

A block sparse approximate inverse with cutoff preconditioner for semi-sparse linear systems derived from Molecular Orbital calculations

Ikuro Yamazaki¹, Masayuki Okada², Hiroto Tadano¹, Tetsuya Sakurai¹ and Keita Teranishi³

¹ Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8578, Japan

² Hitachi, Ltd., Software Division, Kaneichi Bldg. 549-8, Shinano-cho, Totsuka-ku, Yokohama 244-0801, Japan

³ Cray, Inc., 380 Jackson St. Suite 210, St Paul, MN 55101, USA

E-mail yamazaki@mma.cs.tsukuba.ac.jp

Received September 30, 2009, Accepted December 26, 2009

Abstract

We present an approach to preconditioning for large, relatively dense linear systems and verify the validity of our method. We restrict the target of our method to Molecular Orbital (MO) calculations. Sparse Approximate Inverse (SAI) is typically less effective at accelerating the convergence and requires a huge computational cost in its construction when a large number of nonzero entries are kept in the approximate inverse matrix. We explain a construction of Block SAI and a cutoff strategy to reduce the number of nonzero elements, and investigate the efficiency of a cutoff strategy and Block SAI.

Keywords linear system, preconditioning, sparse approximate inverse

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

Density Function Theory (DFT) is a popular method to obtain the potential energy of materials in atomistic scales, and typically involves a solution of large scale eigenvalue problems. In particular for the area of protein folding, we have developed a task-parallel scheme for the eigenvalue problems [1] that achieves a substantial parallel speedup. Yet, our scheme requires each CPU (or a group of CPUs) to solve the solutions of large linear systems

$$Ax = b,$$

where $A \in \mathbb{C}^{n \times n}$ contains relatively large number of nonzero elements (semi-sparse). We have observed that these linear systems are typically derived from Molecular Orbital (MO) calculations.

For the solution of such linear systems on multi-core CPUs, Krylov subspace methods preconditioned with sparse approximate inverse (SAI) based on Frobenius norm minimization appear very attractive because of the good parallel efficiency in both preconditioner construction and application. However, the convergence of the iterative solvers with SAI tends to be slower than those with the conventional preconditioning methods such as incomplete factors. This is because the SAI preconditioner often falls into local minimization with respect to individual columns. In addition to this drawback, the arithmetic costs of constructing the SAI preconditioner grows cubically with the number of nonzero entries per row, making it less feasible than other preconditioning alternatives.

In this paper, we attempt to overcome these performance bottlenecks of SAI using a blocked version of Frobenius norm minimization in order to mitigate the side effect of the minimization process applied to individual columns of the approximate inverse. We also apply different drop-threshold schemes to achieve a reduction of the arithmetic costs of preconditioner construction and application at the cost of a small increase in the iteration counts.

This paper is organized as following. In Section 2, the SAI preconditioner and its block variant are described. We discuss our preconditioner for the semi-sparse linear systems in Section 3. In Section 4, we describe how to profile for a cutoff parameter. In Section 5, we investigate the performance of our preconditioner through the numerical experiments with two matrices obtained from the computation of molecular orbital, followed by the concluding remarks in Section 6.

2. SAI and its block variant

Our approach to constructing a preconditioning matrix is based on Frobenius norm minimization [2, 3]:

$$\min_M \|AM - I\|_F^2, \quad (1)$$

where I is the identity matrix. The Frobenius norm can be minimized in parallel:

$$\|AM - I\|_F^2 = \sum_{k=1}^n \|Am_k - e_k\|_2^2, \quad (2)$$

where m_k and e_k are the k -th column of M and I respectively. Thus, the preconditioning matrix $M =$

$[\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n] \approx A^{-1}$ is constructed by solving n independent least square problems:

$$\min_{\mathbf{m}_k} \|\mathbf{A}\mathbf{m}_k - \mathbf{e}_k\|_2^2, \quad k = 1, 2, \dots, n. \quad (3)$$

