

高速性と信頼性を両立させる AC-IDR(s)法の提案と評価

櫻井隆雄^{†1} 直野 健^{†1} 恵木正史^{†1}
猪貝光祥^{†2} 木立啓之^{†2} 小路将徳^{†2}

科学技術計算において大規模行列の連立一次方程式の求解は最も時間を要する計算の1つであり、その高速な解法はつねに求められている。近年、IDR(s)法という新たな連立一次方程式解法が提案された。この解法は従来のものより高速だが、稀に出力される解が要求精度を満たさずに偽収束するという問題があった。本稿ではこの偽収束の原因が演算量削減を目的とする近似演算による誤差だと解明し、誤差の発生を事前予測して近似演算の使用を自動的に判断するチューニング方式を実装した Auto Corrected-IDR(s)法を提案した。標準の行列を用いた数値実験の結果、出力された解が偽収束せずに要求精度を満たした割合が、従来法は61%であるのに対し提案法は100%を達成できた。

Proposal of Auto-corrected IDR(s) Method for Highly Accurate Krylov Iterative Solvers

TAKAO SAKURAI,^{†1} KEN NAONO,^{†1} MASASHI EGI,^{†1}
MITSUYOSHI IGAI,^{†2} HIROYUKI KIDACHI^{†2}
and MASANORI SHOJI^{†2}

Recently, the IDR(s) method has been emerged as a high performance iterative solver. However, the method occasionally outputs incorrect solutions. To alleviate the problem, we propose an auto-tuning type IDR(s) method, which we call "Auto-corrected IDR(s) method" (AC-IDR(s)). To avoid the incorrectness from the approximation of the original IDR(s), AC-IDR(s) predicts the occurrence of the incorrectness using the residual norm statistics and automatically replaces the approximation for the direct matrix vector multiplication. Numerical experiments show that the AC-IDR(s) solutions avoid the incorrectness in all cases.

1. はじめに

電磁場や流体の解析などに代表される科学技術計算において、大規模な係数行列に対する連立一次方程式の求解は、最も計算時間を必要とする主要な演算である。そのため、連立一次方程式を高速に演算する方式はつねに必要とされている。

本稿では、連立一次方程式解法の1つであるIDR(s)法(Induced Dimension Reduction (s) method)¹⁾を題材とする。IDR(s)法はSonneveldとvan Gijzenにより最近提案された新しいKrylov部分空間を用いた反復解法であり、従来連立一次解法として使われてきたBiCG法²⁾やGMRES(m)法³⁾と比べて短い時間で解が求められると報告されている⁴⁾。IDR(s)法はGMRES(m)法などの他のKrylov部分空間法と同様に様々なパラメータを持ち、その中でもKrylov部分空間の次元数 s は特に重要なパラメータである。この s の値を大きくするとIDR(s)法が出力した近似解と連立一次方程式の真の解の誤差が、使用者の入力した要求精度と比べて非常に大きくなる偽収束がまれに発生するという問題がある。この問題はつねに入力された要求精度よりも誤差の小さい近似解を出力しなければならないという解法の要件を満たすうえで障害となる。よって、この問題の解決がIDR(s)法の大きな課題となっている⁵⁾。

そこで、本稿ではIDR(s)法において上記の問題の原因を解明し、IDR(s)法の短い時間で解を求められる高い性能を維持したうえで、偽収束の発生を抑制しつつに要求精度を満たす解を得られる方式を実現することを目的とする。

2. IDR(s)法の概要とその問題点

2.1 IDR(s)法のアルゴリズム

IDR(s)法のアルゴリズムについて述べる。IDR(s)法は従来連立一次方程式の解法として使われてきたGMRES(m)法やBiCG法と同様のKrylov部分空間法である。IDR(s)法のアルゴリズムは図1のようになる。

2.2 s の増大により発生するIDR(s)法の問題

IDR(s)法におけるパラメータ s はKrylov部分空間の初期次元数を決定するパラメータ

^{†1} 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi, Ltd.

^{†2} 株式会社日立超 LSI システムズ
Hitachi ULSI Systems Co., Ltd.

2 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

```

1: Let  $x_0$  be an initial guess, and put  $r_0 = b - Ax_0$ 
2: DO  $k = 0, s-1$ 
3:  $v_k = Ar_k$ 
4:  $\omega = \frac{(v_k, r_k)}{(v_k, v_k)}$ 
5:  $q_k = \omega r_k, e_k = -\omega v_k$ 
6:  $r_{k+1} = r_k + e_k, x_{k+1} = x_k + q_k$ 
7: CONTINUE
8:  $E_k = (e_{s-1}, \dots, e_0), Q_k = (q_{s-1}, \dots, q_0)$ 
9: DO  $k = s, iter$ 
10: SOLVE  $c_k$  FROM  $P^T E_k c_k = P^T r_k$ 
    
$$P = (p_1, \dots, p_s), p_i = r_i / \|r_i\|$$

    
$$(p_i, p_j) = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$$

11:  $v_k = r_k - E_k c_k$ 
12: IF  $\text{mod}(k, s+1) = s$  THEN
13:  $t_k = Av_k$ 
14:  $\omega = \frac{(t_k, v_k)}{(t_k, t_k)}$ 
15:  $q_k = -Q_k c_k + \omega v_k$ 
16:  $e_k = -E_k c_k - \omega t_k$ 
17: ELSE
18:  $q_k = -Q_k c_k + \omega v_k$ 
19:  $e_k = -Aq_k$ 
20: END IF
21:  $r_{k+1} = r_k + e_k, x_{k+1} = x_k + q_k$ 
22: IF  $\|r_{k+1}\|_2 / \|b\|_2 \leq \varepsilon$  THEN EXIT DO
23: CONTINUE

```

図 1 IDR(s) 法のアルゴリズム

Fig. 1 Algorithm of IDR(s) method.

