Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol II,
IMECS 2012, March 14 - 16, 2012, Hong Kong

High Performance Computation of Iterative Methods for Matrices Appear in The Field of Realistic Electromagnetics

Takashi Sekimoto*, Seiji Fujino†

Abstract—**We estimate performance of the conventional and several hybrids of product-type iterative method for solution of realistic electromagnetic problems. Moreover, performance of new coming iteartive methods based on IDR(s) method will be also examined. As a result of total ranking on performance, we will state what iterative method is the most effective through numerical experiments.**

Keywords: iterative method, BiCGSafe method, GBiCGSTAB(s, L) method, electromagnetic problems, preconditioning

1 Introduction

We consider to solve efficiently a linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$ by state-of-the-art iterative methods. Here A means a large, sparse and real nonsymmetric coefficient matrix, and $\boldsymbol{x}, \boldsymbol{b}$ is the solution vector and right-hand side vector, respectively. Among many iterative methods, product-type of iterative methods e.g., BiCGStab(Bi-Conjugate Gradient Stabilized)[9] and GPBiCG(Generalized Product-type BiCG)[10] are often used for the purpose of solution for realistic problems. They constitute a sequence of polynomial by multiplying Lanczos polynomials by so-called acceleration polynomial. The acceleration polynomial is categorized into two groups according to number of term of recurrence That is, BiCGSTAB method is generated by two-term recurrence only, and GPBiCG method is generated by three-term recurrence only. In addition, it must be noted that we meet with instability of GPBiCG method in many numerical experiments.

On the contrary, BiCGSTAB2 [3] method was proposed by M. Gutknecht in 1993. Moreover, same name of BiCGSTAB2 method was proposed as a hybrid version of GPBiCG method by Zhang [10] in 1997. These two types of BiCGSTAB2 methods consist of combinated Lanczos polynomial and acceleration polynomial, and were used for solution of many electromagnetics problems. It is, however, performance of BiCGSTAB2 methods are demanded for solution of the large scale problems [11]. Recently, on the other hand, various product-type of iterative methods were proposed one after another, e.g., BiCGSafe [2], GPBiCG_AR [4] by the authors and GPBiCG_variant [1] by K. Abe *et al.* Moreover, a family of IDR(s) [8] such as BiIDR(s) [7] and GBiCGSTAB (s, L) [6] has attracted attention. Therefore, these iterative methods have possibility to supply a demand for gaining high performance computing

In this research, we have two objectives. One of them is that we will implement hybrid methods of BiCGSafe, GPBiCG_AR and GPBiCG_variant methods as well as BiCGSTAB2 methods. Another of them is that we will evaluate convergence rate of a series of product-type iteratve method, and a family of IDR(s), and hybrid methods for matrices appear in the field of realistic electromagnetics problems.

This paper is organized as follows: In section 2, a brief outline of BiCGsafe2 method as hybrid of the original BiCGSafe method with combination of two-term and three-term recurrences will be described. In section 3, a short description on GBiCGStab(s, L) method including GBiCG part and MR part will be done. In section 4, computational cost of some iterative methods will be estimated. In section 5, several results of iterative methods will be shown, and it will be made clear that what is the most effective iterative methods through numerical experiments. Finally, in section 6, we have concluding remarks.

2 Hybrid version of BiCGSafe method

We treat with iterative methods for solving a linear system of equations

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where $A \in R^{N\times N}$ is a given nonsymmetric matrix, and $\boldsymbol{x}$, $\boldsymbol{b}$ is a solution vector and right-hand side vector, respectively. The parameters ζ_k, η_k which are included in acceleration polynomial of the original BiCGSafe method are decided from the local minization of the associate residual vector of 2-norm as $||\boldsymbol{a}\_\boldsymbol{r}_k||_2 = ||H_{k+1}(\lambda)R_{k+1}(\lambda)||_2$.

*Graduate School of Information Science and Electrical Engineering, Kyushu University
†Research Institute for Information Technology, Kyushu University Email:fujino@kyushu-u.ac.jp

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The associate residual $\boldsymbol{a\_r}_k$ is defined as follows:

$$\boldsymbol{a\_r}_k = \boldsymbol{r}_k - \zeta_k A\boldsymbol{r}_k - \eta_k \boldsymbol{y}_k. \quad (2)$$

Here $\boldsymbol{r}_k$ denotes the residual vector of the algorithm, and $\boldsymbol{y}_k$ denotes also an auxiliary vector. The parameters ζ_k, η_k of the original BiCGSafe method are computed as follows:

$$\zeta = \frac{(\boldsymbol{b}_k,\boldsymbol{b}_k)(\boldsymbol{c}_k,\boldsymbol{a}_k)-(\boldsymbol{b}_k,\boldsymbol{a}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)}{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{b}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)}, \quad (3)$$

$$\eta_n = \frac{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{a}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{a}_k)}{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{b}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)}, \quad (4)$$

where we impose that $\boldsymbol{a}_k = \boldsymbol{r}_k$, $\boldsymbol{b}_k = \boldsymbol{y}_k$, $\boldsymbol{c}_k = A\boldsymbol{r}_k$.

