Vol. 3, No 1 June 2020
ISSN: 2016-1303 | ***Web:*** *www.cjpas.fulokoja.edu.ng*

VARIATIONS OF CRAMER'S RULE IN WZ FACTORIZATION

Olayiwola Babarinsa[1]& Edogbanya Helen Olaronke[*2]
Department of Mathematical Sciences, Federal University Lokoja, Kogi State, Nigeria.
E-mail:olayiwola.babarinsa@fulokoja.edu.ng, *helen.edogbanya@fulokoja.edu.ng

ABSTRACT
This article compares four established variations of Cramer's rule to solve the linear systems in WZ factorization. The methods are implemented on MATLAB and the results show the matrix norms of the methods are better than classical Cramer's rule. The application of the variations of Cramer's rule in WZ factorization can now be compared on mesh multiprocessors, since the methods minimize round-off error.

Keywords: *Cramer's rule, Linear system, W Z factorization, MATLAB, matrix norm.*

2010 MSC: 5A06, 15A23, 15B99

1.0 INTRODUCTION

WZ factorization of nonsingular matrix was proposed by Evans and Hatzopoulos(D. J. Evans & Oksa, 1997). The factorization has been applied in scientific computing such as in science and engineering due to its existence and uniqueness, see (Bylina, 2018; Efremides, Bekakos, & Evans, 2002; Evans & Hatzopoulos, 1979; D. J. Evans, 2002; Rao, 1997; Rhofi & Ameur, 2016). WZ factorization relies on nonsingular central submatrices where it simultaneously computes two matrix elements(Heinig & Rost, 2004).WZ factorization is an alternative to LU factorization.In WZ factorization of nonsingular matrix B, W-matrix (bow-tie matrix) and Z-matrix (hourglass matrix) which are also known as interlocking quadrant factors of B

Coexist such that B = WZ

Thus, WZ factorization executes components in parallel with $\frac{n}{2}$ steps if n is even or $\frac{n-1}{2}$

steps if n is odd (Bylina & Bylina, 2009). Many 2x2 linear systems are solved during WZ factorization to obtain the elements in W -matrix and then to compute Z-matrix. These 2x2 linear systems in WZ factorization are solved by classical Cramer's rule for over three decades. Although Cramer's rule is assumed to be less practical due to its setbacks, many modifications have been made on Cramer's rule to solve simple linear systems, see(Babarinsa & Kamarulhaili, 2019; Heinig & Rost, 2005; Ufuoma, 2013) and the references therein. Due to round off errors which may become significant on problems with non-integer coefficients, Moler (1974) expressed that Cramer's rule is unsatisfactory even for 2x2 linear systems because of round off errors. However, Dunham (1980) gives a counter example to the statement to show that Cramer's rule is satisfactory. Thus, accurate methods to evaluate determinants make Cramer's rule numerically stable (Habgood & Arel, 2012). In Section 2, we apply four variations of Cramer's rule proposed by 'Babarinsa and Kamarulhaili (2017, 2019). The methods are implemented on MATLAB R2018b for selected sparsematrices to obtain the matrix norms.

2.0 METHOD

SOLVING LINEAR SYSTEMS IN W Z FACTORIZATION WITH VARIATIONS OF CRAMER'S RULE

A linear system is defined as(Hogben, 2007)

$$Bx = c,$$

When

$|B| \neq 0$ $x = [x_1, x_2, \ldots, x_n]^T$,
$c = [c_1, c_2, \ldots, c_n]^T$ $B \in R^{n\times n}; x, c \in R^n$

Theorem 2.1.'(Brunetti & Renato, 2014)*[Cramer's rule] Let* $Bx = c$ *be an n x n system of linear equation and B an n x n nonsingular matrix, then the unique solution x =* $[x_1, x_2, \ldots, x_n]^T$ *to the linear system is given by*

$$x_i = \frac{det(B_{i|c})}{det(B)} \qquad (2)$$

Where $B_{i/c}$ is the matrix obtained from coefficient matrix *B* by substituting the column vector c to the *ith* column of *B*, for *I = 1,2,......n.*