である。s が大きくなると Krylov 部分空間の次元数が増加するため 1 反復あたりの解の探索範囲が広がり、収束に必要な反復回数が減少する。一方で、s が大きくなると 1 反復あたりの演算量が増加するため、IDR(s) 法全体の演算時間が増加することがある。また、演算に必要とするメモリ量も s にもない増加するという問題がある。

さらに、s が大きくなると、より深刻な問題が発生する。IDR(s) 法に限らずあらゆる反復解法は、ソルバが収束と見なしたときに出力された近似解ベクトル x_n に対して、真の解との相対残差の 2 ノルム $\|b - Ax_n\|_2 / \|b\|_2$ (以後、これを「真の相対残差 r_t 」と呼ぶ) が

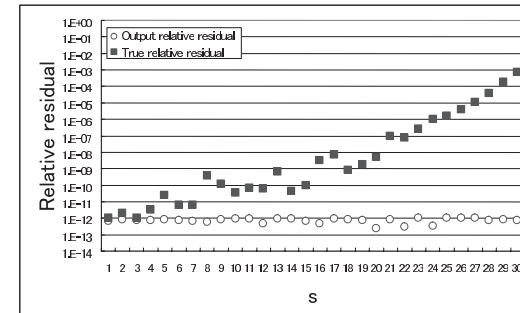


図 2 出力した残差と真の相対残差

Fig. 2 Output and true relative residual.

必ず要求精度 ε を下回っていないなければならない。しかし、IDR(s) 法は、s が増加すると真の相対残差 r_t が要求精度よりも大きくなる偽収束という現象が発生するという問題がある。また、このときソルバの出力する残差 r_n のノルムは ε を下回っており、ライブラリ使用者は真の相対残差 r_t を求める検算をしなかった場合、偽収束の発生に気づかずに精度が不十分な解を別の演算に用いる恐れがある。

図 2 にある行列 ex26 を入力としたときの各 s におけるソルバの出力した残差 r_n と真の相対残差 r_t のそれぞれのノルムを示した。なお、この行列 ex26 はフロリダ大学の Sparse Matrix collection に掲示されたものである。s の増加にもない r_t が増大していることが読み取れる。

文献 1) によると、内部パラメータ ω をベクトル v_k と t_k の角度 ρ を使い調整し、偽収束を抑制する方法が記述されている。この方法は図 1 の 14 行目で ω を算出する際に以下のように調整する。ここでパラメータ κ はユーザが静的に与える。

$$\rho = (t_k, v_k) / \|t_k\|_2 \|v_k\|_2$$

$$\text{if } |\rho| < \kappa \text{ then } \omega = \omega \cdot \kappa / \rho$$

しかし、上記の方法では完全に偽収束を防げないという測定結果も報告されており⁵⁾、より信頼できる偽収束抑制方法が必要とされている。

よって、本研究の目的は IDR(s) 法における偽収束発生の原因を特定し、それを完全に抑制する方式を考案することとした。上記の達成により高性能かつ解が信頼できる IDR(s) 法を実現できる。

3. 高速性と信頼性を両立させる方式の提案

3.1 偽収束発生の原因とその対策による課題

図 1 に示した IDR(s) 法のアルゴリズムにおいて、 k 回目の反復における残差 r_k の微小変化ベクトル e_k の算出法は、近似解ベクトル x_k の微小変化ベクトル q_k を用いて $-Aq_k$ の演算で求める直接演算と、 $-E_k c_k - \omega t_k$ の演算により求める近似演算の 2 通りが存在する。この 2 通りの算出法は、 $q_k = -Q_k c_k + \omega v_k$ であり、 $t_k = Av_k$ 、 $E_k = -AQ_k$ であることを考慮すると、

$$\begin{aligned} e_k &= -E_k c_k - \omega t_k \\ &= -(-AQ_k) c_k - \omega(Av_k) \\ &= -A(-Q_k c_k + \omega v_k) \\ &= -Aq_k \end{aligned}$$

となり、同値であることが分かる。行列 A とベクトルの乗算の演算量は非常に大きく、この回数をいかに少なくするかがアルゴリズムの高速性を高めるうえで重要となる。16 行の処理では、すでに 13 行で求めている $t_k = Av_k$ を利用して $-E_k c_k - \omega t_k$ の近似演算をする方が、 $-Aq_k$ を演算するよりも演算量が少ない。一方で、19 行の処理は t_k を求めていないため、 $-E_k c_k - \omega t_k$ の近似演算を用いても行列とベクトル乗算の回数は変わらず、 $-Aq_k$ を直接演算した方が演算量は少ない。

