モデルにおいて、入射角を x 軸から 60° 、入射波長 λ を、8 節点 2 次要素の場合は 197.492m 、 152.328m 、 105.187m または 55.034m として、4 節点 1 次要素の場合は 197.492m (分割 A のみ) または 55.034m (分割 A 及び B) としてベンチマーク解析を行う。

5.1 8 節点 2 次要素の場合

ここでは高速多重極展開法 (FMM)¹⁴⁾ を適用せず、係数行列 A を保存して、ILUC による前処理行列を作成する方法をとる。

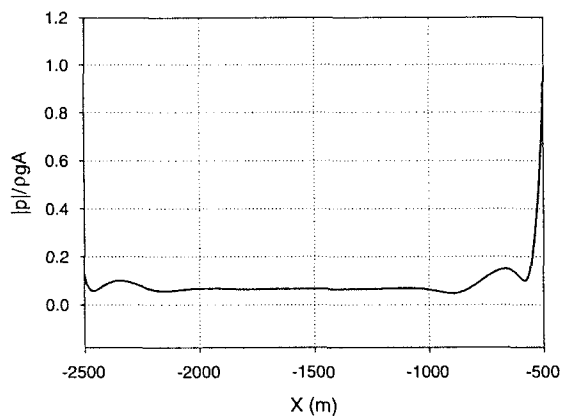


図-1 ポンツーン 入射波長=197.492m

図-1, 2, 3, 4 は、浮体底面 ($z = -1.5$) の x 軸上における圧力分布 $p(x, 0, -1.5)$ を図にしたものである。また、前処理を適用した場合と適用しなかった場合とで圧力分布がほぼ完全に一致することを確認している。

表-2, 3, 4, 5 はそれぞれ、入射波長 λ を 197.492m 、 152.328m 、 105.187m または 55.034m として ILUC-GMRES と ILUT-GMRES を比較して表にしたものである。反復はすべて初期値 $x_0 = 0$ から始め、京都大学学術情報メディアセンターのスーパーコンピュータ HPC2500 において倍精度でスレッド並列計算 (32CPU) をした。ここで、 τ は適当に選んだ正の数であり、LU 分解の計算過程においてその値より小さい数は 0 とみなされる。 τ が ∞ の時は、前処理を適用しない通常の GMRES に対応している。“Ratio” は ILUC または

