ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 5, November 2015

Cramer-Gonzalez: A New Algorithm to Solve Linear System of Equations using cuBLAS on GPU

González, H.E, Carmona L, J.J.
[1] Information Technology Department, ININ, Carretera México-Toluca s/n, C.P. 52750, La Marquesa, Ocoyoacac, México.

***Abstract*—In this paper a parallel code for solving simultaneous linear equations with real coefficients are used. We postulate a new linear transformation in matrix product form and we apply this linear transformation to a matrix (A) and next we apply again this linear transformation to a vector (b) and so we obtain Cramer Rule solution of the linear system of equations in efficient way. We use subroutines of cuBLAS 2^{nd} and 3^{rd} levels in double precision and we obtain correct numeric results.**

***Index Terms*—Cramer-Gonzalez Method, Linear System of Equations, Cramer´s Rule, Adjugate Matrix Method, Non-Structure Dense Matrix, cuBLAS on GPUs.**

I. INTRODUCTION

A Linear System of Equations (**LSE**) can be defined as a set of *m* equations with *n* unknowns represented by a matrix **A**, a vector **b** and an unknown vector **x**, namely, $\mathbf{Ax}=\mathbf{b}$ .Many methods have been proposed to solve such linear equations. A famous one is Cramer's rule, where each component of the solution is determined as the ratio of two determinants. When trying to solve a system of *n* equations using Cramer's rule, one needs to compute n+ 1 determinant, each of order *n*. If these are computed in a straightforward way, using the Laplace Expansion, the solution to the linear system takes $(n+1)(n!)(n-1)$ multiplications, plus a similar number of additions. Although Cramer´s rule possesses a fundamental theoretical importance, it may result impractical in computations. It is for that reason that this method is seldom recommended [1]-[6]. Cramer´s rule has at least one attractive property: it computes every element of the solutions independently. For this reason, it can be a practical method for some special linear systems on parallel computers [7]. Another approach, with a certain mathematical appeal but considerable computational pitfalls, finds the solution to a linear system of equations using the inverse matrix $\mathbf{A}^{-1}$. However, in virtually every application, it is unnecessary and inadvisable to compute the inverse matrix explicit. The inverse requires more arithmetic and produces a less accurate answer. Therefore, neither of the above methods are recommended [8].

II. GAUSS TRANSFORMATION (GT)

There are several ways to make the process of elimination of elements in an array for triangulation [9], when the matrix **A** is square, dense and non-structured; the **GT** is the best in most practical cases. We take this notation for explain the proposed new linear transformation and from the decomposition of gauss derive the new transformation [10]. Assume:

$$x \in R^n \text{ with } x_k \neq 0$$

If $\alpha_i = \frac{x_i}{x_k} \forall i = k+1,\ldots,n \quad (1)$ and

$$M_k = I - \alpha e_k^T \ \therefore \alpha = \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \alpha_{k+1} \\ . \\ . \\ . \\ \alpha_n \end{pmatrix} \quad (2)$$

Then:

$$M_k \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ x_{k+1} \\ . \\ . \\ . \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ . \\ . \\ . \\ x_k \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \quad (3)$$

The matrix M_k is said to be a GT and has the numeric interesting property following:

$$M_k = I - \alpha e_k^T \quad (4)$$

The inverse matrix is obtained easily by changing the sign:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 5, November 2015

$$M_k = I + \alpha e_k^T \quad (5)$$

So if:

$$M = M_k \cdots M_1 \quad (6) \Rightarrow \quad M_i = I_n - \alpha^{(i)} e_i^T \quad (7)$$

$$\alpha^{(i)} = \begin{pmatrix} 0 \\ . \\ . \\ . \\ 0 \\ -l_{i+1,i} \\ . \\ . \\ . \\ -l_{ni} \end{pmatrix} \quad (8)$$

Follows that **M** is a lower triangular matrix (*nxn*) that is usually expressed as a factor that completely, that is, for **k <n**:

$$\left(-\alpha^{(1)}, -\alpha^{(2)}\right) = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ . & . \\ . & . \\ . & . \\ l_{n1} & l_{n2} \end{pmatrix}; k = 2 \quad (9).$$