この 2 通りのうち、直接演算とした $-Aq_k$ では e_k の算出が q_k に依存している。それに対し、近似演算では e_k の算出が $-E_k c_k - \omega t_k$ であり、 q_k と独立で行われている。この近似演算により算出される e_k と $-Aq_k$ の間に誤差が生じる可能性があり、これが偽収束の発生に関連があると考えられる。

そこで、2 章の計測で用いた行列 ex26 を入力としたときの各 s について、収束までの $-E_k c_k - \omega t_k$ と $-Aq_k$ の誤差 $\|(-E_k c_k - \omega t_k) - (-Aq_k)\|_2 / \|b\|_2$ (以後、これを近似誤差と呼称する) の演算開始から終了までの合計値を測定した。図 3 にその結果と真の相対残差 r_t の比較を示す。

図 3 から近似誤差の合計が r_t と一致していると読み取れる。よって IDR(s) 法で偽収束が発生するのは e_k 算出において演算量削減を目的として近似演算を用いて求めるためであり、つねに直接演算を用いればこれを抑制できると考えられる。

しかし、近似演算を用いるのは演算量を削減するためであり、それを用いない場合、ソルバ全体の演算量が増加するという新たな問題が生じる。また、IDR(s) 法では近似演算は

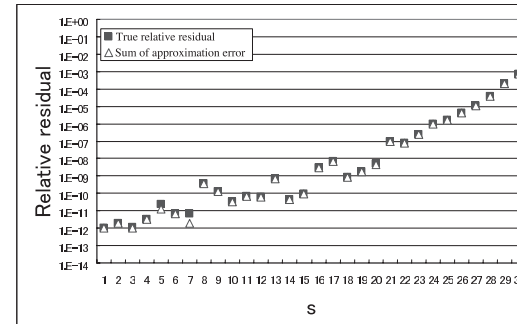


図 3 近似誤差の合計と真の相対残差の関係

Fig. 3 Relationship between sum of approximation error and true relative residual.

$s+1$ 反復に 1 回行われていたため、 s が小さいときは r_t が十分小さいにもかかわらず、求解性能の劣化が顕著となると考えられる。実際、入力される要求精度 ε が通常は $10^{-8} \sim 10^{-12}$ 程度であることを考慮すると、 $-E_k c_k - \omega t_k$ と $-Aq_k$ の誤差が 10^{-12} 以上となり、偽収束発生につながるケースは 1 割に満たない。そのため、大部分は e_k を近似演算で算出して問題がない。

そこで、偽収束発生を予測する何らかの指標を用いて自動的に近似演算と直接演算を使い分けるチューニング方式を用いれば IDR(s) 法の高い演算性能を維持したままつねに要求精度を満たす解を出力できると考えられる。このチューニング方式実現のためには、偽収束の発生を e_k 算出の前に予測する指標を策定する必要がある。この指標の策定が高速性と解の信頼性を両立させる IDR(s) 法実現のための新たな課題となる。

3.2 偽収束発生を予測する指標の策定

本節では前節で述べた偽収束発生を予測する指標を策定する。IDR(s) 法は、 s が大きくなると、Krylov 部分空間の次元数が増大し、探索空間が広大になるため収束が早まるが、相対残差ノルムは反復回数が少ない場合、一時的に非常に大きな値をとるときがある。図 4 に行列 ex26、 $s = 10$ における相対残差ノルムの履歴を示す。図 4 の場合、相対残差ノルムが一時的に 10^4 を超えていると分かる。

この高い相対残差ノルムが近似誤差に影響しているか確認するため、その関係を調査した。調査に使用した入力行列は表 1 の 4 つである。これらは Sparse Matrix collection より取得した。これらの行列を IDR(s) 法で e_k の算出に $-E_k c_k - \omega t_k$ を用いた際の $-Aq_k$ との近似誤差とその時点での相対残差ノルムの散布図を図 5 に示した。パラメータ s は各行

4 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

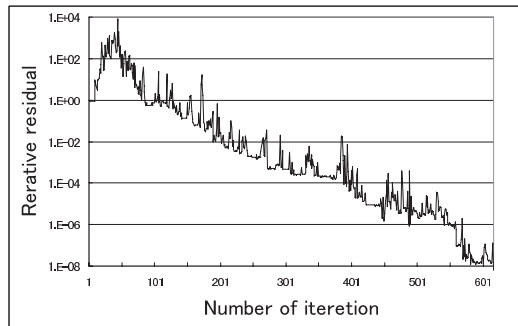


図 4 行列 ex26, $s = 10$ の相対残差ノルムの履歴

Fig. 4 Residual history for matrix ex26 by IDR(s) method when parameter s was 10.

表 1 調査に使用した行列

Table 1 Test matrices for examination.