It is a well-established theorem that if the *ith* column of matrix B is the sum or difference of the *ith* column of matrix *ith* column of matrix D and other columns in *C and D* are equal to the corresponding columns in *B* (Lipschutz, Lipson, & Lipschutz, 2009). Then det (B) = det (C) + det (D) (3)

More so, if the *ith* column of matrix *B* is replaced with the row sum of its matrix to obtain a new matrix $B\alpha\ i$ with allother columns in *B* and $B\alpha i$

remain the same, for *i* = 1,2,.....n, Then, the determinant of the matrix and the obtained matrix are equal (Babarinsa & Kamarulhaili, 2017). That is, $det\ (B) = det\ (B^{\alpha_i}).$ (4)

Now, we can deduce that if column vector c is added to or subtracted from the *ith* column of matrix $B\alpha i$ (i.e the *ith* column of matrix *B* where its row sum replaced), then we can re-write equation (3) via (4) as

$$det\left(B^{\alpha_i}_{i\pm c}\right) = det(B^{\alpha_i}) \pm det\left(B^{\alpha_i}_{i|c}\right),$$

where $B^{\alpha_i}_{i\pm c}$ is the matrix obtained from $B\alpha i$ by

adding (or subtracting) the column vector c to (or from) the B^{α_i} $B^{\alpha_i}_{i/c}$ is the matrix obtained from B^{α_i}

by substituting column vector c to the *ith* of matrix B is replaced by its row sum. It is important to note that if $det\ (B) = det\ (B^{\alpha_i})$, then

$$det\left(B_{i|c}\right) = det\left(B^{\alpha_i}_{i|c}\right). \qquad (6)$$

Corollary 2.2.(Babarinsa & Kamarulhaili, 2017) *Let Bx=c be qn n x n system of linear equation and B an n x n nonsingular matrix of x, then the ith entry xi of the unique solution* $x = (x_1, x_2, \ldots, x_n)^T$ *to the linear system is given by*

$$x_i = \frac{det(B_{i+c})}{det(B)} - 1, \qquad (7)$$

When B_{i+c} is the matrix obtained from B by adding the constant terms of vector C to the ith column of B, for I= 1, 2, ..., n

Corollary 2.3. (Babarinsa & Kamarulhaili, 2017) *Let Bx=C be an n* x *n system of linear equation and B an n* x *n nonsingular matrix of x, then the ith entry xi of the unique solution*

$x = (x_1, x_2, \ldots, x_n)^T$ *to the linear system is given by*

$$x_i = 1 - \frac{det(B_{i-c})}{det(B)}, \qquad (8)$$

When B_{i-c} is the matrix obtained from *B* by subtracting the constant terms of vector *C* from the *ith* column of *B,* for *I = 1, 2, ..., n*

Corollary 2.4. (Babarinsa & Kamarulhaili, 2019) *Let Bx=C be an n* x *n system of linear equation and B a square matrix of x, then the ith entry xi of the unique solution x =* $(x_1, x_2, \ldots, x_n)^T$ *to the linear system is given by*

$$x_i = \frac{\det\left(B^{\alpha_i}_{i+c}\right)}{\det(B^{\alpha_i})} - 1, \qquad (9)$$

When $B^{\alpha_i}_{i+c}$ *is the matrix obtained from* B^{α_i} *adding the column vector c to the ith column of* B^{α_i} *and* B^{α_i} *is the matrix obtained from B with it ith its column being replaced by the row sum of B, for I = 1, 2, ..., n*

Corollary 2.4. (Babarinsa & Kamarulhaili, 2019) *Let Bx=C be an n* x *n system of linear equation and B a square matrix of x, then the ith entry xi of the unique solution x =* $(x_1, x_2, \ldots, x_n)^T$ to the linear system is given by $x_i = 1 - \frac{\det\left(B^{\alpha_i}_{i-c}\right)}{\det(B^{\alpha_i})},$ (10)

When $B^{\alpha_i}_{i+c}$ *is the matrix obtained from* B^{α_i} by subtracting the column vector c from the column of B^{α_i} *and* B^{α_i} is the matrix obtained from B with it *ith* column being replaced by the row sum of B from *i -= 1, 2, ..., n*