On the other hand, BiCGSafe2 method with similar property as BiCGSTAB2 and GPBiCG_AR alternates computation of parameters ζ_k, η_k of the original BiCGSafe method. In this approach, we set η_k to be zero at even iteration step. We present an algorithm of BiCGSafe2 method as below.

Algorithm of BiCGSafe2 method

Let $\boldsymbol{x}_0$ be an initial guess, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0, \boldsymbol{r}_0^*) \neq \boldsymbol{0}$, set $\beta_{-1} = 0$
for $k = 0, 1, \ldots,$ until $||\boldsymbol{r}_{k+1}|| \leq ||\boldsymbol{r}_0||$ do :
begin

$$\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_{k-1}(\boldsymbol{p}_{k-1} - \boldsymbol{u}_{k-1}) \quad (5)$$
$$A\boldsymbol{p}_k = A\boldsymbol{r}_k + \beta_{k-1}(A\boldsymbol{p}_{k-1} - A\boldsymbol{u}_{k-1}) \quad (6)$$
$$\alpha_k = (\boldsymbol{r}_k, \boldsymbol{r}_0^*)/(A\boldsymbol{p}_k, \boldsymbol{r}_0^*) \quad (7)$$
$$\boldsymbol{a}_k = \boldsymbol{r}_k,\ \boldsymbol{b}_k = \boldsymbol{y}_k,\ \boldsymbol{c}_k = A\boldsymbol{r}_k \quad (8)$$

if $\mathrm{mod}(k, 2) \neq 0$, then

$$\zeta_k = (\boldsymbol{c}_k, \boldsymbol{a}_k)/(\boldsymbol{c}_k, \boldsymbol{c}_k),\ \eta_k = 0 \quad (9)$$
$$\boldsymbol{u}_k = \zeta_k A\boldsymbol{p}_k + \beta_{k-1}\boldsymbol{u}_{k-1} \quad (10)$$
$$\boldsymbol{z}_k = \zeta_k \boldsymbol{r}_k - \alpha_k \boldsymbol{u}_k \quad (11)$$
$$\boldsymbol{y}_{k+1} = \zeta_k A\boldsymbol{r}_k - \alpha_k A\boldsymbol{u}_k \quad (12)$$

else

$$\zeta = \frac{(\boldsymbol{b}_k,\boldsymbol{b}_k)(\boldsymbol{c}_k,\boldsymbol{a}_k)-(\boldsymbol{b}_k,\boldsymbol{a}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)}{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{b}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)} \quad (13)$$

$$\eta_n = \frac{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{a}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{a}_k)}{(\boldsymbol{c}_k,\boldsymbol{c}_k)(\boldsymbol{b}_k,\boldsymbol{b}_k)-(\boldsymbol{b}_k,\boldsymbol{c}_k)(\boldsymbol{c}_k,\boldsymbol{b}_k)} \quad (14)$$

$$\boldsymbol{u}_k = \zeta_k A\boldsymbol{p}_k + \eta_k(\boldsymbol{y}_k + \beta_{k-1}\boldsymbol{u}_{k-1}) \quad (15)$$
$$\boldsymbol{z}_k = \zeta_k \boldsymbol{r}_k + \eta_k \boldsymbol{z}_{k-1} - \alpha_k \boldsymbol{u}_k \quad (16)$$
$$\boldsymbol{y}_{k+1} = \zeta_k A\boldsymbol{r}_k + \eta_k \boldsymbol{y}_k - \alpha_k A\boldsymbol{u}_k \quad (17)$$

end if

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k + \boldsymbol{z}_k \quad (18)$$
$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k - \boldsymbol{y}_{k+1} \quad (19)$$
$$\beta_k = \frac{\alpha_k}{\zeta_k}\frac{(\boldsymbol{r}_{k+1}, \boldsymbol{r}_0^*)}{(\boldsymbol{r}_k, \boldsymbol{r}_0^*)} \quad (20)$$

