International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:10, No:7, 2016

A Time-Reducible Approach to Compute Determinant |I-X|

Wang Xingbo

***Abstract*—**Computation of determinant in the form |I-*X*| is primary and fundamental because it can help to compute many other determinants. This article puts forward a time-reducible approach to compute determinant |I-*X*|. The approach is derived from the Newton's identity and its time complexity is no more than that to compute the eigenvalues of the square matrix *X*. Mathematical deductions and numerical example are presented in detail for the approach. By comparison with classical approaches the new approach is proved to be superior to the classical ones and it can naturally reduce the computational time with the improvement of efficiency to compute eigenvalues of the square matrix.

***Keywords*—**Algorithm, determinant, computation, eigenvalue, time complexity.

I. Introduction

COMPUTATION of determinants has been a topic ever since the being of the determinant in the 18th century, as stated in Ershaidat's introductory essay [1]. Approaches and Algorithms have been continuously developed to improve the efficiency of the computations. This point of view can be testified by part of literatures published in the last 15 years. For example, [2] and [3] researched Division-Free Algorithm that was early researched by [4]; [5] studied general computation of Determinant; [6] improved algorithms for computing determinants and resultants; [7] put forward a way to compute determinants by contour integrals; [8] summarized and exhibited conventional and skillful methods to compute determinants through concrete computing examples in 2010 and [9] presented a robust algorithm for accurately computing the determinant by using accurate matrix factorizations in 2012. Reference [10] presented a new parallel algorithm for finding determinants of $n \times n$ matrices, [11], [12] also researched parallel algorithms; [13] studied the way to compute determinants in the multiprocessor computer. Special algorithms to special determinants were also developed as see in [14]-[16]. All these facts imply that a better algorithm is still in need.

In the family of determinants, a kind in the form of |*I-X*|, where the symbol |*A*| denotes the determinant of a square matrix *A*, is particularly essential because it can be used to compute the other determinants. For example, the approach to compute |*I-X*| is obviously available to compute $|I-X^k|(k>1)$, and hence $|I+X|$ is easily computed since $I+X=(I-X^2)(I-X)^{-1}$ provided $|I-X|\neq 0$. Actually, there are more behaviors that show significance of |*I-X*|. Therefore, this article makes a pioneer investigation and proposes an approach to compute such determinant. The article is composed of 5 parts. The first part is this introductory section; the second part lists some necessary preliminaries; the third part demonstrates main results with their mathematical deductions; the fourth part presents an algorithm for the computation and numerical examples. In the end, conclusions and expectations are put forward.

Wang Xingbo is with the Department of Mechatronics, Foshan University, Foshan City, Guangdong Province, PRC (e-mail: wxbmail@msn.com).

II. Preliminaries

In this whole article, symbol $|A|$ is to denote the determinant of a square matrix *A*; and the following lemmas, which are stated in [17]-[19], are necessary for later sections.

Lemma 1. The characteristic polynomial of an *n*×*n* matrix *A*, $f_A(\lambda)$, is defined by:

$$f_A(\lambda)=|\lambda I-A|=\lambda^n+c_1\lambda^{n-1}+...+c_k\lambda^{n-k}+...+c_{n-1}\lambda+c_n \tag{1}$$

where the coefficients $c_k(k=1,2,...,n)$ are real numbers.

The roots of $f_A(\lambda)$, $\lambda_i(1\le i\le n)$, are *A*'s eigenvalues and it needs no more than $O(n^3)$ time to compute all the eigenvalues .

Lemma 2. Let $\lambda_i(i=1,2,...,n)$ be the *n* complex eigenvalues of an *n*×*n* matrix *A* and $T_k=\sum_i \lambda_i^k$; then it holds:

$$T_k+c_1T_{k-1}+c_2T_{k-2}+...+c_iT_{k-i}+...+c_{k-1}T_1+kc_k=0, k=1,2,...,n \tag{2}$$

Lemma 3. T_k defined in Lemma 2 is the trace of A^k, namely, $T_k=tr(A^k)$.

III. Main Results and Proofs

Theorem 1. Let the characteristic polynomial of an *n*×*n* matrix *X* be given by

$$f_X(\lambda)=|\lambda I-X|=\lambda^n+c_1\lambda^{n-1}+...+c_{n-1}\lambda+c_n \tag{3}$$

then

$$|I-X|=1+\sum_{i=1}^{n}c_i \tag{4}$$

Proof. (Omit).

