Partial Reconfiguration and Novel Algorithm for Matrix Inverse Computations

Etienne Aubin Mbe Mbock *

***Abstract*—Partial reconfiguration of algorithms is becoming increasingly attractive in many computational applications. This research article covers two aspects of the reconfiguration approach: The first aspect shows that partial reconfiguration is capable of reconstructing computations. The second aspect will construct a theoretical hardware device that realises these computations. With this research article, we analyse the importance of partial reconfiguration for algorithms in one hand and in the second hand we use and apply this concept for the invention of a method that computes two matrices that are inverses of each other. In this paper we specify the computation of two inverse upper and lower matrices using the partial dynamic reconfigurabilty concept. We propose for this novel algorithm a pseudo code implementation and its hardware construction.**

***Index Terms*—Partial, Reconfiguration, Linear, Matrix Recursion, Hardware.**

I. Introduction

PARTIAL reconfiguration computing is nowadays a subject that is establishing itself as a basic scientific field. This concept is especially developed in computing sciences and computer technology [2], [3], [6]. It covers various classical subjects including Xilinx technologies, FPGA's construction, migration towards control theory with the implementation of the Kalman Filter on FPGA, algorithm analysis with the creation of the recursive dynamic process and engineering fields with algorithm optimization. Reconfiguration in hardware implies the dynamic modification of blocks of logic by downloading partial bit files while the remaining logic continues to operate without interruption. The Partial Reconfiguration technology allows designers to change functionalities of their hardware devices. As a consequence, computer technology designers will migrate to devices with lower hardware complexity. Because the benefits of this hardware definition are various including among other performance development, hardware complexity reduction from the technological point of view. It becomes interesting to analyse and apply these features in computing and algorithm creation. For this research article we suppose that we can partially reconfigure any algorithm. This reconfiguration is guaranteed by theorems on dynamic partial reconfiguration of algorithms. We will admit that the general recursive linear, specified by:

$$\left\{q_1,\quad q_j=\sum_{i=1}^{j-1}\alpha_{ji}q_i,\quad j\in\{2,3,\cdots,N\}\right\}$$

is given. All matrix operations are performed on two intial $n\times n$ matrices A, B with entries

$$\left\{a_{i,j},\ 1\le i\le n,\ 1\le j\le n\right\}$$

these matrices will be considered as n-length column vectors, represented by:

$$A:=\begin{bmatrix}A_1 & A_2 & \cdots & A_n\end{bmatrix}=\left(A_j,\ j\in\{1,2,\cdots,n\}\right)$$

$$B:=\begin{bmatrix}B_1 & B_2 & \cdots & B_n\end{bmatrix}=\left(B_j,\ j\in\{1,2,\cdots,n\}\right).$$

This research article will provide the advantages of reconfiguration of algorithms, and with these features of reconfiguration we demonstrate the construction of an algorithm. This novel algorithm solves the matrix inversion construction problem. That is, the given two matrices A, B represented by its respective column entries

$$\left(A_j,\ B_j,\ j\in\{1,2,\cdots,n\}\right),$$

construct with a partial reconfiguration two matrices that are inverse to each other. We state the theorem, demonstrate it and propose an algorithm for the matrix inverses. In addition to this description, we construct the related hardware to this algorithm for future FPGA implementations. This research article assumes some matrix analysis and computation basic results [11]–[16]. We suppose that the experimentation in this research article takes place in a n-dimensional vector space. The aim of this research article, is the construction of the matrix inverse algorithm and demonstrate that the algorithm performs correctly. The inverted matrix will have the following expansion

$R_{1,1}$	$R_{1,2}$	$\cdots$	$R_{1,k}$	$\cdots$	$R_{1,n}$
0	$R_{2,2}$	$\cdots$	$\cdots$	$\cdots$	$R_{2,n}$
$\vdots$	0	$\ddots$	$\ddots$	$\ddots$	$\vdots$
$\vdots$	$\vdots$	$\ddots$	$R_{k-1,k-1}$	$\cdots$	$R_{k-1,n}$
$\vdots$	$\vdots$	0	$\ddots$	$\ddots$	$\vdots$
0	0	$\cdots$	0	0	$R_{n,n}$

In addition to the proposed algorithm, a theoretical construction of the hardware for the proposed algorithm will be provided.