行列	次元数	非零要素数	解析分野
ex26	2163	94033	流体力学
ex28	2603	77781	流体力学
raefsky2	3242	293551	流体力学
wang1	2903	19093	電気回路

列について 1~30 まで動かした．図 5 から大部分において近似誤差と相対残差ノルムは比例関係にあると読み取れる．これは相対残差ノルムが大きくなると， $e_k (= r_{k+1} - r_k)$ のノルムも大きくなるため，その時点の e_k から見れば微小であるはずの誤差が収束時には無視できない大きさになるためと考えられる．

一方で，図 5 から ex26, ex28, wang1 の 3 つの行列について相対残差ノルムが 1 となる近辺において，近似誤差との相関から大きく外れた点が存在する．これらは最初に Krylov 部分空間を生成する反復 s 回目の点である．

単純な解決策として反復 s 回目はつねに直接演算を用い，以後は相対残差ノルムを指標として近似演算と直接演算を使いかける方式が考えられる．しかし，行列 raefsky2 では s 回目の反復時でも近似誤差がさほど大きな値となっていないこと，他の 3 つの行列においても反復 s 回目における近似誤差の大きさに幅があることから，反復 s 回目における近似誤差の大きさを示す指標があると考えられる．

そこで，ex26, ex28, raefsky2 につき， $s = 30$ ，反復 30 回目における様々な行列，ベク

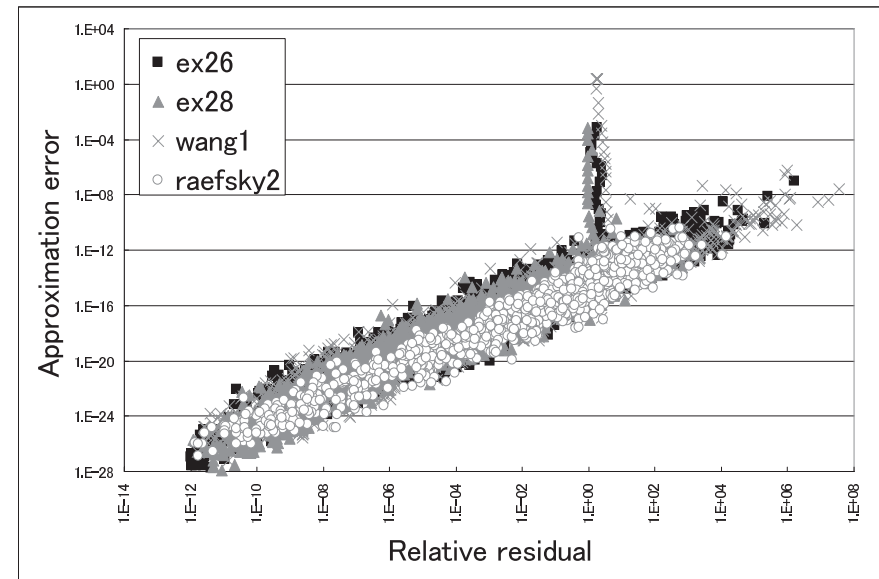


図 5 近似誤差と相対残差ノルムの関係

Fig. 5 Relationship between approximation error and relative residual norm.

トルを比較したところ，ベクトル c_k の要素のとり値の範囲に顕著な差異があると分かった．ex26, ex28 と raefsky2 について， $s = 30$ ，反復 30 回目におけるベクトル c_k の要素の最大値を最小値で割った値（要素幅）を図 6 に示す．近似誤差の大きい ex26, ex28 では要素幅は 10^{12} 以上であり，非常に広い．一方，近似誤差の小さい raefsky2 では要素幅は 10^6 以下であり比較的狭い．

そこで，図 7 に表 1 の行列，パラメータ $s = 1 \sim 30$ について近似誤差とベクトルの要素幅の関連を表す散布図を示した．図 5 とは逆に反復 s 回目の点に強い相関があり，その他の点が無相関となっている．図 5 と図 7 より， s 回目の反復時以外で強い相関のある相対残差ノルムと s 回目の反復時のみに強い相関のあるベクトルの要素の幅を組み合わせることで近似誤差の大きさを示す指標になると期待できる．

単純な組合せ方式として反復 s 回目はベクトルの要素幅で判定し，それ以降は相対残差ノルムで判定するものが考えられる．しかし，図 8 に示したように [相対残差ノルム] \times [c_k の要素幅] と近似誤差の関係を検査したところ，反復 s 回目以外の点についてもより高い相

5 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

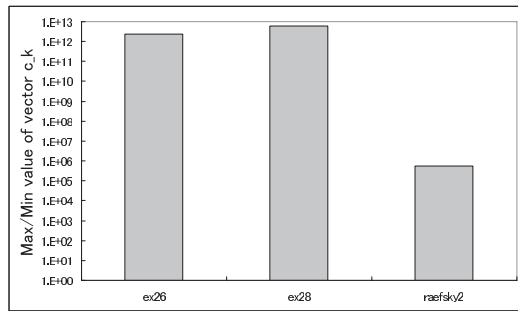


図 6 $s = 30$, 反復 30 回目におけるベクトル c_k の要素幅

Fig. 6 Max/Min value of vector c_k element when parameter s and number of iteration were 30.