Theorem 2. Let X be an $n\times n$ matrix; then the time complexity of computing $|I-X|$ is no more than that of computing all the eigenvalues of X.

Proof. By Lemma 2 we can drive out the so-called Newton's identities as follows

$$
\begin{aligned}
&T_1 + c_1 = 0\\
&T_2 + c_1T_1 + 2c_2 = 0\\
&T_3 + c_1T_2 + c_2T_1 + 3c_3 = 0\\
&.....\\
&T_k + c_1T_{k-1} + c_2T_{k-2} + ... + c_{k-1}T_1 + kc_k = 0\\
&.....\\
&T_{n-1} + c_1T_{n-2} + c_2T_{n-3} + ... + c_{n-2}T_1 + (n-1)c_{n-1} = 0\\
&T_n + c_1T_{n-1} + c_2T_{n-2} + ... + c_{n-2}T_2 + c_{n-1}T_1 + nc_n = 0
\end{aligned}
$$

That is

$$
\begin{cases}
c_1 = -T_1\\
c_2 = -\dfrac{1}{2}(T_2 + c_1T_1)\\
c_3 = -\dfrac{1}{3}(T_3 + c_1T_2 + c_2T_1)\\
.....\\
c_k = -\dfrac{1}{k}(T_k + c_1T_{k-1} + c_2T_{k-2} + ... + c_{k-1}T_1)\\
.....\\
c_{n-1} = -\dfrac{1}{n-1}(T_{n-1} + c_1T_{n-2} + c_2T_{n-3} + ... + c_{n-2}T_1)\\
c_n = -\dfrac{1}{n}(T_n + c_1T_{n-1} + c_2T_{n-2} + ... + c_{n-2}T_2 + c_{n-1}T_1)
\end{cases}
\tag{5}
$$

This says that, $c_k(k=1,2,...,n)$ can be computed recursively in term of $T_j(j=1,2,...,k)$.

Now it shows the way to compute all T_k by $T_k = \sum_i \lambda_i^k \ (k=1,2,...,n)$; It is believed that this procedure will take $O(n^2)$ time complexity and n^2 spatial complexity. In fact, if all the eigenvalues $\lambda_i \ (1\le i\le n)$ are computed, $T_k(1\le k\le n)$ can be obtained according to following strategy.

$$
\begin{aligned}
&\lambda_1 \ \rightarrow \ \lambda_1^2,\ \lambda_1^3,\ ...,\ \lambda_1^n \ \Rightarrow \ n-1 \ \textit{multiplications}\\
&\lambda_2 \ \rightarrow \ \lambda_2^2,\ \lambda_2^3,\ ...,\ \lambda_2^n \ \Rightarrow \ n-1 \ \textit{multiplications}\\
&.....\\
&\lambda_i \ \rightarrow \ \lambda_i^2,\ \lambda_i^3,\ ...,\ \lambda_i^n \ \Rightarrow \ n-1 \ \textit{multiplications}\\
&.....\\
&\downarrow \qquad \downarrow \qquad ...,\ \downarrow\\
&\sum_i \lambda_i,\ \sum_i \lambda_i^2,\ ... \qquad \sum_i \lambda_i^n,\ \Rightarrow \ n^2 \ \textit{additions}
\end{aligned}
$$

Obviously, the lowest row, $\sum_i \lambda_i^k (1\le k\le n)$, gives the values of $T_k(1\le k\le n)$.

Now $c_k(1\le k\le n)$ can be computed by (5) and it will take at most $1+3+...+(2n-1)=n^2$ operations, namely, $O(n^2)$ time complexity. Consequently it is clear that time complexity of computing $c_k(1\le k\le n)$ is no more than that of computing all the eigenvalues of X.

Preposition 1. Let $\boldsymbol{A}$ be a square matrix; then $|A|$ can be computed in no more than $O(n^3)$ time.

Proof. Let $X = I - A$; then $|A| = |I - X|$. By theorem 2, the time complexity of computing $|I-X|$ is no more than that of computing all the eigenvalues of X. By Lemma 1, it takes no more than $O(n^3)$ time to compute all the eigenvalues of X. Hence the theorem holds.

