JSIAM Letters Vol.2 (2010) pp.127–130 ©2010 Japan Society for Industrial and Applied Mathematics JSIAM Letters

Parallel stochastic estimation method of eigenvalue distribution

Yasunori Futamura[1], Hiroto Tadano[1] and Tetsuya Sakurai[1]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8573, Japan

E-mail *futamura@mma.cs.tsukuba.ac.jp*

Received July 29, 2010, Accepted October 22, 2010

Abstract

Some kinds of eigensolver for large sparse matrices require specification of parameters that are based on rough estimates of the desired eigenvalues. In this paper, we propose a stochastic estimation method of eigenvalue distribution using the combination of a stochastic estimator of the matrix trace and contour integrations. The proposed method can be easily parallelized and applied to matrices for which factorization is infeasible. Numerical experiments are executed to show that the method can perform rough estimates at a low computational cost.

Keywords eigenproblem, contour integration, stochastic estimation

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

1. Introduction

Interior eigenvalue problems arise in many kinds of scientific calculation, and they are the most time consuming part of these calculations. To solve these eigenvalue problems, the Arnoldi method with the shift invert technique [1], the Jacobi-Davidson method [1], and the Sakurai-Sugiura method [2,3] are reasonable choices. These methods require the specification of their parameters, such as shift points, the number of basis vectors, or closed curves on the complex plane. One could specify these parameters effectively if one had a rough estimation of the eigenvalue distribution. To estimate this distribution, some methods have been proposed, including the method using Sylvester's law of inertia and the algebraic substructure method [4]. Both methods require a matrix factorization, such as the LDLT factorization. However, it is not feasible to apply these method to large sparse matrices or matrices that are only referenced in the form of matrix-vector multiplications. In this paper, we propose a stochastic estimation method of the eigenvalue distribution that is based on a stochastic estimator of the matrix trace. We evaluate the performance of the proposed method by applying it to matrices from practical applications.

This paper is organized as follows. In Section 2, a stochastic estimator of an eigenvalue distribution and its parallelization are described. We show a simple implementation of our method in Section 3. In Section 4, we investigate the performance of our method through numerical experiments with four matrices from Matrix Market [5] and a matrix derived from a real-space density functional calculation. This is followed by the concluding remarks in Section 5.

2. A stochastic estimator of eigenvalue distribution

Let $A, B \in \mathbb{C}^{n\times n}$, $z \in \mathbb{C}$ be such that $(zB - A)$ is a regular matrix pencil. It is known that matrices A, B can be decomposed $A = URV^{\mathrm{H}}$, $B = UTV^{\mathrm{H}}$, where R, T are upper triangular matrices whose diagonal elements are r_{jj}, t_{jj}, respectively, and U, V are unitary matrices. Since

$$(zB - A)^{-1}B = V(zT - R)^{-1}TV^{\mathrm{H}},$$

and the matrix trace is similarity-invariant,

$$\begin{aligned}\mathrm{tr}((zB - A)^{-1}B) &= \mathrm{tr}((zT - R)^{-1}T) \\ &= \sum_{j=1}^{n} \frac{t_{jj}}{zt_{jj} - r_{jj}} \\ &= \sum_{j=1}^{n'} \frac{1}{z - \lambda_j}, \end{aligned} \tag{1}$$

where

$$t_{jj} \begin{cases} \neq 0 & (1 \leq j \leq n'), \\ = 0 & (n' + 1 \leq j \leq n), \end{cases}$$

and $\lambda_j = r_{jj}/t_{jj}$ $(j = 1, 2, \ldots, n')$ are finite eigenvalues of the matrix pencil (A, B).

When the contour integration

$$\begin{aligned}\mu &= \frac{1}{2\pi \mathrm{i}} \oint_\Gamma \mathrm{tr}((zB - A)^{-1}B)\mathrm{d}z \\ &= \frac{1}{2\pi \mathrm{i}} \oint_\Gamma \sum_{j=1}^{n'} \frac{1}{z - \lambda_j}\mathrm{d}z \end{aligned} \tag{2}$$

is performed, the eigenvalue count μ in a positively oriented Jordan curve Γ is derived by the residue theorem. To discretize (2), an N-point quadrature rule is applied and we approximate μ by

$$\mu \approx \hat{\mu} = \sum_{k=0}^{N-1} w_k \operatorname{tr}((z_k B - A)^{-1} B), \tag{3}$$