End

3 GBiCGSTAB(s, L) method

GBiCGSTAB(s, L) method is derived from GBiCG(s) methods by introducing L-degree stabilization polynomial. Computation per iteration of GBiCGSTAB(s, L) consists of GBiCG(s) part and MR(Minimum Residual) part. In GBiCGSTAB(s, L) method, residual vector $\boldsymbol{r}_k$, solution vector $\boldsymbol{x}_k$ and auxiliary matrix U_{k-1} are updated to $\boldsymbol{r}_{k+L}$, $\boldsymbol{x}_{k+L}$ and U_{k+L-1} respectively at every L iteration. Here, $k = mL(m = 1, 2, \ldots)$. We denote vectors $\boldsymbol{r}_k$ and matrices U_k of GBiCG(s) method as $\boldsymbol{r_k}^{\mathrm{GB}}$ and U_k^{GB}, respectively. Then, we set $\boldsymbol{r}_k$ and U_k as

$$\boldsymbol{r}_k = Q_k(A)\boldsymbol{r}_k^{\mathrm{GB}}, U_{k-1} = Q_k(A)U_k^{\mathrm{GB}}, \quad (21)$$

and define approximate solution vectors $\boldsymbol{x}_k$ and $\hat{\boldsymbol{x}}_k^{(i)}$ as

$$\boldsymbol{b} - A\boldsymbol{x}_k = \boldsymbol{r}_k = Q_k(A)\boldsymbol{r}_k^{\mathrm{GB}}, \quad (22)$$
$$\boldsymbol{b} - A\hat{\boldsymbol{x}}_k^{(i)} = \boldsymbol{r}_{k+i} = Q_k(A)\boldsymbol{r}_{k+1}^{\mathrm{GB}}. \quad (23)$$

Here, $Q_k(t) = p_m(t)\cdots p_2(t)p_1(t)$, product of MR polynomials $p_i(t)(i = 1, 2, \ldots, m)$.

3.1 GBiCG part

In GBiCG part, in k iteration, we update $A^jQ_k\boldsymbol{r}_{k+L}^{\mathrm{GB}}$, $A^jQ_kU_{k+L-1}^{\mathrm{GB}}$, and $\hat{\boldsymbol{x}}_k^{(L)}(j = 0, \ldots, L)$. Here, $k = mL(m = 1, 2, \ldots)$. First, we give $Q_k\boldsymbol{r}_k^{\mathrm{GB}}$, $Q_kU_{k-1}^{\mathrm{GB}}(j = 0, \ldots, i)$, and $\boldsymbol{x}_k$ to this part. Next, for i-th iteration$(i = 0, 2, \ldots, L-1)$, $A^jQ_kU_{k+i}^{\mathrm{GB}}(j = 0, \ldots, i)$ is updated from $A^jQ_kU_{k+i}^{\mathrm{GB}}\boldsymbol{e}_t = A^jQ_k\boldsymbol{r}_{k+i}^{\mathrm{GB}} - A^jQ_kU_{k+i-1}^{\mathrm{GB}}\beta_{k+1}(t = 0, \ldots, s)$, and update $A^jQ_k\boldsymbol{r}_{k+i+1}^{\mathrm{GB}}$ from $A^jQ_k\boldsymbol{r}_{k+i+1}^{\mathrm{GB}} = A^jQ_k\boldsymbol{r}_{k+i}^{\mathrm{GB}} - A^{j+1}Q_kU_{k+i}^{\mathrm{GB}}\alpha_{k+i}(j = 0, 1, \ldots, i)$. For $\hat{\boldsymbol{x}}_k^{(i)}$, we update such that $\hat{\boldsymbol{x}}_k^{(i+1)} = \hat{\boldsymbol{x}}_k^{(i)} + \alpha_{k+i}Q_kU_{k+i}^{\mathrm{GB}}$ Finally this part output $A^jQ_k\boldsymbol{r}_{k+L}^{\mathrm{GB}}$, $A^jQ_kU_{k+L-1}^{\mathrm{GB}}$, and $\hat{\boldsymbol{x}}_k^{(L)}(j = 0, \ldots, L)$.