 ISSN: 2016-1303 | ***Web:*** *www.cjpas.fulokoja.edu.ng*

NUMERICAL EXAMPLE

Given the linear equation

$$2x - 3y = -7$$
$$4x + 5y = -3$$

Corollary 1

$$B_{1+c} = \begin{bmatrix} 2+(-7) & -3 \\ 4+(-3) & 5 \end{bmatrix} = \begin{bmatrix} -5 & -3 \\ 1 & 5 \end{bmatrix}$$

$$B_{2+c} = \begin{bmatrix} 2 & -3+(-7) \\ 4 & 5+(-3) \end{bmatrix} = \begin{bmatrix} 2 & -10 \\ 4 & 2 \end{bmatrix}$$

$$x = \frac{det(B_{1+c})}{det(B)} - 1 = \frac{-22}{22} - 1 = -2$$

$$y = \frac{det(B_{2+c})}{det(B)} - 1 = \frac{44}{22} - 1 = 1$$

Corollary 2

$$B_{1-c} = \begin{bmatrix} 2-(-7) & -3 \\ 4-(-3) & 5 \end{bmatrix} = \begin{bmatrix} 9 & -3 \\ 7 & 5 \end{bmatrix}$$

$$B_{2-c} = \begin{bmatrix} 2 & -3-(-7) \\ 4 & 5-(-3) \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}$$

$$x = 1 - \frac{det(B_{1-c})}{det(B)} = 1 - \frac{66}{22} = -2$$

$$y = 1 - \frac{det(B_{2-c})}{det(B)} = 1 - \frac{0}{22} = 1$$

Corollary 3

Row sum $= \begin{bmatrix} 2+(-3) \\ 4+5 \end{bmatrix} = \begin{bmatrix} -1 \\ 9 \end{bmatrix}$

Row sum + c = $B_c^\alpha = \begin{bmatrix} -1+(-7) \\ 9+(-3) \end{bmatrix} = \begin{bmatrix} -8 \\ 6 \end{bmatrix}$

$$B_{1+c}^\alpha = \begin{bmatrix} -8 & -3 \\ 6 & 5 \end{bmatrix} \quad B_{2+c}^\alpha = \begin{bmatrix} 2 & -8 \\ 4 & 6 \end{bmatrix}$$

$$x = \frac{\det(\mathrm{B}_{i+c}^{\alpha_1})}{\det(\mathrm{B}^{\alpha_1})} - 1 = \frac{-22}{22} - 1 = -2$$

$$y = \frac{\det(\mathrm{B}_{2+c}^{\alpha_2})}{\det(\mathrm{B}^{\alpha_2})} - 1 = \frac{44}{22} - 1 = 1$$

Corollary 4

Row sum $= \begin{bmatrix} 2+(-3) \\ 4+5 \end{bmatrix} = \begin{bmatrix} -1 \\ 9 \end{bmatrix}$

Row sum c = $B_c^\alpha = \begin{bmatrix} -1-(-7) \\ 9-(-3) \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$

$$B_{1-c}^\alpha = \begin{bmatrix} 6 & -3 \\ 12 & 5 \end{bmatrix} \quad B_{2-c}^\alpha = \begin{bmatrix} 2 & 6 \\ 4 & 12 \end{bmatrix}$$

$$x = 1 - \frac{\det(\mathrm{B}_{1-c}^{\alpha_1})}{\det(\mathrm{B}^{\alpha_1})} = 1 - \frac{66}{22} = -2$$

$$y = 1 - \frac{\det(\mathrm{B}_{2-c}^{\alpha_2})}{\det(\mathrm{B}^{\alpha_2})} = 1 - \frac{0}{22} = 1$$

It was shown that Corollary 2.2, Corollary 2.3, Corollary 2.4 and Corollary 2.5 are equal and are equivalent to Theorem 2.1. In fact, Corollary 2.3 and Corollary 2.5 are equal as well as Corollary 2.2 and Corollary 2.4. However, the computational cost of the methods increases as the order of the linear system increases notwithstanding, the highlighted methods especially for a non-integer