where z_j and w_j are a quadrature point and a weight, respectively. In the case of the trapezoidal rule on a circle with a center γ and a radius ρ, quadrature points and weights are defined by

$$z_k = \gamma + \rho e^{\frac{2\pi \mathrm{i}}{N}\left(k+\frac{1}{2}\right)}, \quad k = 0, 1, \ldots, N-1,$$

and

$$w_k = \frac{z_k - \gamma}{N}, \quad k = 0, 1, \ldots, N-1,$$

respectively, where i is the imaginary unit. According to [3], when the contour path is a circle, (3) is written as

$$\hat{\mu} = \sum_{j=1}^{n'} \frac{1}{1 + \left(\dfrac{\gamma - \lambda_j}{\rho}\right)^N}, \tag{4}$$

where $|(\gamma - \lambda_1)/\rho| \leq |(\gamma - \lambda_2)/\rho| \leq \cdots \leq |(\gamma - \lambda_{n'})/\rho|$. Let m' be an integer such that $\rho/\{1 + [(\gamma - \lambda_j)/\rho]^N\} = O(\varepsilon)$ for any j with $m' < j \leq n'$ for sufficiently small $\varepsilon > 0$. Then (4) can be expressed as

$$\hat{\mu} = \sum_{j=1}^{m'} \frac{1}{1 + \left(\dfrac{\gamma - \lambda_j}{\rho}\right)^N} + O(\varepsilon). \tag{5}$$

Thus, the eigenvalues that exist nearby and outside of Γ are attributed to quadrature error.

According to [6, 7], an unbiased estimation of the matrix trace is given by

$$\operatorname{tr}((z_k B - A)^{-1} B) \approx \frac{1}{s} \sum_{j=1}^{s} \boldsymbol{v}_j{}^{\mathrm{T}} (z_k B - A)^{-1} B \boldsymbol{v}_j, \tag{6}$$

where s is the number of sample vectors and $\boldsymbol{v}_j$ are vectors whose entries take 1 or -1 with equal probability. Using (6), one can estimate $\hat{\mu}$ as

$$\begin{aligned} \hat{\mu} &\approx \tilde{\mu} \\ &= \frac{1}{s} \sum_{k=0}^{N-1} w_k \sum_{j=1}^{s} [\boldsymbol{v}_j{}^{\mathrm{T}} (z_k B - A)^{-1} B \boldsymbol{v}_j]. \end{aligned} \tag{7}$$

Thus, the most time consuming part of the estimation of the trace of $(z_k B - A)^{-1} B$ is the solution of s independent linear systems

$$\begin{aligned} &(z_k B - A)\boldsymbol{x}_j^k = B\boldsymbol{v}_j, \\ &\quad j = 1, 2, \ldots, s, \quad k = 0, 1, \ldots, N-1. \end{aligned} \tag{8}$$

The subscript of $\boldsymbol{x}_j^k$ refers the sample vector $\boldsymbol{v}_j$ and the superscript refers the quadrature point z_k. If the matrices A and B are large sparse matrices or they are only referenced in the form of matrix-vector multiplications, an iterative method is a reasonable choice to solve these linear systems. Additionally, if B is the identity matrix I, the linear systems (8) are written as $(z_k I - A)\boldsymbol{x}_j^k = \boldsymbol{v}_j$. In this case, the shifted Krylov subspace method [8, 9] can be applied to solve simultaneously the linear systems $(z_k I - A)\boldsymbol{x}_j^k = \boldsymbol{v}_j$ for the scalar parameters z_k. By using the shifted Krylov subspace method, the total number of matrix-vector multiplications in each iteration is reduced to $1/N$ that of solving N systems separately by the normal Krylov subspace method. When A is a real symmetric matrix, $(z_k I - A)$ is a complex symmetric (but not Hermitian) matrix. The shifted conjugate orthogonal conjugate gradient (COCG) method [10, 11] is a reasonable choice to solve linear systems of complex symmetric matrices. Furthermore, our method does not require the solution vectors $\boldsymbol{x}_j^k$, but only the inner products $\boldsymbol{v}_j{}^{\mathrm{T}}\boldsymbol{x}_j^k$. In such a case, we can calculate these inner products by scalar recurrences (see [11]). Thus, memory allocation for the solution vectors and the auxiliary vectors of the shifted systems is not required.

```
Input : $A, B, \alpha, \beta, n_{\mathrm{c}}, N, s$
Output : $\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{n_{\mathrm{c}}}$
Set $\boldsymbol{v}_j$ whose elements take 1 or $-1$ with equal probability, for $j = 1, 2, \ldots, s$
$\rho = (\beta - \alpha)/2n_{\mathrm{c}}$
for $\ell = 1, 2, \ldots, n_{\mathrm{c}}$ do
    $\gamma_\ell = \alpha + (2\ell - 1)\rho$
    $z_{\ell k} = \gamma_\ell + \rho e^{(2\pi \mathrm{i}/N)(k+1/2)}$
    Solve $(z_{\ell k} B - A)\boldsymbol{x}_j^{\ell k} = B\boldsymbol{v}_j$ for $j = 1, 2, \ldots, s$, $k = 0, 1, \ldots, N-1$
    $\tilde{\mu}_\ell = [\rho/(sN)] \sum_{k=0}^{N-1} e^{(2\pi \mathrm{i}/N)(k+1/2)} \sum_{j=1}^{s} \boldsymbol{v}_j{}^{\mathrm{T}} \boldsymbol{x}_j^{\ell k}$
end for
```