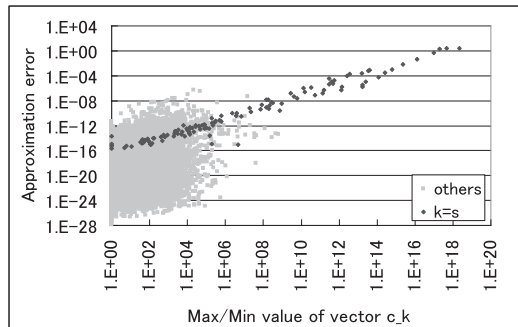


図 7 近似誤差とベクトル c_k の要素幅の関係

Fig. 7 Relationship between approximation error and Max/Min value of vector c_k element.

関が見られた。実際、反復 s 回目を除いた点について相関係数を計算すると、相対残差ノルムのみでは 0.15 であるのに対し、[相対残差ノルム] \times [c_k の要素幅] は 0.54 となっていた。

そこで、本稿では直接演算と近似演算と使い分けるための指標 I として相対残差ノルムとベクトル c_k の要素幅の積を用いることとした。

3.3 Auto Corrected-IDR(s) 法の提案

前節で策定した指標を使い、高速性と解の信頼性の双方を満たす Auto Corrected-IDR(s) 法 (以下、AC-IDR(s) 法) を提案する。

従来の IDR(s) 法との相違点は、図 9 に示すように図 1 のアルゴリズムの 16 行における e_k 算出の際に指標 I が I_{th} 以下ならば近似演算 $-E_k c_k - \omega t_k$, I_{th} 以上ならば直接演算

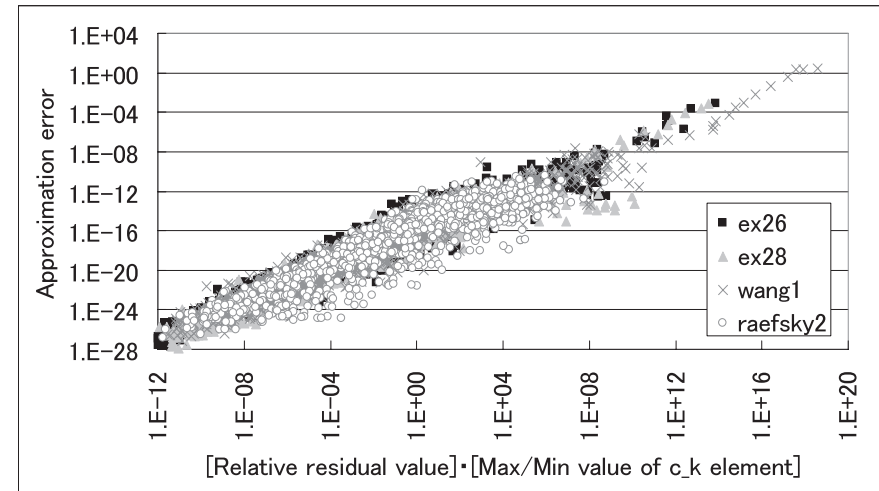


図 8 近似誤差と相対残差ノルム \times [c_k の要素幅] の関係

Fig. 8 Relationship between approximation error and [Relative residual value] \cdot [Max/Min value of c_k element].

16: IF $I \leq I_{th}$ THEN $e_k = -E_k c_k - \omega t_k$
ELSE $e_k = -A q_k$

$$[I = (\|r_k\|_2 / \|b\|_2) \times (c_{k_max} / c_{k_min})]$$

図 9 AC-IDR(s) 法のアルゴリズム変更部分

Fig. 9 Change part of AC-IDR(s) algorithm.

$-A q_k$ で求める点である。直接演算と近似演算のどちらを用いるかの閾値 I_{th} はチューニング方式のために新たに設定しなければならないパラメータとなる。しかし、図 8 から I_{th} を要求精度 ε の 10^{11} 倍から 10^{12} 倍に設定すればよいと読み取れるため、 I_{th} はライブラリ作成時にあらかじめ値を決められ、使用者からは見えない内部パラメータにできる。

また、この方式は新たに必要となる演算量とメモリ量が非常に少ないという特徴がある。主な演算は指標 I を求める部分だが、それに必要となる相対残差ノルムは収束判定時のものを使えばよく、ベクトル c_k の要素幅も $s \times 2$ 回の大小比較演算と 1 回の除算で求まる。メモリ確保もベクトル c_k の最大、最小要素の実数 2 個分である。

この AC-IDR(s) 法により、IDR(s) 法の利点である高速性を維持しながらつねに要求精

度を満たす解を出力することが期待できる。

4. 数値実験

4.1 実験条件

前章で提案した AC-IDR(s) 法の有効性を確認するため、従来の IDR(s) 法と e_k 算出に つねに直接演算を用いた Direct-IDR(s) 法（以下、D-IDR(s) 法と呼ぶ）の 2 方式に対して演算時間と解の信頼性を評価した。表 2 に実験環境および IDR(s) 法のパラメータを示す。指標 I の閾値 I_{th} は ε を 10^{-10} としたため、その 10^{11} 倍となる 10 と設定した。また、実験に使用した行列を表 3 に示す。これらの行列は MatrixMarket と Sparse Matrix Collection より入手した。表 3 の非零/次元は非零要素数を次元数で割った値である。

表 2 実験のパラメータ
Table 2 Parameters for evaluations.