We shall attribute WZ, WZ_1, WZ_2, WZ_3 and WZ_4 factorization respectively for using Theorem 2.1, Corollary2.2, Corollary 2.3, Corollary 2.4 and Corollary 2.5 to solve the linear systems of the factorization. Now, for WZ factorization algorithm, we obtain the *(n-1)* the element of the *(i-1)th* and *(n-i+1)th* column of w-matrix by computing $w_{i,k}^{(k)}$ and $w_{i,n-k+1}^{(k)}$ from

$$\begin{cases} z_{k,k}^{k-1} w_{i,k}^{(k)} + z_{n-k+1,k}^{(k-1)} w_{i,n-k+1}^{(k)} = -z_{i,k}^{(k-1)} \\ z_{k,n-k+1}^{(k-1)} w_{i,k}^{(k)} + z_{n-k+1,n-k+1}^{(k-1)} w_{i,n-k+1}^{(k)} = -z_{i,n-k+1}^{(k-1)} \end{cases} \quad (11)$$

which update the elements of Z-matrix from

$$z_{i,j}^{(k)} = z_{i,j}^{(k-1)} + w_{i,k}^{(k)} z_{k,j}^{(k-1)} + w_{i,n-k+1}^{(k)} z_{i,n-k+1,j}^{(k-1)} \quad 12$$

and we then proceed similarly for the central submatrices of size (n-2k) and so on. Where $k = 1,2, \ldots \lfloor \frac{n}{2} \rfloor, i,j = k+1, \ldots, n-k$ and $z_{i,j}^{(k)} \in \mathbb{R}$. We can now re-write Equation (11) in matrix form as

$$\overbrace{\begin{bmatrix} z_{k,k}^{(k-1)} & z_{n-k+1,k}^{(k-1)} \\ z_{k,n-k+1}^{(k-1)} & z_{n-k+1,n-k+1}^{(k-1)} \end{bmatrix}}^{B} \begin{bmatrix} w_{i,k}^{(k)} \\ z_{i,n-k+1}^{(k)} \end{bmatrix} = \overbrace{\begin{bmatrix} -z_{I,k}^{(k-1)} \\ -z_{I,n-k+1}^{(k-1)} \end{bmatrix}}^{c} \quad (13)$$

If we apply Theorem 2.1 to derive W-matrix by computing $w_{i,k}^{(k)}$ and $w_{i,n-k+1}^{(k)}$

with respect to the first and second column B from equation (13), we obtain

$$w_{i,k}^{(k)} = \frac{\det(B_{1|c})}{\det(B)} \text{ and } w_{i,n-k+1}^{(k)} = \frac{\det(B_{2|c})}{\det(B)}, \quad (14)$$

When

$$det(B) = z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}$$

$$det(B_{1|c}) = z_{n-k+1,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} - z_{n-k+1,n-k+1}^{(k-1)} z_{i,k}^{(k-1)}$$

$$det(B_{2|c}) = z_{k,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{k,k}^{(k-1)} z_{i,n-k+1}^{(k-1)}$$

The complete MATLAB code of VWZ factorization is given in Listing 1.

Now, if we apply corollary 2.2 to compute $w_{(i,k)}^{(k)}$ and $w_{(i,n-k+1)}^{(k)}$ in Equation (13) . Then,

$$w_{i,k}^{(k)} = \frac{det(B_{1+c})}{det(B)} - 1 \text{ and } w_{i,n-k+1}^{(k)} = \frac{det(B_{2+c})}{det(B)} - 1,$$

When $$det(B) = z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)},$$

$$det(B_{1+c}) = z_{n-k+1,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}, - z_{n-k+1,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} + z_{k,k}^{(k-1)} z_{n-k+1,n-k+1}^{(k-1)}$$

$$det(B_{2+c}) = z_{k,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{k,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} z_{n-k+1,n-k+1}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}.$$

The *Wz1* factorization is the factorization obtained from using Corollary 2.2 and its MATLAB code for computing the elements of W -matrix is given in Listing 2.

Furthermore, if we apply corollary 2.3 top compute W $w_{i,k}^{(k)}$ and $w_{i,n-k+1}^{(k)}$ in Equation (13).