The Block SAI preconditioner is proposed by Barnard and Grote [4, 5] in order to improve the accuracy of the preconditioning matrix. In the Block SAI preconditioner, problem (1) can be approximated in parallel:

$$\|AM - I\|_F^2 = \sum_{k=1}^L \|AM_k - E_k\|_F^2, \quad (4)$$

where l is a block size, $L = \lceil n/l \rceil$ and E_k is a sub-matrix of the identity matrix I such that $I = [E_1, E_2, \dots, E_L]$. Thus, the preconditioning matrix $M = [M_1, M_2, \dots, M_L]$ is constructed by solving L independent least square problems:

$$\min_{M_k} \|AM_k - E_k\|_F^2, \quad k = 1, 2, \dots, L. \quad (5)$$

The initial sparsity pattern M_0 of the preconditioning matrix is decided by the following:

$$\text{spy}(A) = \text{spy}(M_0), \quad (6)$$

where “spy” denotes the sparsity pattern of a matrix.

3. Block SAI with cutoff (BSAIC)

Molecular orbital calculation of the biochemistry application [6] requires to solve the following generalized eigenvalue problem:

$$F\mathbf{v} = \lambda S\mathbf{v},$$

where $F \in \mathbb{R}^{n \times n}$ is symmetric and $S \in \mathbb{R}^{n \times n}$ is symmetric positive definite. In molecular orbital calculation, eigenpairs around Highest Occupied Molecular Orbital (HOMO) - Lowest Unoccupied Molecular Orbital (LUMO) are important to analyze the chemical reactions. Thus, interior eigenvalue problems are required to solve. For instance, the inverse of the shifted matrix $(F - \sigma S)^{-1}$ is required to find some eigenvalues around σ in some methods. In the Sakurai-Sugiura (SS) method [1], the solution of the system of linear equations $A\mathbf{x} = \mathbf{b}$, where the coefficient matrix A is given by

$$A = \omega S - F,$$

is required. Since F is semi-sparse, A is also semi-sparse.

Firstly, we describe why the sparsity pattern of A is used as the initial sparsity pattern of M_0 . The nonzero pattern of A^{-1} is changed depending on ω . When the sparsity pattern of A is similar to that of A^{-1} , $A\mathbf{x} = \mathbf{b}$ is relatively easy to solve. Meanwhile, when the sparsity pattern of A is not similar to that of A^{-1} , it is difficult to solve $A\mathbf{x} = \mathbf{b}$. In practice, the sparsity pattern of A^{-1} with important ω is relatively similar to that of A , and thus the sparsity pattern of A can be used as the initial sparsity pattern of M_0 .

Secondly, we describe how to reduce the computational cost of SAI. The coefficient matrix A contains relatively large number of nonzero elements, and thus the computational cost of SAI is huge if SAI is applied to these matrices. For this reason, we propose a cutoff

that is applied to the coefficient matrix A to reduce the number of nonzero elements. In our cutoff, off-diagonal nonzero elements of A are dropped if they are small compared to a cutoff parameter θ as shown below:

$$A_c = [\tilde{a}_{ij}], \quad \tilde{a}_{ij} = \begin{cases} a_{ij}, & (|a_{ij}| > \theta \text{ or } i = j), \\ 0, & (\text{otherwise}), \end{cases} \quad (7)$$

where θ is a nonnegative real value. As a result, the computational cost of SAI is reduced because of fewer nonzero elements in A . However, a larger value of θ leads to a less effective preconditioning matrix with a large number of iterations, though it reduces the computational costs of SAI even further.

After applying the cutoff strategy, least square problems with the approximate matrix A_c :

$$\min_{M_k} \|A_c M_k - E_k\|_F^2, \quad k = 1, 2, \dots, L \quad (8)$$

are solved. The matrix $M = [M_1, M_2, \dots, M_L]$ is employed as the preconditioning matrix. We call this preconditioner the Block SAI with Cutoff (BSAIC) preconditioner.