3.2 MR part

In MR part, we use output of GBiCG part to update residual vector, multiple auxiliary vectors and solution vector. First, we choose parameter $\gamma_i^{(m+1)}(i = 1, 2, \ldots, L)$ in the L-degree MR polynomial $\boldsymbol{p}_{m+1(t)} = 1 - \sum_{i=2}^{L}\gamma_i^{(m+1)}$ such that the norm of updating residual $\boldsymbol{r}_{k+L}$ is minimum. Next, from definition of $Q_{k+L}(t)$, $Q_{k+L-1}(t) = p_t(k)p_t(k-L)\cdots p_1(t) = p_t(k)Q_k(t)$, we update $\boldsymbol{r}_{k+L}, U_{k+L}$and $\boldsymbol{x}_{\mathrm{k+L}}$ as follows:

$$Q_{k+L}\boldsymbol{r}_{k+L}^{\mathrm{GB}} = Q_k\boldsymbol{r}_{k+L}^{\mathrm{GB}} - \sum_{i=1}^{L}\gamma_i^{(m+1)}A^iQ_k\boldsymbol{r}_{k+L}^{\mathrm{GB}}, \quad (24)$$

$$Q_{k+L}U_{k+L-1}^{\mathrm{GB}} = Q_k\boldsymbol{U}_{k+L-1}^{\mathrm{GB}} - \sum_{i=1}^{L}\gamma_i^{(m+1)}A^iQ_kU_{k+L-1}^{\mathrm{GB}}. \quad (25)$$

From eqns.(23) and (24), $\boldsymbol{x}_{k+L}$ is updated as

$$\boldsymbol{x}_{k+L} = \hat{\boldsymbol{x}}_k^{(L)} - \sum_{i=1}^{L}\gamma_i^{(m+1)}A^{i-1}Q_k\boldsymbol{r}_{k+L}^{\mathrm{GB}}. \quad (26)$$

We present an algorithm of BiCGSafe2 method as below.

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithm of GBiCGSTAB(s, L) method

Let $\boldsymbol{x}_0$ be an initial guess, and put $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
choose $N \times s$ matrices P_0
Set $U_0 = [\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{s-1}\boldsymbol{r}_0]$
Set $U_1 = AU_0, M = P^T U_1, \boldsymbol{m} = P^T \boldsymbol{r}_0$
Solve $M\boldsymbol{\gamma} = \boldsymbol{m}$ for $\boldsymbol{\gamma}$ (27)
$\boldsymbol{r}_0 = \boldsymbol{r}_0 - U_1\boldsymbol{\gamma}, \boldsymbol{x}_0 = \boldsymbol{x}_0 + U_0\boldsymbol{\gamma}$ (28)
$\boldsymbol{r}_1 = A\boldsymbol{r}_0$, iter $= 0, \omega = -1$ (29)
While $||\boldsymbol{r}_n||_2/||\boldsymbol{r}_0||_2 > \epsilon$ Do
  $M = -\omega M$ (30)
  For $i = 0, 1, \ldots, L-1$ Do
    If iter $= 0$ and $i = 0$ then $i = 1$
    $\boldsymbol{m} = P^T \boldsymbol{r}_i$ (31)
    $\boldsymbol{x}_0 = \boldsymbol{x}_{iter}, \boldsymbol{r}_0 = \boldsymbol{r}_{iter}$ (32)
    For $j = 1, 2, \ldots,$ s Do
      If $j = 1$ then
        Solve $M\boldsymbol{\gamma} = \boldsymbol{m}$ for $\boldsymbol{\gamma}$ (33)
        $U_k \boldsymbol{e}_j = \boldsymbol{r}_k - \sum_{q=1}^{s} U_k \boldsymbol{e}_q \gamma(q)$ $(k = 0, 1, \ldots, i)$ (34)
      Else
        Solve $[\boldsymbol{m}, M\boldsymbol{e}_1, \ldots, M\boldsymbol{e}_{j-2}, M\boldsymbol{e}_j, \ldots, M\boldsymbol{e}_s]\boldsymbol{\gamma}$
        $= M\boldsymbol{e}_{j-1}$ for $\boldsymbol{\gamma}$ (35)
        $U_k \boldsymbol{e}_j = U_{k+1}\boldsymbol{e}_{j-1} - \boldsymbol{r}_k \gamma(1) - \sum_{q=1}^{j-2} U_{k+1}\boldsymbol{e}_q \gamma(q+1)$
        $- \sum_{q=j}^{s} U_k \boldsymbol{e}_q \gamma(q)$ $(k = 0, 1, \ldots, i)$ (36)
      End If
      Compute $U_{i+1}\boldsymbol{e}_j = AU_i \boldsymbol{e}_j$ (37)
      $M\boldsymbol{e}_j = P^T U_{i+1}\boldsymbol{e}_j$ (38)
    End Do
    Solve $M\boldsymbol{\gamma} = \boldsymbol{m}$ for $\boldsymbol{\gamma}$ (39)
    $\boldsymbol{r}_k = \boldsymbol{r}_k - U_{k+1}\boldsymbol{\gamma}$ $(k = 0, 1, \ldots, i)$ (40)
    $\boldsymbol{x}_0 = \boldsymbol{x}_0 + U_0\boldsymbol{\gamma}$ (41)
    $\boldsymbol{r}_{i+1} = A\boldsymbol{r}_i$ (42)
  End Do
  For $j = 1, 2, \ldots, L$ Do
    $\tau_{ij} = \frac{1}{\sigma_i}(\boldsymbol{r}_i, \boldsymbol{r}_j), \boldsymbol{r}_j = \tau_{ij}\boldsymbol{r}_i$ $(i = 1, 2, \ldots, j-1)$ (43)
    $\sigma_j = (\boldsymbol{r}_j, \boldsymbol{r}_j), \gamma_j^{'} = \frac{1}{\sigma_j}(\boldsymbol{r}_j, \boldsymbol{r}_0)$ (44)
  End Do
  $\gamma_L = \gamma_L^{'}, \omega = \gamma_L$ (45)
  $\gamma_j = \gamma_j^{'} - \sum_{i=j+1}^{L} \tau_{ji}\gamma_i$ $(j = L-1, L-2, \ldots, 1)$ (46)
  $\gamma_j^{''} = \gamma_{j+1} + \sum_{i=j+1}^{L-1} \tau_{ji}\gamma_{i+1}$ $(j = 1, 2, \ldots, L-1)$ (47)
  $\boldsymbol{x}_0 = \boldsymbol{x}_0 + \gamma_1 \boldsymbol{r}_0, \boldsymbol{r}_0 = \boldsymbol{r}_0 - \gamma_L^{'}\boldsymbol{r}_L, U_0 = U_0 - \gamma_L U_L$ (48)
  $U_0 = U_0 - \gamma_j U_j$ $(j = 1, 2, \ldots, L-1)$ (49)
  $\boldsymbol{x}_0 = \boldsymbol{x}_0 + \gamma_j^{''}\boldsymbol{r}_j, \boldsymbol{r}_0 = \boldsymbol{r}_0 - \gamma_j^{'}\boldsymbol{r}_j$ $(j = 1, 2, \ldots, L-1)$ (50)
  iter $=$ iter $+ (s+1)L$ (51)
  $\boldsymbol{x}_{iter} = \boldsymbol{x}_0, \boldsymbol{r}_{iter} = \boldsymbol{r}_0$ (52)
End While