The matrix **M** represented in this way has the advantage that is equivalent to storing the first **k** columns $M^{-1} = M_1^{-1} \cdots M_k^{-1}$, such that if the numerical property applies reverse holds, we have:

$$M_i^{-1} = I + \alpha^{(i)} e_i^T \quad (10);$$

$$e_j^T \alpha^{(i)} = 0 \forall j < i \quad (11) \Rightarrow$$

$$M^{-1} = (I + \alpha^{(1)} e_1^T) \cdots (I + \alpha^{(k)} e_k^T) = I + \sum_{i=1}^{k} \alpha^{(i)} e_i^T \quad (12)$$

III. MATRIX LU DECOMPOSITION

Assume: $A \in R^{mxn}$ and for some $k<min\{m,n\}$, **GT** is determined as follows: $M_1, \ldots, M_{k-1} \in R^{mxm}$ so that:

$$A^{(k-1)} \equiv M_{k-1} \cdots M_1 A = \begin{pmatrix} A_{11}^{(k-1)} & A_{12}^{(k-1)} \\ 0 & A_{22}^{(k-1)} \end{pmatrix} \begin{matrix} (k-1) \\ (m-k+1) \end{matrix} \quad (13)$$
$$(k-1) \quad (n-k+1)$$

Where: $A_{11}^{(k-1)}$ is an upper triangular matrix.

Now, if: $A_{22}^{(k-1)} = \begin{pmatrix} a_{kk}^{(k-1)} & . & . & . & a_{kn}^{(k-1)} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ a_{mk}^{(k-1)} & . & . & . & a_{mn}^{(k-1)} \end{pmatrix}$ (14) and

$a_{kk}^{(k-1)} \neq 0$ (15), then the *multipliers:*

$$l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}; i = k+1, \ldots, m \quad (16)$$

Now we define:

$M_k = I - \alpha e_k^T \therefore \alpha = \begin{pmatrix} 0_1 \\ . \\ . \\ . \\ 0_k \\ \alpha_{k+1} \\ . \\ . \\ . \\ \alpha_n \end{pmatrix}$ (17). Then, we must have:

$$A^{(k)} \equiv M_k A^{(k-1)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{matrix} (k) \\ (m-k) \end{matrix} \quad (18)$$
$$(k) \quad (n-k)$$

Where $A_{11}^{(k)}$ is an upper triangular matrix.

The above process illustrates the *k-th* step of matrix decomposition.

Recalling that:

$$(M_k \cdots M_1)^{-1} = M_1^{-1} \cdots M_k^{-1} = \prod_{i=1}^{k} (I_m + \alpha^{(i)} e_i^T) = I_m + \sum_{i=1}^{k} \alpha^{(i)} e_i^T \quad (19)$$

It has the following final expression for matrix decomposition process:

$$A = \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_{m-k} \end{pmatrix} \begin{pmatrix} I_k & 0 \\ 0 & A_{22}^{(k)} \end{pmatrix} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & I_{n-k} \end{pmatrix} \quad (20) \Rightarrow$$

$$(M_k \ldots M_1)^{-1} \equiv \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_{m-k} \end{pmatrix} = I_m + (\alpha^{(1)}, \ldots, \alpha^{(k)}, 0, \ldots, 0). \quad (21)$$

IV. LU DECOMPOSITION THEOREM

Using the above final expression, one can establish the following theorem:

"Whether A_k denote the leader or principal sub-matrix *(kxk)* of $A \in R^{mxn}$. If A_k is not singular matrix for *k = 1, ..., s*; where

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 5, November 2015

$s = min\{m-1, n\}$, then there is a lower triangular matrix $L \in R^{mxm}$ and an upper triangular matrix $U \in R^{mxn}$ such that A = LU. Furthermore, $|A_k| = u_{11}...u_{kk}\ \forall k = 1,...,min\{m,n\}$."

Proof.

In the final expression for the decomposition of **A**, performing operations for the block (1,1) concludes that:

$$A_k = L_{11}^{(k)} A_{11}^{(k)} \quad (22) \quad \Rightarrow$$

$$|A_k| = |L_{11}^{(k)}| \cdot |A_{11}^{(k)}| = 1 \cdot a_{kk}^{(k-1)} \cdot |A_{11}^{(k-1)}| =$$

$$= a_{11}^{(0)} \cdots a_{kk}^{(k-1)} \therefore a_{11}^{(0)} \equiv a_{11} \quad (23)$$

The amounts $a_{kk}^{(k-1)}$ are called *pivots*. Thus, if the conditions of this theorem are satisfied, *nonzero pivots* are guaranteed during triangular decomposition.