* Ruprecht-Karls Universität Heidelberg, PHD-Candidate at the Faculty of Mathematics and Computer Sciences, Lecturer at the Applied University of Furtwangen Robert Gerwig Platz 1, 78120, Furtwangen, Schwarzwald, Germany, Email: mbe@hs-furtwangen.de. Mailing Adress: Eichendorffstrasse 35, 78120 Furtwangen, Germany

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Advantages of Reconfiguration of Algorithms. A Software Point of View

Algorithms that are nowadays software are complex because in a system in which they are to be integrated are increasing, for the simple case of the Kalman Filter Software. If we consider a sequence of n estimations problems p_i, $i \in \{1, 2, \cdots, n\}$. The partial reconfiguration of the software will adapt the the Kalman Filter algorithm to each p_i, $i \in \{1, 2, \cdots, n\}$. In this way the partial reconfiguration will not only optimize the Kalman Filter algorithm, but also it creates a new strategy for problem estimation using the software. The software will now have a new functionality depending upon the problem to be solved. This means that partial reconfiguration aims at algorithm optimization in terms of the complexity of the algorithm and its flexibility. This will still have a consequence on the hardware construction. If the algorithm is now optimized per reconfiguration then the constructed hardware will be cheaper for the designer, space, cost and power consumption reduction. These points are general when it comes to partial reconfiguration. In this research we want to give some concrete improvement base on our research:

1) The application of this concept in the special case of the Kalman Filter extends the method with the second condition for optimality, "Six Valid Error Estimation and the Stop Mod reconfiguration ", "Three Optimal Error Covariance Matrix and Stop Mod ", "Three Computation of the Error Covariance and the Stop Mod "
2) Linear Recursive Process Creation and Optimization
3) Matrix [Q,R]-Decomposition
4) Reconfiguration Speed Problem Resolution
5) Real Vectors Coding

The improvements that we listed are more related to algorithms and computational optimization, which has also been applied in robotics [17], [18]. The idea of partial reconfiguration of algorithms aims at algorithms functionalities extension and this in turn still alters the true hardware complexity of the algorithm. The previous cited advantages of partial reconfiguaration of algorithms constitutes the background we need to present the novel algorithm for matrix inverse computations.

III. Principles and Theorems

Theorem III.1. *Under consideration of the existence of the recursive dynamic process, there exists two matrices that are upper and lower matrices and inverse to each other.*

Proof:

1) Assuming that the recursive dynamic process exist, we can construct the recursive linear process algorithm.
2) Given a $n \times n$ matrix A. There is a $n \times n$ matrix B associated to the matrix A. The process computes the following entries $R_{i,j}$, the process being a linear recusive process will not compute the $R_{i,j}$. We then suppose that $R_{i,j} = 0$ for $j \in \{0, 1, 2, \cdots, n-1\}$ a recursive vector will exist and we denote it by $V_j^{(k)}$ a vector of length n. The vector $V_j^{(k)}$ is related to the entries $R_{i,j}$ according to the following scheme:
$$V_j^{(k)} = V_j^{(k-1)} - R_{k,j} \cdot V_j^{(k)}$$
for all $j \in \{1, 2, \cdots, n\}$
3) The matrice R with entries $R_{i,j}$ will be computed from the input column matrix A according to the following scheme $R_{k,j} = \begin{cases} V_j^{(k)} \cdot A_j & if \quad k \neq j \\ \left\| V_j^k \right\| & if \quad k = j \end{cases}$
4) The commutative product of the matrix R and the column vectors $V_j^{(k)}$ is the identity matrix and completes the proof.

■

A. Construction of the Algorithm and Computations

Assuming that the $n \times n$ matrices A are given, and applying theorem III.1, we then propose the following algorithm that will compute two matrices that are inverse to each other. The proof provided above summarized in four steps in which the values of the matrix V are computed according to the recursive formular $V_j^{(k)} = V_j^{(k-1)} - R_{k,j} \cdot V_j^{(k)}$ and the entries of the matrix R are all the values $R_{i,j}$ computed in step three of the proof above. We will initialise the matrix R to zero.

