International Journal of Mathematics and
Computer Science, **14**(2019), no. 1, 9–16

Comparison of Computer Execution Time of Cornice Determinant Calculation

Armend Salihu[1], **Azir Jusufi**[1], **Fatlinda Salihu**[2]

[1]Universiteti per Biznes dhe Teknologji
Kalabria p.n., Prishtine, Kosovo
[2]Universiteti i Prishtines, Bregu i
Diellit p.n., Prishtine, Kosovo

email: ar.salihu@gmail.com, azir.jusufi@ubt-uni.net,
musliu.fatlinda@gmail.com

(Received March 20, 2018, Accepted April 28, 2018)

Abstract

Based on the facts that determinants and matrices have extensive application in scientific research and that in this technology epoch almost all different researches and simulations are made using computer systems, we present a comparison of the time duration for calculating cornice determinants through computer systems. In this paper, we develop computer functions for calculating determinants using the Laplace expansion method, Gjanbalaj-Salihu's method, and Salihu's SemiDiagonal Method. We compare the time duration for the execution of these functions to calculate the final result of the determinant.

1 Main definitions and lemmas

The Determinant of a $n \times n$ matrix, is the sum:

$$\det(A) = |A| = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum_{S_n} \varepsilon_{j_1,j_2,\ldots,j_n} a_{1j_1} a_{2j_2} \ldots a_{nj_n} \quad (1.1)$$

Key words and phrases: Determinants, computer function, determinant calculation, time comparison.
AMS (MOS) Subject Classifications: 08A70, 11C20, 15A15, 65F40, 65Y04, 97N80.
ISSN 1814-0432, 2019, http://ijmcs.future-in-tech.net

ranging over the symmetric permutation group S_n, where

$$\varepsilon_{j_1,j_2,\ldots,j_n} = \begin{cases} +1, \text{ if } j_1, j_2, \ldots, j_n \text{ is an even permutation} \\ -1, \text{ if } j_1, j_2, \ldots, j_n \text{ is an odd permuation} \end{cases} \tag{1.2}$$

Definition 1: Every determinant having the second row and $(n-1)$th row, as well as the second column and $(n-1)$th column except the first and last elements of these rows/columns, is called the cornice determinant [1].

Lemma 2: Suppose that A is a matrix of order $n \times n$. Fix row i and column j for every $i, j \in \{1, 2, \ldots, n\}$ [2]. The determinant of a matrix A can be calculated as follows:

$$|A| = \sum_{i`=1}^{n} a_{i`j} |C_{i`j}| = \sum_{j`=1}^{n} a_{ij`} |C_{ij`}| \tag{1.3}$$

Lemma 3: Every cornice determinant $|A_{n\times n}|$, $n \times n$, $(n \geq 5)$ can be computed by reducing the order of the determinant by four [1]:

$$|A_{n\times n}| = (a_{12}a_{21}a_{n,n-1}a_{n-1,n} - a_{12}a_{2,n}a_{n,n-1}a_{n-1,1} - a_{21}a_{n2}a_{n-1,n}a_{1,n-1} + \\ + a_{1,n-1}a_{2,n}a_{n2}a_{n-1,1}) \cdot |A_{(n-4)\times(n-4)}| \tag{1.4}$$

Lemma 4: Every cornice determinant can be transformed into the semi-diagonal determinant and the result can be calculated using formula below [3]:

$$|A_{n\times n}| = a_{12} \cdot a_{21} \cdot b_{n-1,n-1} \cdot b_{nn} \cdot |A_{(n-4)\times(n-4)}| \tag{1.5}$$

where

$$b_{n-1,n-1} = a_{n,n-1} - \frac{a_{1,n-1} \cdot a_{n2}}{a_{12}}$$

$$b_{nn} = a_{n-1,n} - \frac{a_{2,n} \cdot a_{n-1,1}}{a_{21}}$$

$$|A_{(n-4)\times(n-4)}| = \begin{vmatrix} a_{33} & \cdots & a_{3,n-2} \\ \vdots & \ddots & \vdots \\ a_{n-2,3} & \cdots & a_{n-2,n-2} \end{vmatrix} \tag{1.6}$$

2 Computer comparison of determinants calculation methods

For computer computation of the duration of the determinant calculation, we used a computer with the characteristics presented in Table 1:

Table 1: Computer characteristics used to simulate the calculation of determinants:

Name:	Lenovo
Model:	Ideapad 700-15ISK
CPU:	Intel Core i7 6700HQ 2.6Ghz
RAM:	16 GB DDR4
GPU:	FULL HD Display 15.6" 1920x1080, nVidia GTX 950 4096 mb dedicated graphics
HDD:	256 GB SSD

while the software used for this simulation is presented in table 2:

Table 2: Computer tools used for determinant calculation simulation:

OS	Windows 10 Pro 64-bit, Version 1703 (OS Build 15063.483
Software	MATLAB, Version 9.0 (R2016a), 64-bit (win64)

2.1 Comparison between: Laplace, Gjonbalaj-Salihu and Salihu's SemiDiagonal

To compare cornice determinant calculations among: Laplace, Gjonbalaj-Salihu and Salihu's SemiDiagonal, we use the following functions:

$function\ d = det\_Laplace(A)$
$[m, n] = size(A)$; % matrix size; check if it is square
if $m \sim= n$ disp('Matrix A is not square, cannot calculate the determinant')
$d = 0$;
return
end
if $n == 1$
$d = A$;
return
elseif $n == 2$
$d = A(1,1) * A(2,2) - A(1,2) * A(2,1)$;

return
end
$w = 1$; $d = 0$;
for $i = 1 : n$
if $i == 1$
$d = d + w * A(1, i) * det\_Laplace(A(2 : n, 2 : n))$;
elseif $i == n$
$d = d + w * A(1, i) * det\_Laplace(A(2 : n, 1 : n - 1))$;
else
$d = d + w * A(1, i) * det\_Laplace(A(2 : n, [1 : i - 1, i + 1 : n]))$;
end
$w = -w$; % alternating sign of the weight
end

$function\ d = det\_Cornice(A)$
$[m, n] = size(A)$; % matrix size; check if it's square
if $m \sim= n$ disp('Matrix A is not square, cannot calculate the determinant')
$d = 0$;
return
end
if $n == 1$
$d = A$;
return
elseif $n == 2$
$d = A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)$;
return
elseif $n == 3$
$d = det(A)$;
return
elseif $n == 4$
$d = det(A)$;
return
end
$d0 = det\_Laplace(A(3 : n - 2, 3 : n - 2))$;
if $d0 == 0$
$d = 0$; % to avoid dividing by zero
else

$d = (A(1,2)*A(2,1)*A(n,n-1)*A(n-1,n)-A(1,2)*A(2,n)*A(n,n-1)*A(n-1,1)-A(2,1)*A(n,2)*A(n-1,n)*A(1,n-1)+A(1,n-1)*A(2,n)*A(n,2)*A(n-1,1))*d0;$
end

$function\ d = det\_SemiDiagonal(A)$
$[m,n] = size(A)$; % matrix size; check if it is square
if $m \sim= n$ disp('Matrix A is not square, cannot calculate the determinant')
$d = 0$;
return
end
$B(n-1,n-1) = A(n,n-1) - A(1,n-1)*A(n,2)/A(1,2)$;
$B(n,n) = A(n-1,n) - A(2,n)*A(n-1,1)/A(2,1)$;
if $n == 1$
$d = A$;
return
elseif $n == 2$
$d = A(1,1)*A(2,2) - A(1,2)*A(2,1)$;
return
elseif $n == 3$
$d = det(A)$;
return
elseif $n == 4$
$d = det(A)$;
return
end
$d0 = det\_Laplace(A(3:n-2, 3:n-2))$;
if $d0 == 0$
$d = 0$; % to avoid dividing by zero
else
$d = A(1,2)*A(2,1)*B(n-1,n-1)*B(n,n)*d0$;
end

$n = m \times m$; % matrix rank
$A = [m \times m]$
disp('Calculation of Cornice Determinants using Gjonbalaj-Salihu's')
tic $d1 = det\_cornice(A)$ toc
disp('Calculation of Cornice Determinants using Laplace Method')

tic $d2 = det\_Laplace(A)$ toc
disp('Calculation of Cornice Determinants using Salihu's SemiDiagonal')
tic $d3 = det\_SemiDiagonal(A)$ toc

Executing the last function we get the results in Table 3, expressed in seconds:

Table 3: Comparison of Execution time of cornice determinant calculation using: Laplace Method, Gjonbalaj-Salihu, and Salihu's SemiDiagonal:

Order of Determinant	Laplace	Gjonbalaj-Salihu	Salihu's SemiDiagonal
	1	2	3
5×5	0.000402	0.000117	0.000113
6×6	0.002432	0.000610	0.000362
7×7	0.012245	0.010810	0.000792
8×8	0.084689	0.000711	0.000436
9×9	0.694358	0.000573	0.000339
10×10	6.530343	0.002588	0.002418
11×11	70.400784	0.012100	0.011520
12×12	1301.954964	0.077508	0.072570

Difference		
1-2	1-3	2-3
0.000285	0.000289	0.000004
0.001822	0.002070	0.000248
0.001435	0.011453	0.010018
0.083978	0.084253	0.000275
0.693785	0.694019	0.000234
6.527755	6.527925	0.000170
70.388684	70.389264	0.000580
1301.877456	1301.882394	0.004938

From table 3, observe that the Gjonbalaj-Salihu's method is executed 82% faster than the Laplace method, Salihu's SemiDiagonal Method is executed 93% faster than Laplace method and 29% faster than the Gjonbalaj-Salihu's method.

2.2 Comparison between: MATLAB det function, Gjonbalaj-Salihu and Salihu's SemiDiagonal

To compare cornice determinants calculation among: MATLAB det function, Gjonbalaj-Salihu, and Salihu SemiDiagonal, we use:

$n = m \times m$; % matrix rank
$A = [m \times m]$
disp('Calculation of Cornice Determinants using Gjonbalaj-Salihu's')
tic $d1 = det\_cornice\_1(A)$ toc
disp('Calculation of Cornice Determinants using Matlab Det Function')
tic $d2 = det(A)$ toc
disp('Calculation of Cornice Determinants using Salihu‘s SemiDiagonal')
tic $d3 = det\_SemiDiagonal\_1(A)$ toc

For better comparison, the functions $det\_cornice\_1$ and $det\_SemiDiagonal\_1$ are the same with $det\_cornice$ and $det\_SemiDiagonal$, with the following difference in row 20 respectively 22 are $d0 = det(A(3 : n-2, 3 : n-2));$ instead of $d0 = det\_Laplace(A(3 : n-2, 3 : n-2));$,

From execution of last function we get the results as provided in Table 4, expressed in seconds:

Table 4: Comparison of Execution time of cornice determinant calculation using: Matlab det function, Gjonbalaj-Salihu, and Salihu's SemiDiagonal:

Order of Determinant	Matlab "det" Function	Gjonbalaj-Salihu	Salihu's SemiDiagonal	Difference		
	1	2	3	1-2	1-3	2-3
5×5	0.000570	0.000465	0.000433	0.000105	0.000137	0.000032
6×6	0.000516	0.000369	0.000361	0.000147	0.000155	0.000008
7×7	0.000502	0.000393	0.000367	0.000109	0.000135	0.000026
8×8	0.000179	0.000178	0.000154	0.000001	0.000025	0.000024
9×9	0.000229	0.000222	0.000197	0.000007	0.000032	0.000025
10×10	0.000533	0.000397	0.000369	0.000136	0.000164	0.000028
11×11	0.000348	0.000277	0.000244	0.000071	0.000104	0.000033
12×12	0.000282	0.000269	0.000253	0.000013	0.000029	0.000016

From Table 4, observe that the Gjonbalaj-Salihu's method is executed 15% faster than the MATLAB det function, Salihu's SemiDiagonal Method is executed 22% faster than Matlab "det" function and 8% faster than the Gjonbalaj-Salihu's method.

References

[1] Q. Gjonbalaj, A. Salihu, Computing the determinants of n x n (n = 5) matrices by reducing the order of the determinant by four, Applied Mathematical E-Notes, (2010), 151-158.

[2] Pierre-Simon (de) Laplace, Expansion of determinants in terms of minors, Researches sur le calcul integral et sur le systeme du monde, Histoire de l'Academie Royale des Sciences (Paris), seconde partie, (1772).

[3] A. Salihu, A modern modification of Gjonbalaj-Salihu cornice determinant, transformation to semi-diagonal determinant, International Journal of Mathematics and Computer Science, **13,**(2018), no. 2, 133-138.