V. NEW LINEAR TRANSFORM IN R^n (GLEZT)

If we suppose that:

$$(A,b) \subset R^n \quad (24) \quad \Rightarrow \quad M \subset R^n \quad (25)$$

For the *n-th* case, there is possibility of taking the elements of the original **U**:

$$D_k = \begin{pmatrix} I_{kk} & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & u_{kk} \cdot I_{n-k} \end{pmatrix} \forall (k = 1,...,n-1) \therefore U_{00} = 1. \quad (26)$$

.

We define:

$$GlezT = \prod_{k=n-1}^{1} D_k \cdot M_k \quad (27) \quad \Rightarrow \quad GlezT \subset R^n \quad (28)$$

And from the **GT**, we obtained **GlezT**.

So if:

$$D = \prod_{k=n-1}^{1} D_k \quad (29) \quad \Rightarrow \quad A = LD^{-1}U \quad (30) \quad \Rightarrow \quad DMA = U \quad (31)$$

.

Then the transformation equations between methods could be expressed as:

$$L_{Glez} = LD^{-1} \quad (32) \quad L = L_{Glez} D \quad (33)$$

$$U_{Glez} = DMA = DU \quad (34) \quad \Rightarrow U = D^{-1} U_{Glez} \quad (35)$$

Note how the element D_{kk} of the matrix **D** can provide a criterion for whether the matrix is positive definite, negative definite or indefinite until that iteration.

VI. RE-EXPRESION OF (GLEZT) USING ELEMENTARY MATRICES

The final expression for the linear transformation is [11]:

$$M_k = \begin{pmatrix} \frac{1}{U_{k-1,k-1}} \cdot e_1^T \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{U_{kk}} \cdot e_k^T \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{U_{k-1,k-1}} \cdot e_n^T \end{pmatrix} \cdot \left[x_k I - \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ 0_k \\ x_{k+1} \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \cdot e_k^T \right] \quad (36) \quad (k = 1,...,n);\ U_{00} = 1\ ;$$

Now, if the *k-th* iteration of A_k, A_k^{Adj}, e_k representing the *k-th* column of A, A^{Adj}, I, respectively, where A^{Adj} it is Adjugate Matrix, then:

$$TGlez_{k-1} \cdots TGlez_1 \cdot (e_{k-1} \mid A_{k-1}) = (|A_{k-1}| \cdot e_{k-1} \mid A_{k-1}^{Adj}) \quad (37),$$

and:

$$TGlez_k \cdot (|A_{k-1}| \cdot e_{k-1} \mid A_{k-1}^{Adj}) = (|A_k| \cdot I \mid A_k^{Adj}) \quad (38)$$

From the last equation, we re-expressing the New Linear Transformation:

$$M_k = \frac{1}{a_{k-1k-1}^{(k)}} \begin{pmatrix} a_{kk}^{(k)} & & & & -a_{1k}^{(k)} & & \\ & \ddots & & & \vdots & & \\ & & \ddots & & \vdots & & \\ & & & a_{kk}^{(k)} & & & \\ & & & & a_{k-1k-1}^{(k)} & & \\ & & & & & a_{kk}^{(k)} & \\ & & & & \vdots & & \ddots \\ & & & & -a_{nk}^{(k)} & & a_{kk}^{(k)} \end{pmatrix} \quad (39)$$

$$(k = 1,...,n);\ a_{00}^{(1)} = 1$$

Then we have:

$$A^{Adj} = M = \prod_{k=n}^{1} M_k \quad (40)$$

Applying this result on:

$$Ax = b \quad (41)$$

We obtain:

$$A^{Adj} Ax = A^{Adj} b \quad (42)$$

Simplifying:

$$|A| \cdot I \cdot x = A^{Adj} b \quad (43)$$

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 5, November 2015

Where is cleared:

$$x = \frac{A^{Adj} b}{|A|} \quad (44)$$

This method is better than the exposed by G. W. Stewart [12] because it is numerically stable and works when the matrix is singular. Although in this work the double-precision arithmetic is illustrated, also be used in integer arithmetic, so that its stability is guaranteed and only would have to care for the growing numbers [13].