We describe the performance improvement of the approximate inverse obtained by a block version of Frobenius norm minimization. Firstly, the performance degradation of SAI by applying our cutoff can be reduced due to extra fill-ins introduced by the blocked version. These fill-ins make the preconditioner more robust and allow larger value of θ for the cutoff to make it sparser than the original SAI.

Let the size of the matrix of i -th QR decomposition of SAI be $m_i \times n_i$ ($m_i \geq n_i$). Its computational cost is $O(m_i n_i^2)$. Let the number of nonzero elements of A be αn^2 ($0 < \alpha \leq 1$), where n is the dimension of A . In many cases, m_i and n_i are proportional to α . Therefore, the decrease of α provides the drastic decrease of the computational cost of the QR decomposition. Indeed, when the matrices derived from the computation of the molecular orbitals are used, m_i and n_i are proportional to α , and the cutoff is effective for the QR decomposition. Applying the cutoff to A , the number of iterations increases slowly in certain range of θ .

Secondly, Block SAI increases a scope of minimization as the original version does it for each column of M . Computing the minimum associated with a block of column reduces the threat of local minimization. In other words, a large l makes M more global minimizer in the sense of the linear space of A . However, a large l often increases the cost of the least square problem associated with each block as the row dimension of the matrix for each least square problem is determined by the number of nonzero rows in the block. This performance drawback can be mitigated with a large cutoff value θ , and our experiments in Section 5 indicates that a large l slows down the increase of the iteration count with respect to θ .

In conclusion, both the cutoff parameter θ and the block size l are preferred to be large as much as possible. However, these values depend on problems, and thus we take matrices obtained from computation of the molecular orbitals for instance.

4. Profiling for cutoff parameter tuning

In the SS method, linear systems $Ax = b$ for various ω which are around HOMO-LUMO need to be solved. The difficulty of solving $Ax = b$ depends on ω . The constructing cost of M are highly dependent on the number of nonzero elements of A . Therefore, we need to set an appropriate cutoff parameter depending on ω . How to profile for a cutoff parameter is shown below:

- 1) Profiling stage
 - a. Check the values of matrix elements.
 - b. Check the constructing time of M with a large cutoff value.
 - c. Estimate the constructing times of M with several cutoff values from the number of nonzero elements of all matrices of QR decomposition.
 - d. Set the range of the cutoff parameter μ with $\mu_{\min} \leq \mu \leq \mu_{\max}$.
- 2) Trial stage
 - a. Set $\mu = \mu_{\max}$.
 - b. Construct M with the cutoff parameter μ .
 - c. Perform one cycle of the preconditioned GMRES(k).
 - d. If $\|b - Ax_k\|/\|b\| \leq \delta$ or $\mu \leq \mu_{\min}$ then exit this trial stage.
Else
set $\mu = \beta\mu$ ($0 < \beta < 1$).
End if
 - e. Go to 2)-b.
- 3) Perform the preconditioned GMRES(k) using M constructed in 2).

5. Numerical experiments

In this section, the BSAIC preconditioner is compared with SAI and Block SAI by numerical experiments. All experiments are carried out by MATLAB 7.4 on MacBook (CPU: Intel Core 2 Duo 2.0GHz, Memory: 2.0Gbytes, OS: Mac OS 10.5.6). The test problems are solved by the preconditioned GMRES(30) method [7]. The stopping criterion for the relative residual is 10^{-10} . The initial guess x_0 is set to $\mathbf{0}$ and all elements of b are set to 1.

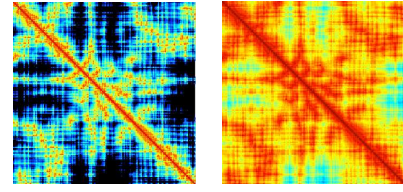
5.1 Example 1

In Example 1, the matrices F and S are derived from the computation of the molecular orbitals of a model DNA. The coefficient matrix A is given by $\omega S - F$, where ω is a real parameter.

The size of A is 1,980 and the number of nonzero elements is 728,080 (18.57%). In this example, the parameter ω for the coefficient matrix A and the block size l are set to -0.16 and 30, respectively.