Algorithm III.2.

function PROBLEM(A, B)
  $[m, n] \leftarrow size(A)$
  $[p, q] \leftarrow size(B)$
  $R \leftarrow zeros(n, n)$
  if $m = p$ *or* $q = n$ ***then***
    $R \leftarrow zeros(n, n)$
    for $j = 2, 3, \cdots, n$ ***do***
      $V(:, j) = B(:, j)$
      for $i = 1 : j - 1$ ***do***
        $R(i, j) \leftarrow V(:, i)^t * A(:, j)$
        $V(:, j) \leftarrow$
        $V(:, j) - R(i, j) * V(:, i)$
      end for
      $R(j, j) \leftarrow \|V(:, j)\|$
      $V(:, j) = \frac{V(:,j)}{R(j,j)}$
    end for
  end if
end function

The proposed algorithm computes in the case of the Hilbert matrix of order 3 in the following way:

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1) $R_{1,1} = \|B_1\| = 1$ and $V_1^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
2) $R_{1,2} = V_{(1)}^{(1)} \cdot A_2$ and $V_2^{(1)} = V_2^{(0)} - R_{1,2} \cdot V_1^{(1)}$, $R_{2,2} = \left\| V_2^{(1)} \right\|$ and $V_2^{(2)} := \frac{V_2^{(1)}}{\left\| V_2^{(1)} \right\|}$
3) $R_{1,3} = V_{(1)}^{(1)} \cdot A_3$
4) $V_3^{(1)} = V_3^{(0)} - R_{1,3} \cdot V_1^{(1)}$ and define $V_3^{(1)} := \frac{V_3^{(1)}}{\left\| V_3^{(1)} \right\|}$
5) $R_{2,3} = V_{(2)}^{(2)} \cdot A_3$, $V_3^{(2)} = \frac{V_3^{(1)}}{\left\| V_3^{(1)} \right\|} - V_{(2)}^{(2)} \cdot A_3 \cdot V_2^{(2)}$ and $R_{3,3} = \left\| V_3^{(2)} \right\|$
6) $V_3^{(3)} := \frac{V_3^{(2)}}{\left\| V_3^{(2)} \right\|}$

IV. Reconfigurations Analysis

The recursive linear process algorithm is given, we can provide as follows the reconfiguration analysis of this method. This will consider the following algorithm's parts:

1) Inputs parameters, $m \times n$ matrices Alpha and Beta
2) The body of the linear recursive process

The analysis of reconfiguration of the linear recusive process algorithm aims at optimizing the process towards the construction of two matrices that are inverse to each other. Considering the first part for reconfiguration, we postulate the following principles and facts.

The matrix Gama and Beta of the must be reconfigured into square matrices of the same size. In this paper reconfigure the matrix Beta to the identity matrix.

Principle IV.1. Parameters Reconfiguration

The starting index $j \in \{2, 3, \cdots, n\}$ of the recursive linear algorithm will reconfigure in $j \in \{2, 3 \cdots, n\}$, $Gama_j$ will reconfigure to $V_j^{(0)}$, The index i will not reconfigure, $R_{i,j}$ will reconfigure to the scalar product $V_i \cdot A_j$, $Gama_j$ will reconfigure to a normed vector

Principle IV.2. Body Reconfiguaration of the Algorithm

The reconfiguration steps are finite, the algorithm converges and compute the expected matrix decomposition

Principle IV.3. Termination of the Algorithm

V. General Computations Presentation for the Inverses Matrix Computations

The algorithm can be performed in two ways, we propose with the following presentation a combination of of top-down and up-down computation model. The second method will be recursive starting from $V_1^{(1)}$. The figures that will next follow give numerical values in the special case of the Hilbert matrix

1) Write $\left\{ V_i^{(j)}, R_{j,i} \right\}$, for $j \in \{1, 2 \cdots, n\text{-}1\}$ and $i \in \{n, n\text{-}1, \cdots, 1\}$.
2) Compute $V_1^{(1)}$.
3) Compute $\left\{ V_i^{(j)}, R_{j,i} \right\}$, for $j \in \{1, \cdots, i\text{-}1\}$ and $i \in \{1, 2, \cdots, n\}$.