IV. Algorithm Design and Numerical Experiments

A. Algorithm Design

Based on Theorem 2, an algorithm, which is for convenience called Algorithm I in later section, can be designed to compute the determinant |*I*-*X*| by following 4 steps.

Step 1. Compute all the eigenvalues $\lambda_1,\lambda_2,...,\lambda_n$ of X;

Step 2. Compute $T_k = \sum_i \lambda_i^k$;

Step 3. $c_0 = 1$; compute $c_k(1\le k\le n)$ by $c_{k+1} = -\dfrac{1}{k}\boldsymbol{C}_k \cdot \boldsymbol{T}_k$, where $\boldsymbol{C}_k = (c_0, c_1, c_2, ..., c_{k-1})$ and $\boldsymbol{T}_k = (T_k, T_{k-1}, T_{k-2}, ..., T_1)$

Step 4. $C = \sum_{i=1}^{n} c_i$ and $|I - X| \Leftarrow 1 + C$.

Note that, the Step 3 adopts donations of vectors and their dot product.

B. Numerical Example

The above algorithm I can be demonstrated by the following sample. Just consider a matrix *I*+*X* defined by

$$
I - X = \begin{pmatrix} -6 & 12 & -4 \\ -4 & -8 & -4 \\ -4 & -12 & -6 \end{pmatrix}
$$

then;

$$
X = \begin{pmatrix} 7 & -12 & 4 \\ 4 & 9 & 4 \\ 4 & 12 & 7 \end{pmatrix}
$$

So that

$$
f_X(\lambda) = \lambda^3 - 5\lambda^2 + 3\lambda + 9 = (\lambda - 11)(\lambda - 9)(\lambda - 3)
$$

and the roots of $f_X(\lambda)$ are:

$$\lambda_1 = 3, \lambda_2 = 9, \lambda_3 = 11$$

Consequently, it yields;

$$T_1 = \lambda_1 + \lambda_2 + \lambda_3 = 23,$$
$$T_2 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 211,$$
$$T_3 = \lambda_1^3 + \lambda_2^3 + \lambda_3^3 = 1331 + 729 + 27 = 2087$$

and

$$c_1 = -T_1 = -23$$
$$c_2 = -\frac{1}{2}(T_2 + c_1 T_1) = -\frac{1}{2}(211 - 529) = 159$$
$$c_3 = -\frac{1}{3}(T_3 + c_1 T_2 + c_2 T_1) = -\frac{1}{3}(2087 - 4853 + 3657) = -297$$
$$| I + X | = 1 + c_1 + c_2 + c_3 = -160$$

Note that

$$| I - X | = \begin{vmatrix} -6 & 12 & -4 \\ -4 & -8 & -4 \\ -4 & -12 & -6 \end{vmatrix} = -160$$

It verifies that Algorithm I is valid.

V. Conclusions and Expectations

As introduced in the introductory section, algorithms to compute determinants have continuously been developed and each algorithm has its own life cycle and application occasion. Generally speaking, any new algorithm or approach shall ensure its existence, necessity, superior and what to be improved. Hence, this section will establish such things of the new approach.

A. Necessity and Superior

Many schoolbooks exhibit that determinant of a square matrix X can be calculated by multiplication of all the eigenvalues of X. Based on this fact, the following algorithm, which is called algorithm II, can be designed by two steps.

Step 1. Compute all the eigenvalues $\lambda_1, \lambda_2, \ldots$ of *I-X*;

Step 2. $| I - X | = (-1)^n \lambda_1 \lambda_2 \ldots \lambda_n$.

This algorithm looks simple and fits for the knowledge of schoolbooks. But, in point of view of an algorithm, it is not a good one because it has shortcoming when it is used to compute $| I - X^k | (k \geq 1)$. In fact, by the algorithm II it needs first computing X^k and $I - X^k$ before it begins to compute the eigenvalues of the matrix $I - X^k$. The time complexity of computing X^k is $O((k-1)n^3)$. Obviously, when $k > n$, it is bigger than $O(n^4)$. On the contrary, the algorithm I only needs computing the eigenvalues of X, saving a lot of time. Therefore, the algorithm I is superior to the algorithm II.