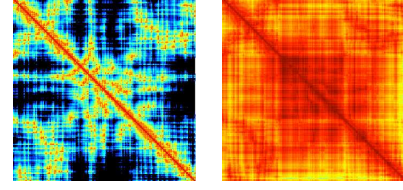
When A_c^{-1} with small θ is used as a preconditioning matrix, the number of iterations is very small (e.g., the number of iterations is 5 with $\theta = 10^{-3}$). The eigenvalue corresponding to HOMO is -0.16538 . In molecular orbital calculations, eigenpairs around HOMO-LUMO are desired. Therefore, the system of linear equations $Ax = b$ for several parameters ω which are close to HOMO need to be solved.

The results of Example 1 are reported in Table 1. The number of iterations of SAI is 9 and that of Block SAI



(a) $\text{spy}(A)$. (b) $\text{spy}(A^{-1})$.

Fig. 1. The sparsity pattern with $\omega = -0.24$.



(a) $\text{spy}(A)$. (b) $\text{spy}(A^{-1})$.

Fig. 2. The sparsity pattern with $\omega = -0.5$.

with $l = 30$ is 7. Block SAI takes less iterations and preconditioning time than those of SAI. SAI with Cutoff takes less preconditioning time than that of SAI and Block SAI. BSAIC converges faster than other preconditionings.

5.2 Example 2

In Example 2, the matrices F and S are derived from the computation of the molecular orbitals of Lysozyme. In this example, we also use the real parameter ω for the coefficient matrix $A = \omega S - F$. The matrices F and S are real symmetric and real symmetric positive definite, respectively. The size of A is 6,005 and the number of nonzero elements is 3,275,925 (9.08%). The parameter ω for coefficient matrix A and the block size l are set to -0.24 and 30, respectively.

Fig. 1 shows sparsity patterns of A and A^{-1} with $\omega = -0.24$, respectively. Fig. 2 shows sparsity patterns of A and A^{-1} with $\omega = -0.5$, respectively. The eigenvalue which corresponds to HOMO is -0.25086 .

The results of Example 2 are reported in Table 2. The number of iterations of SAI is 6 and that of Block SAI with $l = 30$ is 5. BSAIC converges faster than other preconditionings.

Fig. 3 shows the actual preconditioning time and the estimated time with respect to several cutoff values using $l = 30$ for Example 2. One iteration take 0.2 [sec]. When $\theta = 10^{-5}$ is used, too large comparison with time for one iteration. Thus, we set the range of cutoff parameter from 10^{-1} to 10^{-4} .

The results of our profiling are presented in Table 3. The threshold δ of relative residual are set to 10^{-3} . Total time denotes the times including profiling, Cutoff, preconditioning and iteration. An appropriate cutoff parameter depending on ω is found. When $\theta = 10^{-5}$ is used, the preconditioning time is 140.92 [sec] without profiling. Thus, when a small value of θ is used without profiling, large computational cost is required.

Fig. 4 shows the relative residual of our profiling with $\omega = -0.45$ for Example 2. $Ax = b$ is solved with variable θ .

Table 1. Results for Example 1.

Preconditioner	The number of iterations	Wall clock time [sec]			
		Cutoff	Preconditioning	Iteration	Total
SAI	9	—	474.94	0.24	475.18
Block SAI	7	—	56.85	0.24	57.09
SAI with Cutoff	$\theta = 10^{-2}$	123	1.15	2.17	3.44
	$\theta = 10^{-4}$	21	16.82	0.50	17.48
BSAIC	$\theta = 10^{-2}$	44	1.07	0.77	1.96
	$\theta = 10^{-4}$	14	5.46	0.29	5.91

Table 2. Results for Example 2.