The figure that follows describes the computional presentation of the algorithm

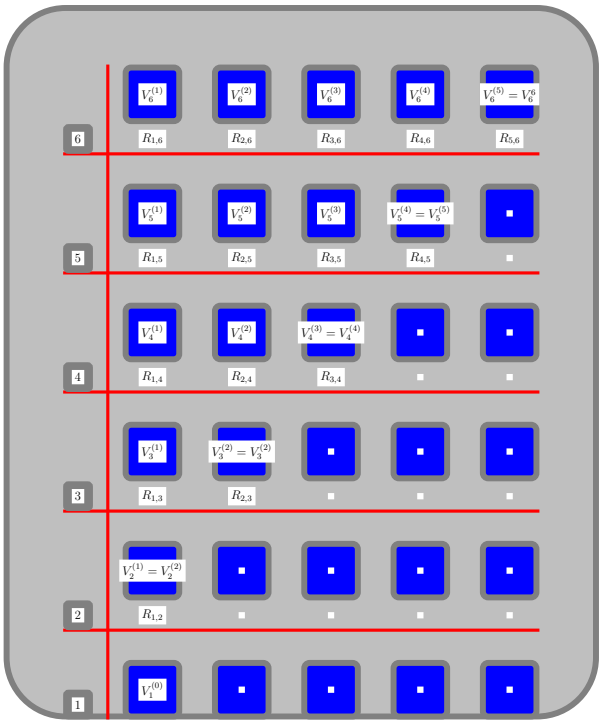

Fig. 1
Presentation of the Matrix Inverse Computations

The numerical values of the algorithm in the special case of a Hilbert $n \times n$ matrix gives the following values of the matrix V and R. The third column vector of the matrix V corresponds to V_3^3 its formula is given by the following

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

vector

$$\begin{pmatrix} \frac{-a_{13}}{\left\|V_3^{(1)}\right\|\left\|V_3^{(2)}\right\|} - \frac{a_{12}^2 \cdot a_{13}}{\left\|V_1^{(1)}\right\|^2\left\|V_3^{(2)}\right\|} + \frac{a_{23} \cdot a_{12}}{\left\|V_2^{(1)}\right\|\left\|V_3^{(2)}\right\|\left\|V_1^{(1)}\right\|} \\ \frac{a_{12}^2 \cdot a_{13}}{\left\|V_2^{(1)}\right\|^2\left\|V_3^{(2)}\right\|} - \frac{a_{23}}{\left\|V_2^{(1)}\right\|^2\left\|V_3^{(2)}\right\|} \\ \frac{1}{\left\|V_3^{(1)}\right\|^2\left\|V_3^{(2)}\right\|} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The matrix entry $R(3,4)$ is given by the following formular:

$$\begin{aligned} R_{3,4} = & \left(\frac{-a_{13} \cdot a_{14}}{\left\|V_1^{(3)}\right\| \cdot \left\|V_3^{(2)}\right\|} - \frac{a_{12}^2 \cdot a_{13} \cdot a_{14}}{\left\|V_2^{(1)}\right\|^2 \cdot \left\|V_3^{(2)}\right\|} \right) \\ & + \left(\frac{a_{23} \cdot a_{12} \cdot a_{14}}{\left\|V_2^{(1)}\right\|^2 \cdot \left\|V_3^{(2)}\right\|} + \frac{a_{12} \cdot a_{13} \cdot a_{24}}{\left\|V_2^{(1)}\right\|^2 \cdot \left\|V_3^{(2)}\right\|} \right) \\ & \left(- \frac{a_{23} \cdot a_{24}}{\left\|V_2^{(1)}\right\|^2 \cdot \left\|V_3^{(2)}\right\|} - \frac{a_{34}}{\left\|V_3^{(1)}\right\|\left\|V_3^{(2)}\right\|} \right) \end{aligned} \quad (1)$$

1.0000	0.5000	0.3333	0.2500	0.2000	0.1667	0.1429
0	1.1180	0.0745	0.0671	0.0596	0.0532	0.0479
0	0	1.0462	0.0749	0.0686	0.0626	0.0573
0	0	0	1.0235	0.0644	0.0598	0.0555
0	0	0	0	1.0147	0.0545	0.0512
0	0	0	0	0	1.0106	0.0468
0	0	0	0	0	0	0.0083

Fig. 2
Numerical Values of the Algorithm, Computation of R

The following two matrices that represent V and R are inverses of each other

1.0000	−0.4472	−0.2868	−0.1940	−0.1391	−0.1046	−0.0815
0	0.8944	−0.0637	−0.0540	−0.0448	−0.0376	−0.0319
0	0	0.9559	−0.0699	−0.0602	−0.0518	−0.0450
0	0	0	0.9770	−0.0620	−0.0545	−0.0481
0	0	0	0	0.9855	−0.0532	−0.0476
0	0	0	0	0	0.9895	−0.0459
0	0	0	0	0	0	0.9918