CPU	Opteron 2.0 GHz
Memory	16 GB (DDR2-667)
s	1 ~ 30 pitch 1
要求精度 ε	10^{-10}
行列 P	初期残差ベクトル r_0 と $s - 1$ 個の乱数ベクトルを正規直交化したもの
右辺項 b	解の全要素が 1 となるベクトル
初期近似解	全要素が 0 のベクトル
閾値 I_{th}	10

表 3 実験に使用した行列
Table 3 Test matrices for evaluations.

行列	次元数	非零要素数	非零/次元	解析分野
ex26	2163	94033	43.5	流体力学
ex28	2603	77781	29.9	流体力学
raefsky2	3242	293551	90.8	流体力学
wang1	2903	19093	6.6	電気回路
wang4	26088	177196	6.8	電気回路
memplus	17758	126150	7.1	電気回路
poisson3Da	13514	352762	26.1	構造解析
poisson3Db	85623	2374949	27.7	構造解析

4.2 実験結果

行列 poisson3Da についてパラメータ s を 1 から 30 まで動かした際の従来の IDR(s) 法、D-IDR(s) 法、AC-IDR(s) 法の 3 方式の演算時間を図 10、出力された解の真の相対残差 r_t の 2 ノルムを図 11 に示した。

図 10 から D-IDR(s) 法の演算時間は従来の IDR(s) 法と比べて最大で 30% 程度増加しているが、AC-IDR(s) 法は演算時間増加が D-IDR(s) 法と比べて抑制されていることが読み取れる。この演算時間増加の抑制効果は特に s が小さい値のときに顕著で、従来の IDR(s) 法と同等の演算時間であった。一方、図 11 から従来の IDR(s) 法が $s = 25$ より大きくなると真の相対 AC-IDR(s) 法、D-IDR(s) 法はともにすべての s で要求精度を満たしていたと

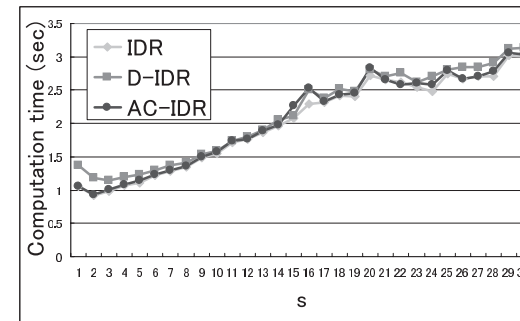


図 10 3 方式による行列 poisson3Da の演算時間
Fig. 10 Computation time for matrix poisson3Da by 3 methods.

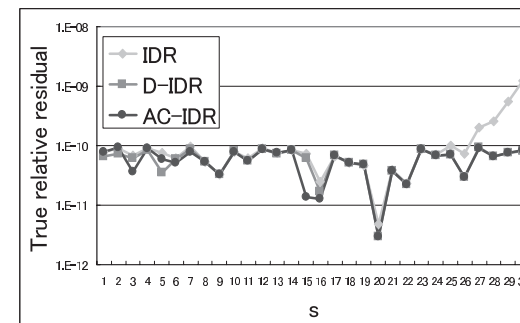


図 11 3 方式による行列 poisson3Da の真の相対残差
Fig. 11 True relative residual for matrix poisson3Da by 3 methods.

7 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

表 4 3方式の演算時間の比率と正答率

Table 4 Computation time and correct convergence rate by 3 methods.

行列	非零/次元	演算時間の比率			正答率 (%)		
		IDR	D-IDR	AC-IDR	IDR	D-IDR	AC-IDR
ex26	43.5	1.000	1.101	1.036	56.7	100.0	100.0
ex28	29.9	1.000	1.063	0.984	60.0	100.0	100.0
raefsky2	90.8	1.000	1.087	1.017	96.7	100.0	100.0
wang1	6.6	1.000	1.018	1.000	23.3	100.0	100.0
wang4	6.8	1.000	1.070	1.036	53.3	100.0	100.0
memplus	7.1	1.000	1.025	0.991	30.0	100.0	100.0
poisson3Da	26.1	1.000	1.068	1.019	83.3	100.0	100.0
poisson3Db	27.7	1.000	1.115	1.058	86.7	100.0	100.0
平均値		1.000	1.068	1.018	61.3	100.0	100.0

表 5 最適の s を選択した際の演算時間と反復回数

Table 5 Computation time and number of iteration in the case of optimized s.