4 Estimation of computational cost

Table 1 presents computational cost per one iteration of representative five kinds of methods. In Table 1, "$A\boldsymbol{u}$" means number of matrix-vector multiplications. "$\boldsymbol{u}^T\boldsymbol{v}$" means also number of inner products. Similarly "$\boldsymbol{u} \pm \boldsymbol{v}$" means number of an addition of two vectors. "$\alpha\boldsymbol{u}$, $\boldsymbol{u}/\alpha$" means number of a scalar multiplication of a vector. "N" denotes dimension of matrix, and "NNZ" denotes number of nonzero entries of matrix. Computational cost per $(s+1)$ iterations of GBiCGSTAB(s, L) method is shown in Table 1.

5 Numerical experiments

5.1 Computational environment and conditions

All computations were done in double precision floating point arithmetics of Fortran90, and performed on Dell PowerEdge R210 II with CPU of Intel Xeon E3-1220, clock of 3.1GHz, main memory of 8GB and OS of Scientific Linux 6.0. Optimum option "-O3" was used. Stopping criterion of iterative methods is less than 10^{-7} of the relative residual 2-norm $||\boldsymbol{r}_{k+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$. In all cases the iteration was started with the initial guess solution $\boldsymbol{x}_0 = (0, 0, \ldots, 0)^T$. We examined performance of ierative methods for two matrices which stem from the field of electromagnetic analysis. The iterative methods shown in bold means the hybrid method which combine between two-term and three-term recurrences. In addition, we examined performance based IDR(s) method, i.e., IDR(s), BiIDR(s) and GBiCGSTAB(s, L) methods.

1. GPBiCG, BiCGSTAB, **BiCGSTAB2**
2. GPBiCG_v1, **GPBiCG2_v1**, GPBiCG_v2, **GPBiCG2_v2**
3. GPBiCG_AR, **GPBiCG_AR2**
4. BiCGSafe, **BiCGSafe2**
5. IDR(s), BiIDR(s), GBiCGSTAB(s, L)

All test matrices were normalized with diagonal scaling. Maximum iteration was fixed as 10000. ILU(0) preconditioning without extra fill-ins are applied to all iterative methods. Acceleration parameter γ for diagonal entries varied from 1.05 until 1.30 at the interval 0.05. The parameter s of IDR(s), BiIDR(s) and GBiCGSTAB(s, L) methods varied as 1, 2, 4 and 8. In a same way, the parameter L of GBiCGSTAB(s, L) method varied as 2 and 3.