Fig. 1. Algorithm.

A stochastic estimation method of the eigenvalue distribution is defined by the estimator of the eigenvalue count straightforwardly. Let Γ be a given Jordan curve, D the domain closed by Γ, and Γ_ℓ ($\ell = 1, 2, \ldots, n_{\mathrm{c}}$) a Jordan curve which closes sub-domain D_ℓ such that $D = D_1 + D_2 + \cdots + D_{n_{\mathrm{c}}}$. It is easy to see that the estimations of the eigenvalue count in Γ_ℓ can be executed independently. Below this independence, there is another independence: that of the solutions of the linear systems (8). Furthermore, the linear solver can be parallelized, if it is possible. Thus, our method is efficient on modern massively parallel computing environments.

3. Implementation

In this section, we describe a simple implementation of our method in which A is a Hermitian matrix and B is a non-singular Hermitian matrix. The algorithm of the implementation is shown in Fig. 1. For simplicity, we assume the Jordan curves are circles. This algorithm estimates the eigenvalue distribution in the interval $[\alpha, \beta]$ on the real axis. n_{c} circles are placed so that each circle occupies an equally separated sub-interval. ρ is the radius of all circles and γ_ℓ is the center of the ℓth circle. $\tilde{\mu}_\ell$ is the estimated eigenvalue count in the ℓth circle. The same number of quadrature points N is set for each circle.

Table 1. Matrix properties.

Matrix pencil	*Size*	*nnz(A)*	*nnz(B)*	*Type(A)*	*Type(B)*	*Center*	*Radius*	*#eig in* Γ
LUND	147	1298	1294	Indefinite	Indefinite	1.0×10^4	1.0×10^4	40
BCSST07	420	4140	3836	Positive definite	Positive semi-definite	0.23	0.17	398
PLAT1919	1919	17159	—	Indefinite	—	2.0×10^7	2.5×10^7	40
BCSST13	2003	42943	11973	Positive definite	Positive semi-definite	3.0×10^3	2.0×10^3	11

4. Numerical experiments

In this section, we perform numerical experiments to evaluate the efficiency of our method by using the algorithm shown in Fig. 1. Examples 1 and 2 are carried out using Matlab 7.4, and Example 3 is carried out using PGI Fortran 90. All operations are done in double precision arithmetic.

4.1 Example 1

In Example 1, we investigate how the eigenvalue count changes for an increase in the number of quadrature points N. We evaluate the effect of numerical integrations (3) on the eigenvalue count without trace estimations. The exact value of the matrix trace is calculated using the relation described in (1). The eigenvalues λ_j are obtained by Matlab function `eig`. The test problems were taken from Matrix Market; their properties are shown in Table 1. All eigenvalue problems are that of real symmetric matrices. We set $n_\mathrm{c} = 1$ for the algorithm. Columns *nnz(A)* and *nnz(B)* show the number of non-zero entries of matrices A and B, respectively. Columns *Type(A)* and *Type(B)* show the properties of A and B. Columns *Center* and *Radius* show the center and radius of the circles, respectively. The column *#eig in* Γ shows the number of eigenvalues in Γ. The number of eigenvalues is calculated by using the results of `eig`. The number of quadrature points N is set to be 4, 8, 16, 32, and 64. The results of this example are shown in Table 2. All results converge to the exact values.

Table 2. Results for Example 1.