VII. NUMERICAL RESULTS AND MATHEMATICAL EFFICIENCY

As an illustration we use the matrix [14]:

$$A = (y + \delta_{ij} x); \quad \delta_{ij} = 1 \quad \forall \quad i = j;$$

$$\delta_{ij} = 0 \quad \forall \quad i \neq j \quad ; \quad b = \begin{pmatrix} ny + x \\ . \\ . \\ . \\ ny + x \end{pmatrix} \quad ;$$

The value of the determinant and the Adjugate Matrix are known

$$|A| = x^{n-1}(x + ny)$$

$$A^{Adj} = \begin{pmatrix} x^{n-2}[x + (n-1)] & -x^{n-2}y & . \; . \; . & -x^{n-2}y \\ -x^{n-2}y & x^{n-2}[x + (n-1)] & . \; . \; . & -x^{n-2}y \\ . & . & . \; . \; . & . \\ . & . & . \; . \; . & . \\ . & . & . \; . \; . & . \\ -x^{n-2}y & -x^{n-2}y & . \; . \; . & x^{n-2}[x + (n-1)] \end{pmatrix}.$$

The program we used was written in the style of **CUDA-C** using some intrinsic subroutines of this language. We use subroutines of cuBLAS **Dgemv** 2nd level and Dgemm 3rd levels in double precision. The code was tested on a **TESLA K20c** at high performance computing area of the **ININ**.

We choose $x = 1, y = 1$ and we conducted experiments for eight numbers **N** of equations, namely: **N=200,400,600,800,1000,2000,3000,4000** and we used 2496 cores with 4.8 Gb in RAM Memory. The following table and graphic have shown the execution time in **Double Precision Arithmetic**:

Table 1. Dimension of Matrices and Time in Seconds.

Matrices Size	Subroutines: Dgemm, Dgemv. $O(2n^3)+O(2n^2)$	Time in Seconds
200		0.101
400		0.267
600		0.554
800		1.374
1000		2.537
2000		34.136
3000		161.522
4000		509.361

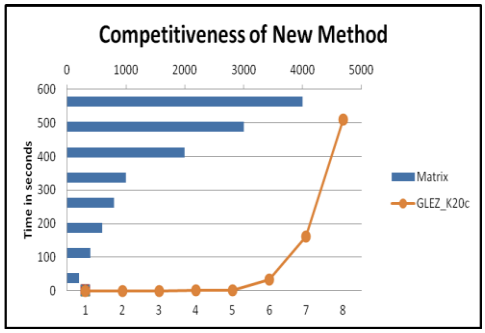

Fig. 1. This results are in Double Precision Arithmetic

The following table and graphic shown the difference in flops for the new algorithm because of its mathematical efficiency

Table 2. Dimension of Matrices, Flops and Mathematical Methods

Matrices Size	Cramer-Glez Subroutines: Dgemm, Dgemv. $O(2n^3)+O(2n^2)$(Flops)	Stewart $O(n^4)$ (Flops)
200	0.1608E+8	1.6E+9
400	0.12832E+9	25.6E+9
600	0.43272E+9	1.296E+11
800	1.02528E+9	4.096E+11
1000	2.002E+9	1E+12
1200	3.45888E+9	2.0736E+12
1400	5.49192E+9	3.8416E+12
1600	8.1971E+9	6.5536E+12
1800	11.67048E+9	1.04976E+13
2000	16.008E+9	1.6E+13
2200	21.30568E+9	2.34256E+13
2400	27.65952E+9	3.31776E+13
2600	35.16552E+9	4.56976E+13
2800	43.91968E+9	6.14656E+13
3000	54.018E+9	8.1E+13
3200	65.55648E+9	1.04858E+14
3400	78.63112E+9	1.33634E+14
3600	93.33792E+9	1.67962E+14
3800	1.09773E+11	2.08514E+14
4000	1.28032E+11	2.56E+14
4200	1.48211E+11	3.1117E+14
4400	1.70407E+11	3.7481E+14
4600	1.94714E+11	4.47746E+14
4800	2.2123E+11	5.30842E+14
5000	2.5005E+11	6.25E+14