Table 2 presents specifications of test matrices stemed from FEM analysis for electromagnetic problems done

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 1: Computational cost per one iteration of five kinds of methods.

operation	GPBiCG	BiCGSTAB2		BiCGSafe	BiCGSafe2		GBiCGSTAB$(s, L)^\star$
		even	odd		even	odd	
$A\boldsymbol{u}$ $(\times 2NNZ)$	2	2	2	2	2	2	$(s+1)$
$\boldsymbol{u}^T\boldsymbol{v}$ $(\times 2N)$	7	7	4	7	7	4	(s^2+s+3)
$\boldsymbol{u} \pm \boldsymbol{v}$ $(\times N)$	16	16	11	14	14	11	$\frac{1}{2}(s^2L+sL+s^2+5s+L+3)$
$\alpha\boldsymbol{u}$, $\boldsymbol{u}/\alpha$ $(\times N)$	13	13	9	13	13	9	$\frac{1}{2}(s^2L+sL+s^2+5s+L+3)$

Remark: The mark "$\star$" shows computational cost per $(s+1)$ iterations.

Table 2: Specifications of test matrices.

matrix	N	NNZ	ave. NNZ
boxshield_20	881,080	30,716,540	36.96
IPMSM_120	3,628,380	113,904,598	31.39

by profs. K. Fujiwara and Y. Takahashi of Doshisha University [11]. In Table 2, "N" means number of dimensions, "NNZ" means number of nonzero entries, and "ave. NNZ" means average number of nonzero entries per one row of matrix.

5.2 Numerical results

Table 3 shows convergence of preconditioned GBiCGSTAB(s, L) method for matrix boxshield_20 when parameter γ is fixed as 1.15. Table 4 presents also convergence of preconditioned GBiCGSTAB(s, L) method for matrix IPMSM_120 when parameter γ is fixed as 1.20. Table 5 exhibits performance of 14 kinds of preconditioned iterative methods for matrix boxshield_20. Similarly, Table 6 shows performance of 14 kinds of preconditioned iterative methods for matrix IPMSM_120. In Table 3-6, "itr." means number of iterations, and "TRR" means values of True Relative Residual of $||\boldsymbol{b} - A\boldsymbol{x}_{k+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$ for the converged solutions $\boldsymbol{x}_{k+1}$, and "ratio" means ratio of CPU time of GPBiCG method to that of other iterative methods. From Table 3, as parameters s and L increase, a number of iterations decreases except for $s = 2$ for matrix boxshield_20.

Table 3: Convergence of preconditioned GBiCGSTAB(s, L) method for matrix boxshield_20 when parameter γ is fixed as 1.15.

L	s	itr.	total time [sec.]	ave. time [msec.]	$\log_{10}$ (TRR)
2	1	1172	116.51	96.49	-7.06
	2	1188	120.29	98.38	-7.10
	4	1070	113.26	102.65	-7.23
	8	1062	122.17	111.82	-7.13
3	1	1146	115.08	97.43	-7.02
	2	1179	120.84	99.58	-7.17
	4	1035	**111.49**	104.41	-7.04
	8	1026	122.75	116.29	-7.03

Table 4: Convergence of preconditioned GBiCGSTAB(s, L) method for matrix IPMSM_120 when parameter γ is fixed as 1.20.

L	s	itr.	total time [sec.]	ave. time [msec.]	$\log_{10}$ (TRR)
2	1	1644	674.88	393.49	-7.09
	2	1218	517.57	401.92	-7.02
	4	1060	**475.99**	422.64	-7.15
	8	990	486.63	463.25	-7.08
3	1	1716	710.82	397.88	-7.08
	2	1224	527.32	407.87	-7.09
	4	1065	487.46	431.36	-7.26
	8	999	510.89	483.41	-7.26

From Table 5, we can gain the following observation.

- GPBiCGSTAB(s, L) method shows the least CPU time, and BiCGSafe method shows the second.
- BiCGSTAB2, BiCGSafe2 and GPBiCG_AR2 methods present lower iterations and faster CPU time compared with each original iterative methods.

From Table 6, we get the following observation.

- GPBiCGSTAB(s, L) method shows the least time, and BiIDR(s) method shows the second of that.
- A family of IDR(s) methods present an excellent convergence rate.