N	*eigenvalue count*			
	LUND	BCSST07	PLAT1919	BCSST13
4	38.024	318.03	55.559	10.917
8	38.268	364.80	42.350	10.926
16	38.880	392.98	40.606	10.988
32	39.373	397.89	39.945	11.000
64	39.749	398.00	39.540	11.000
exact	40.000	398.00	40.000	11.000

Table 3. Results for Example 2.

#vectors	*eigenvalue count*			
	LUND	BCSST07	PLAT1919	BCSST13
10	44.344	391.08	40.759	12.866
20	43.394	392.58	40.371	11.747
30	43.195	391.92	40.926	10.765
40	39.547	393.83	39.874	10.590
50	40.039	393.09	41.018	10.313
100	37.716	392.27	40.632	11.293
200	39.805	393.45	40.341	11.460
500	41.147	392.76	40.542	11.104
1000	39.874	392.53	40.731	11.229
exact	38.880	392.98	40.606	10.988

4.2 Example 2

In Example 2, we investigate how the eigenvalue count changes for an increase in the number of sample vectors s. The test matrices used are the same as those in Example 1, s is set to from 10 to 1000, n_c is set to 1, and the linear systems are solved using the Matlab function `mldivide`. The number of quadrature points is set to $N = 16$. The elements of the sample vectors are given by the Matlab function `rand`, and their random seed is set by `rand('twister',5489)`. The results of this example are shown in Table 3. We consider the exact eigenvalue count $\hat{\mu}$ to be that shown for the $N = 16$ case in Table 2. Increasing s does not much effect the efficiency or accuracy of the eigenvalue count, even though it increases the computational cost. The trace estimation is slow in converging to the exact value because the convergence rate is $O(\sqrt{s})$. Similar results on trace estimations are shown in [6].

4.3 Example 3

In Example 3, the test matrix is derived from real-space density functional calculations [12,13]. It is a standard eigenvalue problem $A\boldsymbol{x} = \lambda \boldsymbol{x}$, where A is a real symmetric matrix and is only referenced in the form of matrix-vector multiplications. Thus, applying conventional approaches mentioned in Section 1 is not feasible in this case. In this problem, the M_B smallest eigenvalues are desired, where M_B is the total number of orbitals. The test matrix is derived from the density functional calculation of a 510-atom system of silicon. The matrix size is $n = 175,616$, and the smallest 1,020 eigenpairs are desired. The linear systems are solved by the shifted COCG method using stopping criterion 10^{-4}. One hundred circles are placed in the interval $[-0.230, 0.243]$. The number of quadrature points of each circle is $N = 8$, and the number of sample vectors is $s = 20$. The results are shown in Fig. 2. The horizontal axis indicates the index of the circles, and the vertical axis indicates the eigenvalue count for the circle's sub-domain. The exact values are calculated by the conjugate gradient method for eigenvalue problems [13]. Although s is significantly smaller than the matrix size n, our method roughly estimates the eigenvalue count. We obtained a rough eigenvalue distribution that can be used in setting parameters for an accurate eigensolver using only a few quadrature points and sample vectors.

The computational cost of the conjugate gradient method for eigenvalue problems is $O({M_\mathrm{B}}^3)$ (see [13]). We confirmed that the number of iteration of the shifted COCG method is proportional to n in preliminary experiments. The cost of the matrix-vector multiplication is $O(n)$ due to the sparsity of the matrix. Therefore, when s is set much less than n and the scalar recurrences are