Fig. 3
Numerical Values of the Algorithm, Computations of V

$$V = \begin{bmatrix} 1 & -\frac{a_{12}}{||V_2^{(1)}||} & & \cdots & \cdots & \\ 0 & \frac{1}{||V_2^{(1)}||} & & & & \vdots \\ & & \frac{1}{||V_3^{(1)}|| \cdot ||V_3^{(2)}||} & & & \vdots \\ \vdots & 0 & \vdots & \ddots & & \\ & \vdots & 0 & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & \frac{1}{||V_n^{(1)}|| \cdot ||V_n^{(2)}|| \cdots ||V_n^{(n-1)}||} \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & a_{12} & & \cdots & \cdots & a_{1n} \\ 0 & ||V_2^{(1)}|| & & & & V_2^{(2)} \cdot A_n \\ 0 & & ||V_3^{(1)}|| \cdot ||V_3^{(2)}|| & \vdots & \cdots & \vdots \\ \vdots & 0 & \vdots & \ddots & \cdots & \vdots \\ 0 & \vdots & 0 & & \ddots & V_{n-1}^{(n-1)} \cdot A_n \\ 0 & 0 & \cdots & \cdots & 0 & ||V_n^{(1)}|| \cdots ||V_n^{(n-1)}|| \end{bmatrix}$$

These two matrices provide the same results as in the figures 2 and 3.

VI. Hardware Construction

The construction of the hardware for this algorithm will be recursive. The process is summarized in the following steps:

1) $V_1^{(0)}$ and construct $R_{1,2}$
2) $V_1^{(2)}$ and construct $R_{1,3}$ $R_{2,3}$
3) $V_3^{(3)}$ and construct $R_{1,4}$ $R_{2,4}$ $R_{2,4}$ $\cdots$

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

4) $V_{n-1}^{(n-1)}$ and construct $R_{1,n}$ $R_{2,n}$ $\cdots$ $R_{n-1,n}$
5) $V_n^{(n)}$

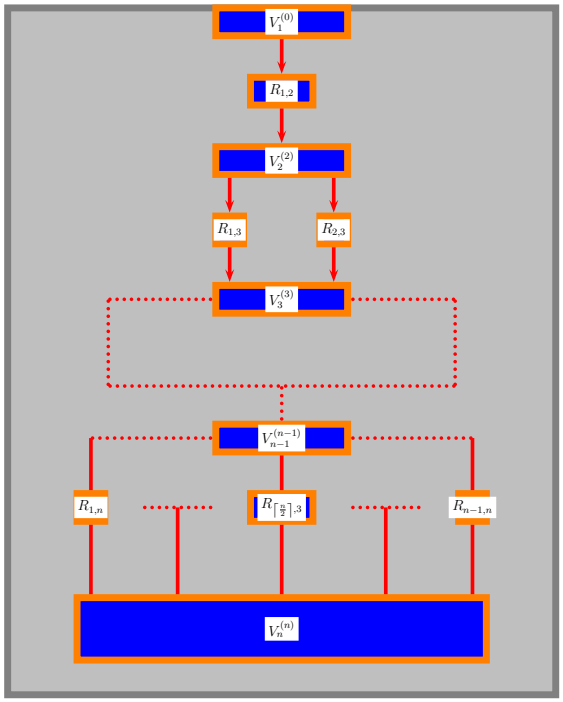

Fig. 4
Hardware Construction

The proposed hardware will not be trivial for FPGA realisation.

VII. Conclusion

This research article points at partial reconfiguration and matrix inverse computation. The mathematical analysis of this paper and the proposed hardware will be a hard tast for FPGA realisation. This algorithm is new and is based on the recursive dynamic process. Although the concept of partial reconfiguration will address hardware near systems. The general approach in this research article in this paper is numerical [19]–[27], in particular the computations presented in section five. The inverse matrix computations from the linear recursive process, its theoretical construction in hardware will have the following impact:

1) Creations of real values hardware instead of integer valued hardware
2) Solve the generalised inverses matrix problem
3) Coding matrices

From the previous resulting research, the inverse computations particular result will serve computer algorithms, particularly numerical matrix based computations. The hardware construction of this algorithm is still open and will have a great research impact on real valued hardware creation. The analyses developed in this paper will be significant for computer scientists and computer engineers.