Then, $w_{i,k}^{(k)} = 1 - \frac{det(B_{1-c})}{det(B)}$ and $w_{i,n-k+1}^{(k)} = 1 - \frac{det(B_{2-c})}{det(B)}$ (16)

$$det(B) = z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}$$

$$det(B_{1-c}) = z_{n-k+1,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} + z_{k,k}^{(k-1)} z_{n-k+1,n-k+1}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}$$

Listing1:MATLABcodeofWZ factorization.

```
functionWZfactorization(B,W,Z)
%stepsofelimination -  from B  to  Z
B= input ( 'matrix B  = ' ) ;
n = size (B,1) ;
W =   zeros ( n ) ;
for   k  =   1 : ceil ( ( n - 1) / 2 )
        k2  =  n  -  k  +   1 ;
        determinant   =  B(k , k)*B(k2 , k2)  - B (k2 , k) *B(k, k2) ;
        if determinant   ==  0
        exitflag   =   0;
                     for   i1   =  k : k2
                          for   i2   =   i1 : k2
                     determinant= B(i1 ,k)*B(i2,k2)- B(i2,k)*B( i1 , k2 ) ;
                           ifdeterminant   ~=  0
                            disp('input matrix cannotbefactorizedto  Z- matrix' )
                                        tmp  =  B ( i1 , k : k2 ) ;
                                        B ( i1 , k : k2 )  =  B ( k , k : k2 ) ;
                                        B ( k , k : k2 )  =  tmp ;
                                        tmp  =  B ( i2 , k : k2 ) ;
                                        B ( i2 , k : k2 )  =  B ( k2 , k : k2 ) ;
                                        B ( k2 , k : k2 )  =  tmp ;
                                         exitflag   =   1 ;
                      break
                                  end
                          end
                 end
                 ifexitflag   ==  0
                         Z  =  B ;
                          return
                 end
         end
    %findingelementsofW
      %  To  compute    ith   tothe(n- 1)thelementof(i- 1)thcolumnofW
     W(k+1:k2- 1,k)=(B(k2,k)*B(k+1:k2- 1,k2)- B(k2,k2)*B(k+1:k2- 1,k))/determinant ;
      %  To  compute   ith   tothe(n- 1)thelementof   (n- i+1)thcolumnofW
     W(k+1:k2- 1,k2)=(B(k,k2)*B(k+1:k2- 1,k)- B(k,k)*B(k+1:k2- 1,k2))/ determinant ;
         for  m= 1 : n
          W( m, m) = 1 ;
          W( m, n+1- m) ;
      end
      %  updating  B
      B(k + 1 : k2 - 1 , k)  =   0;
      B(k + 1 : k2 - 1 , k2)  =   0 ;
    B(k+1:k2- 1,k+1:k2- 1)=B(k+1:k2- 1,k+1:k2- 1)+W(k+1:k2- 1,k)*B(k,k+1:k2- 1)+W(k+1:k2- 1,k2)*
        B(k2,k+1:k2- 1);
        Z  =  B ;
   end
```

Listing2:MATLABcodeofWZ$_1$ factorization.

```
% finding elements of W
 W(k+1:k2- 1,k)=B(k2,k)*B(k+1:k2- 1,k2)- B(k2,k2)*B(k+1:k2- 1,k)
                         - B(k2,k)*B(k,k2)+B(k2,k2)*B(k,k))/determinant)- 1;
 W(k+1:k2- 1,k2)=((B(k,k2)*B(k+1:k2- 1,k)- B(k+1:k2- 1,k2)*B(k,k)
                              - B(k2,k)*B(k,k2) +B(k2,k2)* B(k,k))/determinant)- 1;
```

$$det(B_{2-c}) = z_{k,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} - z_{k,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)} - z_{k,k}^{(k-1)} z_{n-k+1,n-k+1}^{(k-1)}$$

W Z2 factorization is the factorization obtained from using Corollary 2.3 where its MATLAB code for computing elements of W -matrix is given in Listing.