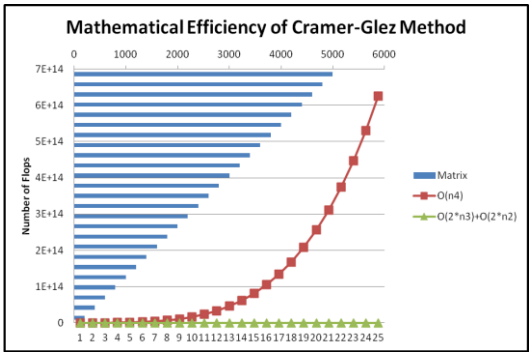

Fig. 2. The difference in Flops is remarkable for the New Algorithm, because it has a Minor Numerical Complexity

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 5, November 2015

VIII. CONCLUSIONS

The computation of the Adjugate Matrix from its definition involves the computation of n^2 determinants of order *(n-1)* – a prohibitively expensive $O(n^4)$ process. On the other hand, the computation from the formula $A^{Adj} = |A| \cdot A^{-1}$ break down when **A** is singular and is potentially unstable when **A** is ill-conditioned with respect to inversion. In this paper we first show that the Adjugate can be calculated in polynomial time $O(n^3)$ and that its numerical stability is unbeatable when running in integer arithmetic. In double precision arithmetic, the numerical results give the correct results for matrices of the order of 4000 or more in efficient way. The new algorithm Cramer-Gonzalez enables the Rule of Cramer becomes a new efficient method for solving systems of linear equations and even if the matrix is singular anyway makes it possible to obtain the Adjugate Matrix.

REFERENCES

[1] Curtis F. Gerald, Wheatley, Patrick O. Applied Numerical Analysis. Addison Wesley Publishing Co.,(1994), Page: 140

[2] Lay, David C., Linear Algebra And Its Applications, Addison Wesley Publishing Co., (1994) Page: 177

[3] Terrence J. Akai, Applied Numerical Methods For Engineers, John Wiley and Sons, Inc., (1994), Page: 56

[4] Valenza, Robert J., Linear Algebra: An Introduction to Abstract , Mathematics, Springer-Verlag, N.Y. (1993) Page: 163

[5] Kolman, Bernard, Introductory Linear Algebra With Applications, Macmillan Publishing Company, (1998) Page: 98

[6] Eaves, Edgar D., Carruth, J. Harvey, Introductory Mathematical Analysis, Wm. C. Brown Publishers (1993) Page: 524

[7] Kahaner, D., Moler, C., Nash, S., Numerical Methods and Software, Prentice Hall, Englewood Cliffs, N.J., (1989) 41-53

[8] Wilkinson, J. H., The algebraic eigenvalue problem, Oxford University Press, Oxford, (1965) 189-264

[9] Golub, G.H., Van Loan Ch. F. Matrix Computations. Jhon Hopkins University Press (1985)

[10] González, H.E., Carmona L., J.J., "A new LU decomposition on hybrid GPU-accelerated multicore systems".Computación y Sistemas. vol. 17 no. 3, (2013) 413-422

[11] González, H.E., "Método Cramer–LU aplicado al Algoritmo Simplex". Tesis Doctoral. DEPFI-UNAM. (2005) 22-25

[12] Stewart. G.W., On the adjugate matrix, Linear Algebra and its Applications 283 (1998) 151-164

[13] González, H.E., Carmona L., J.J., Solving Simultaneous Linear Equations using Finite Fields On Hybrid GPU-Accelerated Multi-core Systems, International Journal of Engineering and Innovative Technology (IJEIT), Volume 4, Issue 6, December (2014)

[14] Young, David M.; Gregory, Robert Todd. A Survey of Numerical Mathematics. Volume II. Dover Publications, Inc., (1972)

AUTHOR BIOGRAPHY

H. E. González. He received the PhD in Operations Research at the National Autonomous University of Mexico (UNAM) in 2005. He has been working in research activities for more than twenty years. He has published a book and more than ten scientific papers. Presently, he is a full-time researcher at the National Institute of Nuclear Research (ININ) at the Department of Systems.

J.J. Carmona L., He received his B.S. in Electronic Engineering from UAM. Presently, he is a full-time researcher at the National Institute of Nuclear Research (ININ) at the Department of Systems.