Table 7 demonstrates the ranking of convergence time and overall for 14 kinds of ILU(0) preconditioned iterative methods. Fig. 1 exhibit history of relative residual 2-norm of six kinds of iterative methods.

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 5: Summary of performance of 14 kinds of preconditioned iterative methods for matrix boxshield_20.

method	γ	s	L	itr	pre. time[s]	itr. time[s]	total time[s]	ave. time[ms]	$\log_{10}$ (TRR)	ratio	ranking
GPBiCG	1.15	-	-	624	3.43	128.08	131.51	205.26	-7.13	1.00	12
BiCGSTAB	1.05	-	-	718	3.42	140.94	144.36	196.30	-7.04	1.10	14
BiCGSTAB2	1.10	-	-	592	3.43	120.68	124.11	203.85	-7.19	0.94	6
GPBiCG_v1	1.10	-	-	581	3.43	122.81	126.24	211.38	-7.34	0.96	8
GPBiCG2_v1	1.10	-	-	593	3.44	123.38	126.82	208.06	-7.14	0.96	9
GPBiCG_v2	1.10	-	-	578	3.43	121.35	124.78	209.95	-7.16	0.95	7
GPBiCG2_v2	1.15	-	-	612	3.43	127.15	130.59	207.76	-7.15	0.99	11
GPBiCG_AR	1.10	-	-	579	3.42	118.23	121.65	204.20	-7.20	0.93	4
GPBiCG_AR2	1.10	-	-	577	3.42	116.94	120.36	202.67	-7.05	0.92	**3**
BiCGSafe	1.10	-	-	582	3.44	119.79	123.23	205.82	-7.27	0.94	5
BiCGSafe2	1.10	-	-	575	3.43	116.90	120.33	203.30	-7.07	0.91	**2**
IDR(s)	1.05	8	-	1010	3.43	134.47	137.90	133.14	-7.05	1.05	13
BiIDR(s)	1.05	4	-	1124	3.44	123.56	126.99	109.93	-7.05	0.97	10
GBiCGSTAB(s, L)	1.15	4	3	1035	3.43	108.06	**111.49**	104.41	-7.03	0.86	**1**

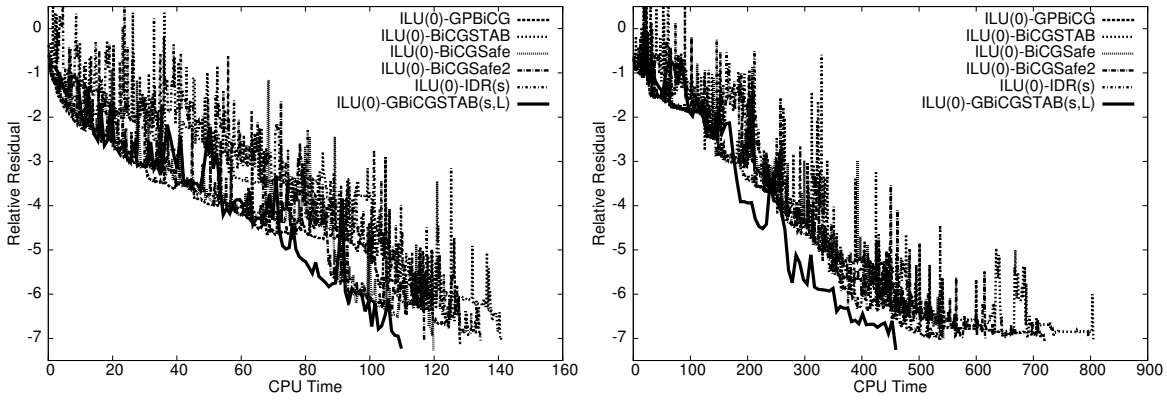
Table 6: Summary of performance of 14 kinds of preconditioned iterative methods for matrix IPMSM_120.