Besides, we can apply Corollary 2.4 to compute $w_{i,k}^{*(k)}$ $w_{i,n-k+1}^{*(k)}$ in equation (13) as

$$w_{i,k}^{(k)} = \frac{det(B_{1+c}^{\alpha_1})}{det(B^{\alpha_1})} - 1 \quad \text{and} \quad w_{i,n-k+1}^{(k)} = \frac{det(B_{2+c}^{\alpha_2})}{det(B^{\alpha_2})} - 1 \qquad (17)$$

$$det(B^{\alpha_1}) = det(B^{\alpha_2}) = -z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)}$$

$$det(B_{1+c}^{\alpha_1}) = z_{n-k+1,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} - z_{n-k+1,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)}$$

$$det(B_{2+c}^{\alpha_2}) = z_{k,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{i,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)}$$

More so Wz3 factorization is the factorization obtained from using Corollary 2.4 and its MATLAB code for computing elements of W -matrix is given in Listing 4.

Lastly, if we apply Corollary 2.5 to compute $w_{i,k}^{(k)}$ and $w_{i,n-k+1}^{*(k)}$ in Equation (13) then

$$w_{i,k}^{(k)} = 1 - \frac{\det(B_{1-c}^{\alpha_1})}{det(B^{\alpha_1})} \text{ and } w_{i,n-k+1}^{*(k)} = 1 - \frac{\det(B_{2-c}^{\alpha_2})}{det(B^{\alpha_2})}. \qquad (18)$$

$$det(B^{\alpha_1}) = det(B^{\alpha_2}) = -z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)}$$

$$det(B_{1-c}^{\alpha_1}) = z_{n-k+1,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{i,n-k+1}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}$$

$$det(B_{2-c}^{\alpha_2}) = z_{i,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} + z_{n-k+1,n-k+1}^{(k-1)} z_{k,k}^{(k-1)} - z_{k,n-k+1}^{(k-1)} z_{i,k}^{(k-1)} - z_{n-k+1,k}^{(k-1)} z_{k,n-k+1}^{(k-1)}.$$

Lastly, the Wz3 factorization is obtrained from using corollary 2.5 where it's MATLAB code for computing element of

Our analysis shows that the matrix norms of *WZ, WZ1, WZ2, WZ3,* and *Wz4* factorization are only influenced by the architecture of the algorithm used. The *WZ* factorization has the worst algorithm for matrix norm because the accuracy of our algorithms based on the relative residual depends more on the Frobenius norm than the matrix size. The matrix norms of all the factorizations increase as the size of their matrices increase. *WZ4* factorization is about 15% better than *WZ* factorization.

3. Conclusion

The advantage of variations Cramer's rule in *WZ* factorization is to minimize round-off error. The methods produce better matrix norms than classical Cramer's rule in the factorizations via MATLAB.

Listing 3: MATLAB code of WZ_2 factorization

```
%finding elementsof W
W(k+1:k2− 1,k)=1− ((B(k2,k2)∗B(k+1:k2− 1,k) − B(k2,k)∗B(k+1:k2− 1,k2)
                    +B(k2,k2)∗B(k,k)− B(k2,k)∗B(k,k2))/determinant);
W(k+1:k2− 1,k2)=1− (B(k+1:k2− 1,k2)∗(B(k,k)− B(k,k2)∗B(k+1:k2− 1,k)
                         − B(k2,k)∗B(k,k2)+B(k2,k2)∗B(k,k))/determinant)
```

Listing 4: MATLAB code of WZ_3 factorization

```
%finding elementsof W
W(k+1:k2− 1,k)=((B(k2,k)∗B(k+1:k2− 1,k2)− B(k2,k2)∗B(k+1:k2− 1,k)
                      − B(k2,k)∗B(k,k2)+B(k2,k2)∗B(k,k))/determinant)− 1;
W(k+1:k2− 1,k2)=((B(k,k2)∗B(k+1:k2− 1,k)− B(k+1:k2− 1,k2)∗B(k,k)
                        − B(k2,k)∗B(k,k2) +B(k2,k2)∗B(k,k))/determinant)− 1;
```