Preconditioner	The number of iterations	Wall clock time [sec]			
		Cutoff	Preconditioning	Iteration	Total
SAI	6	—	21197.78	0.80	21198.58
Block SAI	5	—	1760.92	0.64	1761.56
SAI with Cutoff	$\theta = 10^{-2}$	59	4.00	2.64	7.08
	$\theta = 10^{-4}$	13	75.62	0.45	76.65
BSAIC	$\theta = 10^{-2}$	17	3.91	0.58	4.93
	$\theta = 10^{-4}$	9	36.54	0.43	37.55

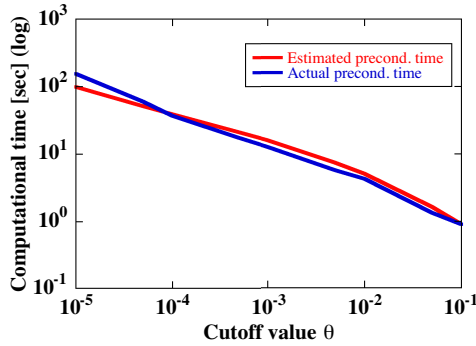
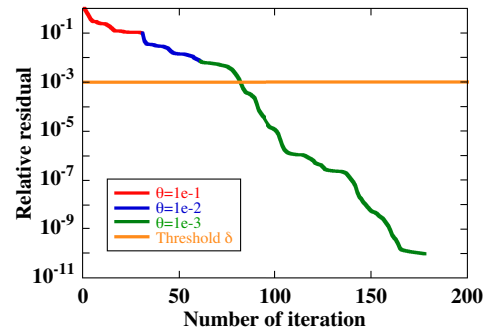
Fig. 3. The actual preconditioning time and the estimated time with respect to several Cutoff value using $l = 30$ for Example 2.

Table 3. Results of our profiling for Example 2.

ω	Appropriate θ	Total time [sec]	The total number of iteration
-0.24	10^{-2}	13.36	46
-0.3	10^{-2}	14.32	59
-0.35	10^{-2}	17.16	130
-0.4	10^{-2}	14.54	69
-0.45	10^{-3}	28.82	178
-0.5	10^{-4}	83.53	793

Fig. 4. The relative residual of our profiling with $\omega = -0.45$ for Example 2.

6. Conclusions

Our method, BSAIC algorithm, reduces the computational cost for generating the approximate inverse M , and overcomes the performance bottlenecks of SAI using the blocked version of Frobenius norm minimization and the cutoff strategy for semi-sparse matrices. Although our empirical study is confined to the problems from electronic calculations, we demonstrate that the performance of BSAIC substantially better than SAI preconditioning, and the application of cutoff parameters further increase the performance advantage of BSAIC, making it robust and efficient preconditioning together. The results of the empirical study also indicate that some possibilities of predicting the cost of constructing M and tuning θ in a small computational overhead.

In future works, we will try to find a better strategy for selecting an appropriate cutoff parameter tuning and apply for large scale problems.

Acknowledgments

This research was supported in part by a Grant-in-Aid for Scientific Research of Ministry of Education, Culture, Sports, Science and Technology, Japan, Grant number: 21246018 and 21105502.

References

- [1] T. Sakurai and H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, *J. Comput. Appl. Math.*, **159** (2003) 119–128.
- [2] M. W. Benson and P. O. Frederickson, Iterative solution of large sparse linear systems arising in certain multi dimensional approximation problems, *Utilitas Math.*, **22** (1982) 127–140.
- [3] E. Chow and Y. Saad, Approximate inverse preconditioners via sparse-sparse iterations, *SIAM J. Sci. Comput.*, **19** (1998) 995–1023.
- [4] S. T. Barnard and M. J. Grote, A block version of the SPAI preconditioner, in *Proc. of the 9th SIAM conf. on Parallel Process. for Sci. Comput*, San Antonio, TX, 1999.
- [5] SPAI, <http://www.computational.unibas.ch/software/spai/>.
- [6] Y. Inadomi, T. Nakano, K. Kitaura and U. Nagashima, Definition of molecular orbitals in fragment molecular orbital method, *Chem. Phys. Letters*, **364** (2002) 139–143.
- [7] Y. Saad and M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **7** (1986) 856–869.