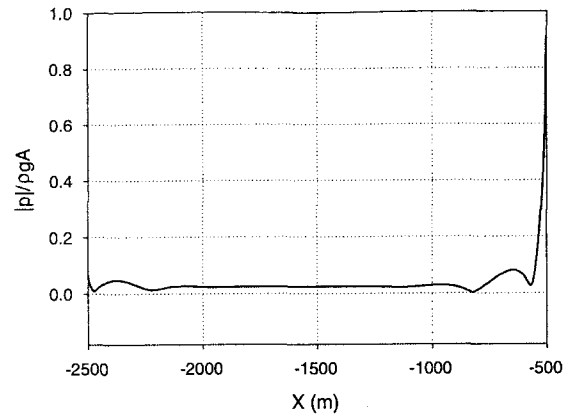


図-2 ポンツーン 入射波長=152.328m

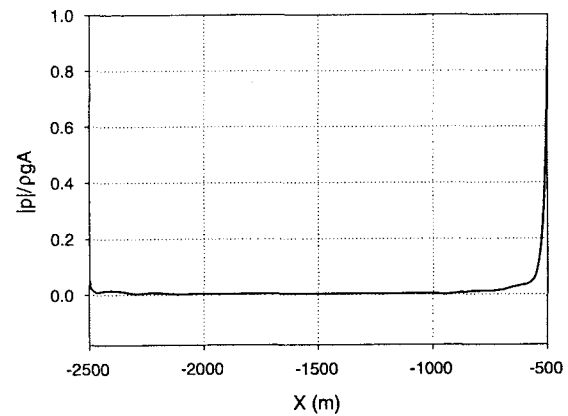


図-3 ポンツーン 入射波長=105.187m

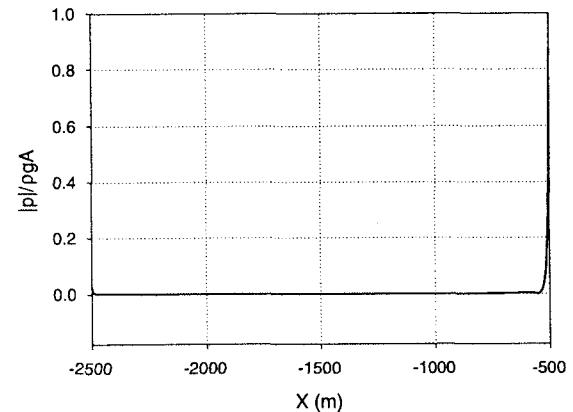


図-4 ポンツーン 入射波長=55.034m

ILUT により得られた L と U の 0 でない成分の個数を係数行列 A の成分の個数 ($= N^2$) で割った比率を、“Its” は反復回数を、“PreSec” は前処理行列の作成時間を、“ItSec” は反復に要した時間を、“TotSec” は全計算時間を表す。“—” は反復回数が 1000 以内で収束が達成されなかったことを意味する。また、収束回数 $\|r\|_2 \leq 10^{-5} \|b\|_2$ が満たされる最小の数とした。

表-2 から、 τ が 0.06 のとき ILUC-GMRES は通常の GMRES より速く解に収束していることが分かる。また、 τ を 0.1 より大きく選ぶと ILUC は収束回数の低減に寄与していないことも分かる。 τ が 0.1 以下の時

表-2 τ を変化させたときの ILUC と ILUT の計算結果. $\lambda = 197.492\text{m}$, $N = 25921$.

τ	ILUC					ILUT				
	Ratio	Its	PreSec	ItSec	TotSec	Ratio	Its	PreSec	ItSec	TotSec
∞	0.0	140	0	1400	1400	0.0	140	0	1400	1400
1.0	0.00009	143	100	1519	1619	0.00007	137	147	1230	1377
0.1	0.008	84	214	1263	1477	0.0001	172	692	1461	2153
0.08	0.019	58	289	879	1168					
0.06	0.061	35	462	344	806	0.0089	—	2097	—	—
0.04	0.187	19	3502	244	3746	0.0089	—	2239	—	—
0.02						0.145	53	18951	550	19501
0.01						0.268	29	31061	300	31361

表-3 τ を変化させたときの ILUC と ILUT の計算結果. $\lambda = 152.328\text{m}$, $N = 25921$.

τ	ILUC					ILUT				
	Ratio	Its	PreSec	ItSec	TotSec	Ratio	Its	PreSec	ItSec	TotSec
∞	0.0	205	0	2386	2386	0.0	205	0	2386	2386
1.0	0.00019	180	99	1911	2010	0.00011	192	152	1593	1745
0.1	0.050	63	341	553	894	0.0051	225	1730	2019	3749
0.08	0.106	29	1207	346	1553					
0.06						0.0043	—	3091	—	—
0.04						0.0114	—	3976	—	—
0.02						0.207	49	29291	587	29878
0.01						0.354	215	37311	2535	39846

表-4 τ を変化させたときの ILUC と ILUT の計算結果. $\lambda = 105.187\text{m}$, $N = 25921$.

τ	ILUC					ILUT				
	Ratio	Its	PreSec	ItSec	TotSec	Ratio	Its	PreSec	ItSec	TotSec
∞	0.0	331	0	3242	3242	0.0	331	0	3242	3242
1.0	0.00022	282	83	2332	2415	0.00012	302	171	2921	3092
0.6	0.00058	270	87	2259	2346					
0.4	0.0027	169	105	1415	1520					
0.2	0.047	96	312	835	1147					
0.1	0.239	48	4291	506	4797	0.079	—	16252	—	—
0.06						0.00015	—	962	—	—
0.04						0.199	102	25413	1000	26413
0.02						0.331	—	34825	—	—
0.01						0.634	—	42054	—	—

は、 τ を小さくしていくと収束回数もほぼ単調に減少している。このとき、必要とする記憶容量がわずかであり、前処理行列の作成にあまり時間がかからないため、ILUC は有効となる。

同様に、表-3, 4 においても、ILUC が有効であることは明らかである。

一方、表-5 においては、 τ を 0.2 より大きく選ぶと ILUC は収束回数の低減に寄与していないことが分かる。実際、 τ が 0.4 のときは反復回数が 1000 を超えても収束しなかった。また τ が 0.8 のとき、通常の GMRES より速く解に収束しているが、この値から τ を減少させたとき、収束回数が単調に減少していない。 τ を 0.2 より小さくすると収束回数の大幅な低減が認められるが、前処理行列の作成に必要な時間が非常にかかるため、この場合には ILUC は有効でないと思われる。