Listing 5: MATLAB code of WZ_4 factorization

```
%finding elementsof W
W( k + 1 : k2 −1 , k ) = 1 − ( (B ( k2 , k2 ) *B ( k + 1 : k2 −1 , k ) +B ( k2 , k2 ) *B ( k , k )
                    −B ( k2 , k ) *B ( k + 1 : k2 −1 , k2 )  −B ( k2 , k ) *B ( k , k2 ) ) / determinant) ;
W( k + 1 : k2 −1 , k2 ) =1 −(B ( k + 1 : k2 −1 , k2 ) * ( B ( k , k ) +B ( k2 , k2 ) *B ( k , k )
                        −B ( k , k2 ) *B ( k + 1 : k2 −1 , k )−B ( k2 , k ) *B ( k , k2 ) ) / determinant) ;
```

Table 1: Norms of + , + 1 , + 2 , + 3 and + 4 factorization on MATLAB R2018b.

Matrixname	Matrixsize	−)	!− /1	!− /2	!− /3	!− /4
*Trefethen*_500	500	1.98E-20	1.73E-20	1.14E-20	0.96E-20	0.98E-20
*tu*b1000	1,000	2.88E-20	2.21E-20	1.79E-20	1.57E-20	1.72E-20
comsol	1,500	3.85E-20	3.43E-20	2.94E-20	2.13E-20	2.19E-20
*olm*2000	2,000	6.62E-20	6.30E-20	6.03E-20	5.32E-20	5.31E-20
*cryg*2500	2,500	8.51E-20	8.22E-20	7.27E-20	6.35E-20	6.36E-20
*nasa*2910	2,910	9.59E-20	9.28E-20	8.96E-20	8.76E-20	8.68E-20
thermal	3,456	1.02E-19	0.88E-19	0.85E-19	0.80E-19	0.78E-19
*ACTIVSg*2000	4,000	2.57E-19	2.29E-19	2.08E-19	1.97E-19	1.90E-19
*bcsstk*28	4,410	3.51E-19	3.32E-19	3.11E-19	2.96E-19	2.79E-19
*rdb*5000	5,000	3.32E-19	3.12E-19	2.88E-19	2.73E-19	2.60E-19