method	γ	s	L	itr	pre. time[s]	itr. time[s]	total time[s]	ave. time[ms]	$\log_{10}$ (TRR)	ratio	ranking
GPBiCG	1.15	-	-	859	28.03	719.64	747.67	837.76	-7.13	1.00	12
BiCGSTAB	1.10	-	-	1010	28.00	804.74	832.74	796.77	-7.04	1.11	13
BiCGSTAB2	1.20	-	-	818	28.04	680.47	708.51	831.87	-7.19	0.95	10
GPBiCG_v1	1.15	-	-	720	28.02	619.46	647.48	860.36	-7.34	0.87	7
GPBiCG2_v1	1.25	-	-	962	28.00	815.82	843.83	848.05	-7.14	1.13	14
GPBiCG_v2	1.10	-	-	779	27.99	666.82	694.81	855.99	-7.16	0.93	9
GPBiCG2_v2	1.15	-	-	823	28.01	696.89	724.90	846.77	-7.15	0.97	11
GPBiCG_AR	1.10	-	-	683	28.04	571.37	599.40	836.56	-7.20	0.80	5
GPBiCG_AR2	1.20	-	-	800	28.02	664.54	692.56	830.68	-7.05	0.93	8
BiCGSafe	1.15	-	-	654	28.03	543.53	571.56	831.09	-7.27	0.76	4
BiCGSafe2	1.10	-	-	744	28.04	616.22	644.26	828.25	-7.07	0.86	6
IDR(s)	1.30	4	-	1042	28.06	541.57	569.64	519.74	-7.05	0.76	**3**
BiIDR(s)	1.30	4	-	1041	28.02	473.27	501.30	454.63	-7.05	0.67	**2**
GBiCGSTAB(s, L)	1.20	4	2	1060	27.99	448.00	**475.99**	422.64	-7.08	0.64	**1**

Table 7: Ranking of CPU time and overall for 14 kinds of preconditioned iterative methods.

method	ranking		total score	overall ranking
	boxshield_20	IPMSM_120		
GPBiCG	12	12	24	13
BiCGSTAB	14	13	27	14
BiCGSTAB2	6	10	16	8
GPBiCG_v1	8	7	15	7
GPBiCG2_v1	9	14	23	12
GPBiCG_v2	7	9	16	8
GPBiCG2_v2	11	11	22	11
GPBiCG_AR	4	5	9	**3**
GPBiCG_AR2	3	8	11	5
BiCGSafe	5	4	9	**3**
BiCGSafe2	2	6	8	**2**
IDR(s)	13	3	16	8
BiIDR(s)	10	2	12	6
GBiCGSTAB(s, L)	1	1	2	**1**

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(i)boxshield_20 (ii)IPMSM_120

Figure 1: History of relative residual 2-norm six kinds of methods.

6 Concluding Remarks

In this paper, we examined convergence rate of several product-type iterative methods and a family of and IDR(s) based on iterative methods for two matrices stemed from FEM analysis for electromagnetic problems. Through numerical experiment, it turned out that **GBiCGSTAB(s, L)** and **BiCGSafe2** methods have effectiveness and robustness compared with the other iterative methods. We will analyse effect of SSOR preconditioning with Eisenstat trick. We would like to consider mathematical theory and aspect as a future work.

Acknowledgments

We show sincere thanks to profs. K. Fujiwara and Y. Takahashi of Doshisha University for giving interesting two matrices stemed from FEM analysis for realistic problems. We appreciate an anonymous referee for giving nice advise (see ref.[1]).

References

[1] K. Abe, G.L.G. Sleijpen, Solving linear equations with a STABilized GPBiCG method, Proc. of 14th Kansetouchi workshop of JSIAM, pp.29-34, Okayama University of Science, January, 2011.(in print)

[2] S. Fujino, M. Fujiwara and M. Yoshida, BiCGSafe method based on minimization of associate residual, JSCES No.20050028, 2008.

[3] M. H. Gutknecht, Variants of BiCGStab for matrices with complex spectrum, SIAM J. CSI. Comput, Vol.14, pp.1020-1033, 1993.

[4] Moethuthu, S. Fujino, Stability of GPBiCG_AR method based on minimization of associate residual, Journal of ASCM, Vol.5081, pp.108-120, 2008.

[5] Moethuthu, S. Fujino, A variant of GPBiCG_AR method with reduction of computational costs, Asian Technology Conference in Mathematics (ATCM), pp.419-428, Thailands, December, 2008.

[6] M. Tanio, M. Sugihara, GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials, J. of Computational and Applied Mathematics, Vol.235, pp.765-784, 2010.

[7] M.B. van Gijzen, P. Sonne veld, An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties, TR 08-21, Delft Univ. of Tech., 2008.

[8] P. Sonneveld, M.B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM Journal on Scientific Computing, Vol.31, pp.1035-1062, 2008.

[9] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., pp.631-644, 1992.

[10] S.-L. Zhang, GPBi-CG: Generalized product-type preconditionings based on Bi-CG for solving nonsymmetric linear systems, SIAM J. Sci. Comput., pp.537-551, 1997.

[11] Y. Takahashi *et al.*, Computation of electromagnetic problems by means of paralleled time-dependent FEM analysis, Proc. of 14th Kansetouchi workshop of JSIAM, pp.207-212, Okayama University of Science, January, 2011. (in Japanese)

ISBN: 978-988-19251-9-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)