References

[1] Xilinx, Inc, "http://www.xilinx.com/support documentation/white-papers/wp374-partial- reconfig-xilinx-FPGAs.pdf," 2012.
[2] Osterloh, Michalik, Habinc, Fiethe, "Dynamic Partial Reconfiguration in Space Applications," *Comference Publication,* pp 336-343, 2009.
[3] A. Donato, F. Ferrandi, M. D. Santambrogio, and D. Sciuto, "Operating System Support for Dynamically Reconfigurable Soc Architectures," *ieee International Soc Conference,* pp 233-238,2005.
[4] E.A. Mbock, "fpga Implementation of the Kalman Filter:" *http://www.dpg-verhandlungen.de, karlsruhe,* 2011.
[5] C. Kao, "Benefits of Partial Reconfiguration," *Xcell Journal, fourth quarter,* pp 65-68, 2005.
[6] Upegui, "http://lslwww.epfl.ch/ upegui/docs/DPR.pdf."
[7] Xilinx, Inc *http://www.xilinx.com/support/ documentation/ white-papers/374-partial-reconfig-xilinx-fpgas.pdf,* 2012.
[8] S. Corbetta, F. Ferrandi, M. Morandi, M. Novati, M.D. Santambrogio, D. Sciuto "Two Novel Approaches to Online Partial Bitstream Relocation in a Dynamically Reconfigurable System," *Proc. of ieee Computer Society Annual Symposium on vlsi,* pp. 457-458, may 2007.
[9] E.A. Mbock, "Algorithms in Matlab, the Reality of Abstraction and the Power of Pseudocodes," *Optimus Verlag,* 2012.
[10] J.W. Demmel, "Applied Numerical Linear Algebra," *Siam, Philadelphia, pa,* 1997.
[11] G.H. Golub and Ch.F. Van Loan, "Matrix Computations," *second edition, Johns Hopkins University press, Baltimore, md,* 1989.
[12] Forsythe, George E., M. A. Malcolm, and C. B. Moler, "Computer methods for Mathematical Computations," *Prentice Hall, Englewood Cliffs, nj,* 1977.
[13] M.T. Heath, "Scientific Computing: an Introductory Survey," *Mcgraw-hill, Boston, ma,* 1997.
[14] R.L. Burden, J.D. Faires, "Numerical analysis," *Brooks/Cole Publishing Company,* 1997-2001.
[15] N.J. Higham, "Accuracy and Stability of Numerical Algorithms," *Siam, Philadelphia, pa,* 1996.
[16] J.W. Demmel, "Applied Numerical Linear Algebra", *Siam, Philadelphia, pa,* 1997.
[17] Meng, Y., "An agent-based mobile robot system using configurable soc technique." *Proceedings of the 2006 ieee International Conference on Robotics and Automation, Orlando, Florida,* 33683373, 2006.
[18] Long, M., Gage, A., Murphy, R., Valavanis, K, "Application of the Distributed Field Robot Architecture to a Simulated Demining Task." *Proceedings of the ieee International Conference on Robotics and Automation, Barcelona, Spain,* pp. 31933200, 2005.
[19] G.H. Golub and Ch.F. Van Loan, "Matrix computations," *second edition, Johns Hopkins University Press, Baltimore, md,* 1989.
[20] M.T. Heath, "Scientific Computing: an Introductory Survey", *Mcgraw-hill, Boston, ma,* 1997.
[21] N.J. Higham, "Accuracy and Stability of Numerical Algorithms," *Siam, Philadelphia, pa,* 1996.
[22] R.D. Skeel and J.B. Keiper, "Elementary numerical Computing with Mathematica," *Mcgraw-hill, New York, ny,* 1993.
[23] B.D. Hahn, "Essential Matlab for Scientists and Engineers," *John Wiley Sons, New York, ny,* 1997.
[24] D.R. Hill and D.E. Zitarelli, "Linear Algebra Labs with Matlab," *second edition, Prentice Hall, Upper Saddle River, nj,* 1996.
[25] R.E. Larson and B.H. Edwards, "Elementary Linear Algebra," *third edition, D.C. Heath and Company, Lexington, ma,* 1996.
[26] S.J. Leon, "Linear Algebra with Applications," *fifth edition, Prentice Hall, Upper Saddle River, nj,* 1998.
[27] G. Strang, "Linear Algebra and its Applications," *second edition, Academic Press, Orlando, fl,* 1980.

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)