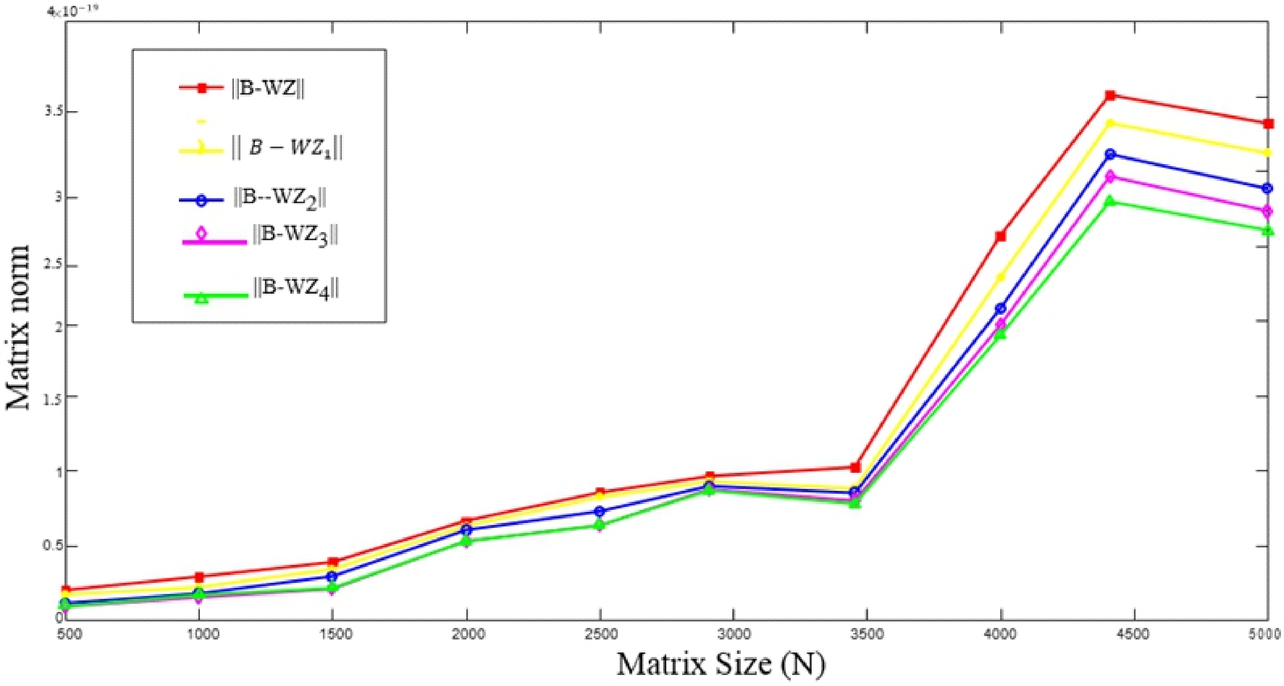

Figure 1: Matrix norms of LU, WZ, VWZ, $Wm1\,Zm1$ and $Wm2\,Zm2$ factorization on MATLAB R2018b.

REFERENCES

Babarinsa, O., & Kamarulhaili, H. (2017). Modification of Cramer's rule. *Journal of Fundamental and Applied sciences, 9* (5), 556-567.

Babarinsa, O., & Kamarulhaili, H. (2019). Modified Cramer’s Rule and its Application to Solve Linear Systems in WZ Factorization. *Matematika, 35*(1), 25-38.

Brunetti, M., & Renato, A. (2014). Old and New Proofs of Cramer’s Rule History, notations and tools. *Applied Mathematical Sciences, 8*(133), 6689-6697.

Bylina, B. (2018). The block WZ factorization. *Journal of Computational and Applied Mathematics, 331*, 119-132.

Bylina, B., & Bylina, J. (2009). Influence of preconditioning and blocking on accuracy in solving Markovian models. *International Journal of Applied Mathematics and Computer Science, 19*(2), 207-217.

Dunham, C. B. (1980). Cramer's rule reconsidered or equilibration desirable. *ACM SIGNUM Newsletter, 15*(4), 9-9.

Efremides, O., Bekakos, M., & Evans, D. J. (2002). Implementation of the generalized WZ factorization on a wavefront array processor. *International Journal of Computer Mathematics, 79*(7), 807-815.

Evans, & Hatzopoulos. (1979). A parallel linear system solver. *International Journal of Computer Mathematics, 7*(3), 227-238.

Evans, D. J. (2002). The QIF singular value decomposition method. *International Journal of Computer Mathematics, 79* (5), 637-645.

Evans, D. J., & Oksa, G. (1997). Parallel solution of symmetric positive definite Toeplitz systems. *Parallel Algorithms and Applications, 12*(4), 297-303.

Habgood, K., & Arel, I. (2012). A condensation-based application of Cramer's rule for solving large-scale linear systems. *Journal of Discrete Algorithms, 10*, 98-109. doi:10.1016/j.jda.2011.06.007

Heinig, G., & Rost, K. (2004). Split algorithms for skewsymmetric Toeplitz matrices with arbitrary rank profile. *Theoretical computer science, 315*(2), 453-468.

Heinig, G., & Rost, K. (2005). Schur-type algorithms for the solution of Hermitian Toeplitz systems via factorization *Recent advances in operator theory and its applications* (pp. 233-252): Springer.

Hogben, L. (2007). *Handbook of linear algebra*. Boca Raton: Chapman & Hall/CRC.

Lipschutz, S., Lipson, M., & Lipschutz, S. (2009). *Linear algebra*. New York: McGraw-Hill.

Moler, C. (1974). Cramer's rule on 2-by-2 systems. *ACM SIGNUM Newsletter, 9*(4), 13-14.

Rao, S. C. S. (1997). Existence and uniqueness of WZ factorization. *Parallel Computing, 23*(8), 1129-1139.

Rhofi, M. K., & Ameur, A. R. (2016). Double power method iteration for parallel eigenvalue problem. *International Journal of Pure and Applied Mathema tics, 108*(4), 945-955.

Ufuoma, O. (2013, 25 october). *A New and Simple Method of Solving Large Linear Systems : Based on Cramer's Rule but Employing Dodgson's Condensation.* Paper presented at the Proceedings of the World Congress on Engineering and Computer Science, San Francisco.