Except for the algorithm II, schoolbooks also show that the Gaussian elimination and the LU decomposition are two classical approaches to compute the determinants, as seen in [20] and [21] respectively. It is known that these two approaches take at least $O(n^3)$ time complexity. Comparing this fact to Theorem 2, it can immediately see that the algorithm I is superior to the classical two ones owning to the following two advantages.

1. Improvement of computing the eigenvalues of a square matrix will consequently increase efficiency of the algorithm I.
2. Computation of T_k in Step 3 of the algorithm I can be done in a parallel procedure, as seen in the proof of Theorem 2.

B. What to Be Improved

According to Theorem 2, high efficiency of computing the eigenvalues of a square matrix becomes an improved point to the algorithm I. This is also the expectation to future work.

Acknowledgement

The research work is supported by the national Ministry of science and technology under project 2013GA780052, Department of Guangdong Science and Technology under projects 2015A030401105 and 2015A010104011, Foshan Bureau of Science and Technology under projects 2013AG10007, Special Innovative Projects 2014KTSCX156, 2014SFKC30 and 2014QTLXXM42 from Guangdong Education Department. The author sincerely presents thanks to them all.

References

[1] Ershaidat, M, N. History of Matrices and Determinants, Yarmouk University, Irbid Jordan, 2007, 1-6.
[2] Gunter Rote. Division-Free Algorithms for the Determinant and the Pfaffian: Algebraic and Combinatorial Approaches, Computational Discrete Mathematics, LNCS 2122, pp. 119–135, 2001.
[3] Seifullin T R. Acceleration of Computation of determinants and Characteristic polynomials without Divisions, Cybernetics and Systems Analysis, Vol. 39, No. 6, 2003.
[4] Kaltofen E. On Computing Determinants of Matrices without Divisions. Proceedings of the 1992 International Symposium on Symbolic Algebraic Computing, 1992:342-349.
[5] Villard G. Computation of the Inverse and Determinant of a Matrix (J), INRIA, (2003), pp. 29–32.
[6] Ioannis Z. Emirisa, and Victor Y. Panb, Improved Algorithms for computing determinants and resultants. Journal of Complexity 21 (2005) 43–71.
[7] Klaus Kirsten. Computation of determinants using contour integrals, American Journal of Physics.76:60-64,2008.
[8] Keček, Damira. Methods of computing determinants of the n-th order, Osječki matematički list, Vol.10 No.1 October 2010. 31–42.
[9] T Ogita. Robust Computation of Determinant (C). American Institute of Physics Conference Series, 2012, 1504(10):1119-1123.
[10] Almalki S, Alzahrani S. A Alabdullatif. New parallel algorithms for finding determinants of N×N matrices, World Congress on Computer & Information Technology, 2013:1-6.
[11] Beliakov G, Matiyasevich Y.A Parallel Algorithm for Calculation of Large Determinants with High Accuracy for GPUs and MPI clusters, arXiv:1308.1536v2,2013(8).
[12] Salman H. Abbas. Computing the Determinants in the Multiprocessor Computer. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 8, August 2014.

[13] G Beliakov, Y Matiyasevich. A parallel algorithm for calculation of determinants and minors using arbitrary precision arithmetic. Bit Numerical Mathematics, 2015:1-18.
[14] Jia L, Yao GT. On computation determinant of circulant matrices and its application (J), Journal of Xinyang Teachers College, 2014, 18(2):131-132.
[15] Elouafi M., Dah Ahmed. A New Algorithm for the Determinant and the Inverse of Banded Matrices (J).Open Access Library Journal, 2014, 01:1-5.
[16] Awol Assen, Venkateswara Rao J.A Study on the Computation of the Determinants of a 3x3 Matrix, International Journal of Science and Research, 2014,3(6):912-921.
[17] Wang Xingbo, Xian Yaoqi. How Difficult to Compute Coefficients of Characteristic Polynomial? *International Journal of Scientific and Innovative Mathematical Research (IJSIMR),* 2016,3(1):7-12.
[18] K Singh.Linear algebra step by Step, Oxford Press, 2013,pp517.
[19] C D Meyer. Matrix Analysis and Applied Linear Algebra, SIAM, Philadelphia, 2000,pp489-660.
[20] G W Stewart, Matrix Algorithms, SIAM, 1998,pp176-177.
[21] S S. Skiena, The Algorithm Design Manual, *Springer- Verlag London Limited*, 2008, pp404-406.