行列	IDR			D-IDR			AC-IDR		
	最適 s	時間 (sec)	反復回数	最適 s	時間 (sec)	反復回数	最適 s	時間 (sec)	反復回数
ex26	1	0.41	647	2	0.52	599	1	0.42	647
ex28	1	0.24	468	1	0.30	460	1	0.24	468
raefsky2	5	1.10	478	8	1.21	444	5	1.10	474
wang1	1	0.13	615	1	0.16	622	1	0.13	615
wang4	1	2.45	688	2	2.82	565	1	2.46	688
memplus	1	6.47	2875	3	8.29	2280	1	6.54	2875
poisson3Da	2	0.92	211	3	1.14	211	2	0.93	211
poisson3Db	3	17.69	400	4	21.21	402	3	17.92	403

読み取れる。

その他の行列の結果を表 4 と表 5 にまとめた。表 4 の「演算時間の比率」は従来の IDR(s) 法の演算時間を 1 としたときの D-IDR(s) 法, AC-IDR(s) 法の演算時間の比について $s = 1 \sim 30$ での平均値を示し, 「正答率」は $s = 1 \sim 30$ で真の残差 r_t が正しく要求精 10^{-10} を満たしていた割合を示す。

表 4 から, AC-IDR(s) 法は従来の IDR(s) 法から平均で 1.8% 演算時間が増加するが, すべての行列で正答率が 100% となった。従来の IDR(s) 法は評価した 3 方式の中で平均して最も高い速度が得られたが, 正答率は 61% であり, 解の信頼性が低かった。D-IDR(s) 法は AC-IDR(s) 法と同様に正答率は 100% であったが, 従来の IDR(s) 法と比較した演算時間の

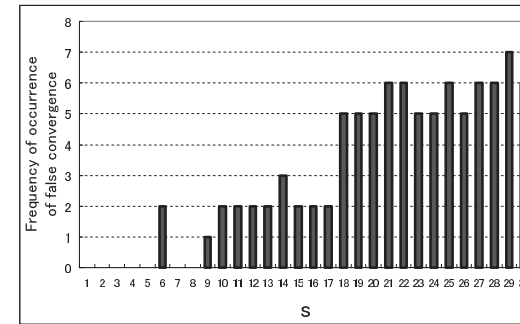


図 12 各 s における偽収束の発生数

Fig. 12 Frequency of occurrence of false convergence with each s.

増加は平均で 6.8% であり, この点で AC-IDR(s) 法に劣っていた。

また, 表 5 によると最適の s が選択されたときの演算時間は, D-IDR(s) 法は IDR(s) 法に対して 10~28% 増加していた。一方, AC-IDR(s) 法は半分以上の行列において同等の性能であり, その他の行列についてもただか 1.3% の増加となっていた。この結果から, 最適な値を選択した際においても AC-IDR(s) 法は従来の IDR(s) 法の演算性能を損なっていないと分かる。

以上から, 提案法の AC-IDR(s) 法はつねに要求精度を満たす解を出力し, 演算性能においても従来の IDR(s) 法と同等となる有効な方式だと確認できた。

また, 図 12 にこの実験で用いた 8 つの行列を従来の IDR(s) 法で解いた際の各 s に対する偽収束の発生頻度を示した。図 12 によると s が 5 以下の場合には偽収束が発生せず, 20 以上になると半数以上の行列で偽収束が発生している。この結果によると AC-IDR(s) 法が従来の IDR(s) 法に対して優位となるのは s が 6 以上の場合となる。

偽収束を抑制する既知の方法として 2.2 節で述べた内部パラメータ ω をベクトル v_k と t_k の角度 ρ を使い偽収束を抑制する方法がある。これに対する提案法の優位性を確認するため比較を行った。入力行列は前の実験で用いた 8 つの行列の中で最も正答率の低かった wang1 を用いた。その結果を図 13 と図 14 に示した。図中の「RHO」が ρ を用いた方式である。ここで, パラメータ κ は文献 1) で推奨されていた 0.7 とした。

図 13 から ρ を用いた方式は演算時間を変化させない対策法であると読み取れる。一方で, 図 14 から読み取れるように ρ を用いた方式には一定の偽収束抑制効果が見られるものの十分ではない。よって, 偽収束を抑制するという観点で提案法である AC-IDR(s) 法が優位で

8 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

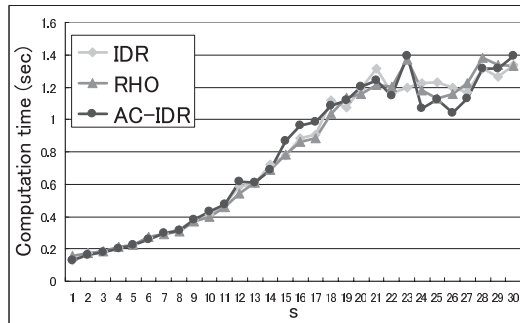


図 13 行列 wang1 における従来法, ρ を用いた方式, 提案法の演算時間

Fig. 13 Computation time for matrix wang1 by previous, using ρ and proposed method.

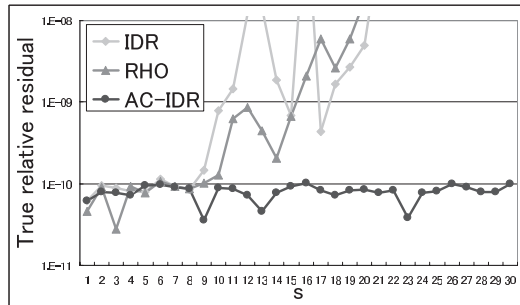


図 14 行列 wang1 における従来法, ρ を用いた方式, 提案法の真の相対残差

Fig. 14 True relative residual for matrix wang1 by previous, using ρ and proposed method.