5.2 4 節点 1 次要素 (分割 A) の場合

ここでは高速多重極展開法 (FMM)¹⁴⁾ を適用し、係数行列 A の成分を保存しないで、ILUC による前処理行列を作成する方法をとる。このとき、各 k step において A の k 行と k 列の一部を計算しながら前処理行列を作成するためその生成に少し時間がかかる。なお、FMM と従来法との比較については、文献^{14), 15)} で述べられており、本論文ではその比較は省略する。

反復はすべて初期値 $x_0 = 0$ から始め、HPC2500 において、倍精度でスレッド並列計算 (8CPU) をした。なお、FMM の並列化においては西村、宗像¹⁶⁾ を参照し、行列-ベクトル積の計算における上向き pass 及び下向き pass に対して適用した。

表-6 から、係数行列を保存しないアルゴリズムのため、前処理行列の作成に時間がかかっているが、 τ の

表-5 τ を変化させたときの ILUC と ILUT の計算結果. $\lambda = 55.034m, N = 25921$.

τ	ILUC					ILUT				
	Ratio	Its	PreSec	ItSec	TotSec	Ratio	Its	PreSec	ItSec	TotSec
∞	0.0	649	0	6783	6783	0.0	649	0	6783	6783
1.0	0.007	765	144	8094	8238	0.007	—	1185	—	—
0.8	0.025	595	182	5282	5464					
0.6	0.161	—	2690	—	—					
0.4	0.386	—	7870	—	—	0.092	—	12036	—	—
0.2	0.437	298	10519	3397	13916	0.005	—	5547	—	—
0.1	0.519	33	27861	580	28441	0.226	470	36296	6660	42956

表-6 ILUC の計算結果. $\lambda = 197.492m, N = 33441$

τ	Ratio	Its	TotSec
∞	0.0	140	2988
1.0	0.00003	138	5320
0.1	0.00067	123	5009
0.04	0.0102	93	4436
0.02	0.1716	28	8959

表-8 ILUC の計算結果. $\lambda = 55.034m, N = 51801$

τ	Ratio	Its	TotSec
∞	0.0	662	17219
1.0	0.0001	573	21794
0.8	0.0001	558	21429
0.6	0.0003	496	19916
0.4	0.0010	335	15787
0.3	0.0039	182	11908
0.2	0.0299	297	16281

表-7 ILUC の計算結果. $\lambda = 55.034m, N = 33441$

τ	Ratio	Its	TotSec
∞	0.0	664	15131
1.0	0.0002	544	15334
0.8	0.0004	475	13705
0.6	0.0012	311	9951
0.4	0.0140	577	16471
0.2	0.1239	309	13867

減少とともに収束回数もほぼ単調に減少していることが分かる。必要とする記憶容量もわずかである。

表-7 より、8 節点 2 次要素のときとは違って、 τ の値が大きくても、収束回数が急激に増加したりしないことが分かる。また τ が 0.6 のとき、ILUC-GMRES は通常の GMRES より速く解に収束しており、高速多重極展開法の適用で前処理行列の作成に時間がかかって、ILUC が有効となっている。

5.3 4 節点 1 次要素 (分割 B) の場合

ここでも高速多重極展開法 (FMM)¹⁴⁾ を適用し、係数行列 A の成分を保存しないで ILUC による前処理行列を作成する方法をとり、HPC2500 において倍精度でスレッド並列計算 (8CPU) をした。

表-8 より、分割をより細かくした場合は、 τ の減少とともに収束回数もほぼ単調に減少しており、特に τ が 0.3 の時には通常の GMRES に対して 27% まで減少した。ただし、 τ が 0.2 の時は逆に収束回数が増えており、表-5、7 の挙動と共にその原因の解明は今後の課題である。

なお、本論文では収束性の面を主に問題にしてきたが、ILUC はそのアルゴリズムから並列計算に向いていないことが知られている。現在、並列計算に向けた前処理として A の逆行列の近似を目的とする the factored

approximate inverse (AINV)¹⁷⁾ などが提案されているが、前処理行列の作成時間やその格納方法に問題があり、現時点では ILUC の方が実用的であると思われる。また、AINV と ILU は強く関係していることも知られている¹⁸⁾。