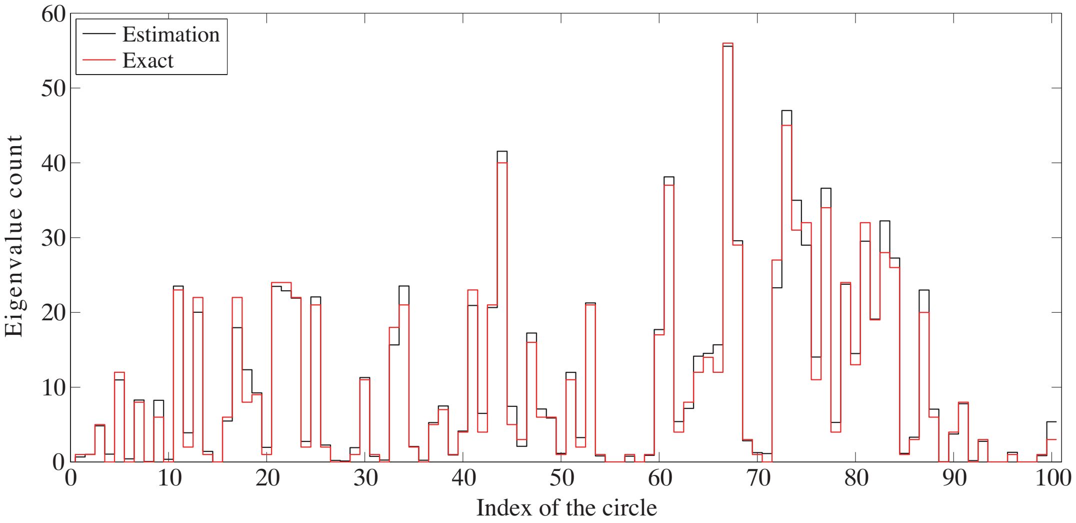

Fig. 2. Eigenvalue distribution of a 510-atom system of silicon.

introduced to the shifted COCG method, the computational cost of our method is $O(n^2)$. Since n, the number of grid points, is set to be proportional to M_{B}, for example $n \approx 200M_{\mathrm{B}}$, the cost of our method is $O({M_{\mathrm{B}}}^2)$ with a large coefficient. When the number of atoms in the target system is large, our method can be employed as a preprocessing of accurate eigensolvers, due to the lower order of computational cost and the high parallel performance.

5. Conclusions

We propose a stochastic estimation method of eigenvalue counting within a given closed curve. Our method is feasible for large sparse matrices or matrices that are only referenced in the form of matrix-vector multiplication. The stochastic estimation method for the eigenvalue distribution is defined by separating the given domain to several sub-domains and estimating the eigenvalue count in each sub-domain. Furthermore, because the computation of our method has independence, it is easy to execute on massively parallel computing environments. An acceleration technique is introduced to standard eigenvalue problems by using the shifted Krylov subspace method. We show using numerical examples that our method roughly estimates the eigenvalue distribution using only a few quadrature points and sample vectors. The parameters of eigensolvers can be effectively set by using a given knowledge of the eigenvalue distribution, and this distribution need not to be accurate, but does need to be computed at low cost. Our method is effective in such situations.

Acknowledgments

The authors would like to thank Jun-Ichi Iwata for helpful comments and suggestions on density functional calculations. This research was supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology, Japan, Grant Nos. 21246018 and 21105502.

References

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst eds., Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.

[2] T. Sakurai and H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, J. Comput. Appl. Math., **159** (2003), 119–128.

[3] T. Sakurai, J. Asakura, H. Tadano and T. Ikegami, Error analysis for a matrix pencil of Hankel matrices with perturbed complex moments, JSIAM Letters, **1** (2009), 76–79.

[4] K. Senzaki, H. Tadano, T. Sakurai and Z. Bai, A method for estimating the distribution of eigenvalues using the AS method, in: Proc. of Recent Advances in Numerical Methods for Eigenvalue Problems (RANMEP2008), to appear.

[5] Matrix Market, http://math.nist.gov/MatrixMarket.

[6] Z. Bai, M. Fahey and G. Golub, Some Large Scale Matrix Computation Problems, J. Comput. Appl. Math., **74** (1996), 71–89.

[7] M. F. Hutchinson, A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines, Commun. Stat., Simulation Comput., **19** (1990), 433–450.

[8] B. Jegerlehner, Krylov space solvers for shifted linear systems, arXiv:hep-lat/9612014v1, 1996.

[9] R. Freund, Solution of shifted linear systems by quasi-minimal residual iterations, in: Numerical Linear Algebra, L. Reichel, A. Ruttan, R. S. Varga and W. de Gruyter eds., Berlin, pp.101–121, 1993.

[10] R. Takayama, T. Hoshi, T. Sogabe, S. -L. Zhang and T. Fujiwara, Linear algebraic calculation of the Green's function for large-scale electronic structure theory, Phys. Rev. B, **73** (2006), 165108.

[11] S. Yamamoto, T. Sogabe, T. Hoshi, S. -L. Zhang and T. Fujiwara, Shifted conjugate-orthogonal-conjugate-gradient method and its application to double orbital extended Hubbard model, J. Phys. Soc. Jpn, **77** (2008), 114713.

[12] J. R. Chelikowsky, N. Troullier, K. Wu and Y. Saad, Higher order finite difference pseudopotential method: An applications to diatomic molecules, Phys. Rev. B, **50** (1994), 11355–11364.

[13] J.-I. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi, S. Okada and K. Yamada, A massively-parallel electronic-structure calculations based on real-space density functional theory, J. Comput. Phys., **229** (2010), 2339–2363.