あると考えられる。

最後に、行列の特性と偽収束発生率の関連について述べる。表 4 において、行列ごとの傾向を見ると、次元数に比して非零要素数が多い、すなわち 1 行あたりの平均非零数の多い raefsky2 は従来の IDR(s) 法での正答率が高く、反対に平均非零数が特に少ない wang1 と memplus は正答率が他と比して低くなっている。一方で、同様に平均非零数が特に少ない wang4 は他と比して正答率が低くはなっておらず、平均非零数のみで正答率は予測できない。この行列の内部構造と正答率の関連の調査は今後の課題となる。

5. ま と め

本稿では、近年提案され、その高い演算性能で注目されている IDR(s) 法を題材とした。IDR(s) 法におけるパラメータ s の増大にともない偽収束が発生するという問題に対し、その原因究明と解決に取り組んだ。その結果、偽収束の原因が残差ベクトルの微小変化ベクトル算出時の近似演算であると突き止めた。

しかし、近似演算を利用しなければ IDR(s) 法の高い演算性能が損なわれるという新たな問題が発生する。そこで、偽収束発生を予測する統計的指標を策定し、その指標により近似演算と直接演算を自動的に切り替えるチューニング方式を考案した。さらに、その方式を適用することで高い演算性能と解の信頼性を兼ね備えた Auto Corrected-IDR(s) 法を提案した。

数値実験による評価の結果、従来法では解が要求精度を満たす割合が 61%であったのに対し、提案法はつねに解が要求精度を満たしており、有効な方式であると確認できた。

今後の課題として次のことが考えられる。まず、本稿で用いた指標は相対残差ノルムとベクトル c_k の要素幅を単純にかけたものである。しかし、たとえば積算する際に重み付けなどを加えることにより、さらに強い関係のある指標が得られる可能性があり、検討の価値がある。また、今回の実験結果によると偽収束の発生が行列の外部特性により予測できる可能性が示唆されている。この点についても調査を進める。

参 考 文 献

- 1) Sonnenveld, P. and van Gijzen, M.B.: IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, TR 07-07, *Dept. Math. Anal.*, Delft, The Netherlands, pp.1-28 (2007).
- 2) van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.13, No.2, pp.631-644 (1992).
- 3) Saad, Y. and Schultz, M.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.7, No.3, pp.856-869 (1986).
- 4) 中嶋徳正, 藤野清次, 立居場光生, 尾上勇介: 多数の誘電体円柱の電磁波散乱問題の高速計算について(3)—IDR(s) 法と GMRES 法との収束性およびメモリ量の比較, SACSIS 2008—2008 年先進的計算基盤システムシンポジウム, pp.123-130 (2008).
- 5) 土持秀之, 尾上勇介, 藤野清次: IDR(s) 法の有用性の実験的検証—GMRES(k) との

9 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価

収束性比較, 応用数学合同研究集会報告集, pp.260-265 (2007).

(平成 20 年 10 月 3 日受付)

(平成 21 年 1 月 21 日採録)



櫻井 隆雄 (正会員)

1980 年生。2003 年東京大学工学部電子情報工学科卒業。2005 年同大学大学院情報理工学研究科電子情報学専攻修士課程修了。同年 (株) 日立製作所入社。以来, 並列計算機向け行列ライブラリ, 業務モニタリングシステムの研究開発に従事。



直野 健 (正会員)

1968 年生。1994 年京都大学大学院理学研究科数理解析専攻修士課程修了。同年 (株) 日立製作所中央研究所入社。以来, 並列計算機 SR2201, SR8000, SR11000, SR16000 向け行列計算ライブラリ, 自動チューニング, 業務モニタリングシステムの研究開発に従事。日本応用数理学会, 日本金融・証券計量・工学学会, 日本計算工学会各会員。



恵木 正史

1971 年生。1994 年名古屋大学工学部応用物理学専攻卒業。2000 年同大学大学院理学研究科素粒子宇宙物理学専攻満了。同年 (株) 日立製作所入社。以来, 情報システムの性能評価, 金融工学, 遺伝統計学および業務モニタリングシステムに関する研究開発に従事。



猪貝 光祥 (正会員)

1963 年生。1987 年横浜市立大学文理学部物理課程卒業。同年, 現 (株) 日立超 LSI システムズ入社。以来, 科学技術計算用ソフトウェアおよびその並列化手法に関する研究開発に従事。



木立 啓之

1977 年生。2000 年室蘭工業大学情報工学科卒業。同年 (株) 日立超 LSI システムズ入社。以来, 科学技術計算用ソフトウェアおよびその並列化手法に関する研究開発に従事。



小路 将徳

1976 年生。2000 年広島大学総合科学科卒業。2002 年広島大学生物圏科学研究科卒業。同年 (株) 日立超 LSI システムズ入社。以来, 科学技術計算用ソフトウェアおよびその並列化手法に関する研究開発に従事。