5.4 τ による制御と p による制御について

不完全分解による前処理は、 τ によって制御されることが多い。これは不完全分解による誤差評価式を比較的容易に導くことができるためであると思われる。しかし、最適となる τ の値は問題に依存するため、その設定は困難である。

LU 分解の計算過程の各 step において現れる成分のうち絶対値の大きい成分から、適当に定めた数 p だけ格納する方法⁷⁾ も知られている。ここで扱った問題に関しては、 p により制御を行い、 $p = 0.05 \times N$ と設定した結果と、 τ により制御を行い、Ratio が 0.05 となるように設定した結果に大きな違いがないことを確認している。

6. おわりに

ここでは、本論文の内容を箇条書きで簡単にまとめる。

1. LU 分解から始めて、不完全 LU 分解が得られるまでを詳しく論じた。
2. 前処理の適用が困難である密行列に対して、ILUC の適用を試みた。
3. 超大型浮体まわりの回折波浪場に関する境界値問題の解析に ILUC を適用したベンチマーク計算を行い、その有効性を確認した。

4. 8 節点 2 次要素のときは、短波長において ILUC の適用が困難な場合があり、その状況を記述した。
5. 4 節点 1 次要素のときは、ILUC の適用が有効であることを示した。

参考文献

- 1) T. Utsunomiya, and E. Watanabe: Wave response analysis of hybrid-type VLFS by accelerated BEM, *Proc. 3rd Int. Conf. Hydroelasticity in Marine Technology*, pp.297-303, 2003.
- 2) Y. Saad, and M. H. Schultz: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, 7, pp.856-869, 1986.
- 3) R. B. Morgan: Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, *SIAM J. Matrix Anal. Appl.*, 21, pp.1112-1135, 2000.
- 4) R. B. Morgan: GMRES with deflated restarting, *SIAM J. Sci. Comput.*, 24, pp.20-37, 2002.
- 5) 巻幡憲俊, 宇都宮智昭, 渡邊英一: 波浪回折問題のための境界要素法への GMRES-IR の適用, 土木学会応用力学論文集, Vol.6, pp.275-281, 2003.
- 6) 巻幡憲俊, 宇都宮智昭, 渡邊英一: 超大型浮体の波浪応答解析への GMRES-DR の適用, 土木学会構造工学論文集, Vol.50A, pp.179-185, 2004.
- 7) Y. Saad: ILUT: a dual threshold incomplete LU factorization, *Numer. Linear Algebra Appl.*, 1, pp.387-402, 1994.
- 8) N. Li, Y. Saad, and E. Chow: Crout version of ILU for general sparse matrices, *SIAM J. Sci. Comput.*, 25, pp.716-728, 2003.
- 9) 森正武, 杉原正顕, 室田一雄: 線形計算「岩波講座, 応用数学」, 岩波書店, 1994.
- 10) Z. Zlatev: Computational Methods for General Sparse Matrices, *Kluwer, Dordrecht, the Netherlands*, 1991.
- 11) N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen: How fast are nonsymmetric matrix iterations?, *SIAM J. Matrix Anal. Appl.*, 13, pp.778-795, 1992.
- 12) 田端正久: 微分方程式の数値解法 II「岩波講座, 応用数学」, 岩波書店, 1994.
- 13) R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, *SIAM, Philadelphia*, 1994. (長谷川里美 他 訳: 反復法 Templates, 朝倉書店, 1996.)
- 14) T. Utsunomiya, E. Watanabe, and N. Nishimura: Fast multipole algorithm for wave diffraction/radiation problems and its application to VLFS in variable water depth and topography, *Proc. 20th Int. Conf. on Offshore Mech. & Arctic Eng.*, OMAE01-5202, 2001.
- 15) 宇都宮智昭, 渡邊英一: 超大型浮体の波浪応答解析のための高速化グリーン関数法の開発, 第 16 回海洋工学シンポジウム, pp.313-320, 2001.
- 16) 西村直志, 宗像秀紀: 多重極境界積分方程式法の簡易並列化について, 京都大学学術情報メディアセンター広報, Vol.3, pp.30-34, 2004.
- 17) M. Benzi, and M. Tuma: A sparse approximate inverse preconditioner for nonsymmetric linear systems, *SIAM J. Sci. Comput.*, 19, pp.968-994, 1998.
- 18) M. Bollhöfer, and Y. Saad: On the relations between ILUs and factored approximate inverses, *SIAM J. Matrix Anal. Appl.*, 24, pp.219-237, 2002.

(2004 年 4 月 16 日 受付)