

## INTEGRATION DE L'OUTIL INFORMATIQUE DANS LA RESOLUTION DES PROBLEMES D'ALGEBRE LINEAIRE

Par

**Sébastien NSUNGU MBUDI**

*Assistant de deuxième mandat à Institut Supérieur Pédagogique de Boma  
Département des Mathématiques*

### RESUME

*Notre objectif ultime dans ces petits manuscrits est de mettre en place un manuel d'apprentissage sur la programmation graphique sous le langage Labview. Un langage encore moins répandu au sein de la communauté scientifique congolaise, pourtant offrant des nombreux avantages (flexibilité, rapidité, double interface, ...). Pour illustrer, nous avons traité le cas de l'algèbre linéaire où par programmation nous avons réalisé les interfaces, ici présentées, associées à leurs blocs-diagramme. Cette étude a débouché sur la programmation d'un mini calculateur matriciel utilisable même en laboratoire.*

**Mots-clés :** *Intégration, Outil informatique, Programmation graphique, Labview 7.1, Programmation en double interfaces, Programmation par flot de données, Virtual instrument, Algèbre linéaire, Vecteurs et matrice, Mini calculateur algébrique*

### ABSTRACT

*Our ultimate aim in these short manuscripts is to produce a textbook on graphical programming using the Labview language. This is a language that is not widely used in the Congolese scientific community, yet it offers many advantages (flexibility, speed, dual interface, etc.). To illustrate this, we used the case of linear algebra, where we programmed the interfaces presented here, together with their block diagrams. This study led to the programming of a mini matrix calculator that can also be used in the laboratory.*

**Keywords:** *Integration, Computer tool, Graphical programming, Labview 7.1, Dual interface programming, Data flow programming, Virtual instrument, Linear algebra, Vectors and matrix, Mini algebraic calculator*

## I. INTRODUCTION

### I.1 Problématique

La résolution des problèmes d'algèbre linéaire, plus particulièrement le calcul matriciel, par des méthodes analytiques conduit souvent à des calculs fastidieux mettant ainsi l'opérateur devant un fait à accomplir.

Le calcul de l'inverse d'une matrice, la diagonalisation d'une matrice carrée ou encore le calcul du déterminant d'une matrice d'ordre supérieur à 2 nécessite une bonne maîtrise des méthodes et procédures, d'une part, et la résolution des systèmes d'équations linéaires ou encore la recherche des valeurs et vecteurs propres d'une matrice fait dans certain cas appel à des méthodes particulièrement complexes surtout lorsqu'il s'agit des systèmes de plus de 3 inconnues : les méthodes de pivot de Gauss ou d'échelonnement des matrices ne sont toujours pas faciles à manipuler, d'autre part.

Lors de la recherche rapide des solutions, une alternative à ces méthodes serait toujours souhaitable. C'est ainsi nous avons choisi l'approche informatique pour palier à cette préoccupation.

### I.2 Objectif de notre recherche

L'objet de notre étude est de proposer une autre méthode pouvant permettre d'aborder les mêmes problèmes avec plus de rapidité mais aussi et surtout plus de précision possible.

### I.3 Motivations

La réalisation de ce travail a été motivée par le fait que cela nous a permis de :

- mettre à la disposition de la communauté estudiantine plus particulièrement ceux en sciences exactes un outil d'initiation à la programmation graphique avec LabView, un langage pas assez répandu dans le système éducatif congolais, pourtant aussi puissant que le Matlab, et plus facile à manipuler que les langages textuels ;
- développer des interfaces utilisateurs lesquelles vont permettre aux apprenants et à d'autres chercheurs de vérifier leurs résultats obtenus après calcul avec les méthodes analytiques ;
- faire découvrir à travers ces quelques pages les ressources qu'offre ce langage de par sa flexibilité de programmer en double interfaces indissociables.

### I.4 Présentation des instruments virtuels <sup>1</sup>

LabView (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation dédié au contrôle d'instruments et l'analyse de

---

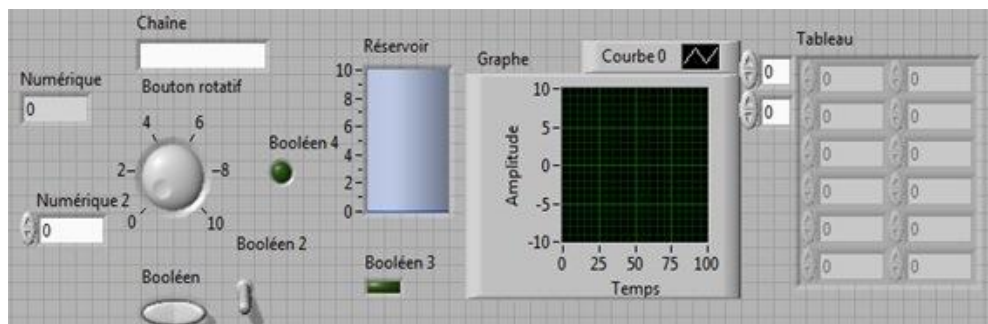
<sup>1</sup> R. Decourt, *LabView*, Notes de cours, Edition 2011.

données qui utilise des icônes à la place de lignes de texte pour créer des applications. Contrairement à la nature séquentielle des langages textuels, LabView est basé sur un environnement de programmation graphique utilisant la notion de flot de données pour ordonnancer les opérations.

Les programmes LabView portent la dénomination d'instruments virtuels ou VIs, car leur apparence et leur fonctionnement imitent celui d'instruments réels tels que les oscilloscopes et les multimètres. Chaque VI utilise des fonctions qui manipulent les entrées de l'interface utilisateur ou d'autres sources et qui affichent ces informations ou les déplacent vers d'autres fichiers ou ordinateurs.<sup>2</sup>

Un VI contient les trois composants principaux suivants : la Face Avant, le Diagramme et l'Icône/Connecteur.

- *La face avant d'un VI* est avant tout une combinaison de commandes et d'indicateurs. Les commandes sont les entrées des Vis et fournissent les données au diagramme. Les indicateurs sont les sorties des VIs et affichent les données générées par le diagramme. On peut utiliser plusieurs types de commandes et d'indicateurs tels que les commandes et indicateurs numériques, à curseur, booléens, chaîne de caractères, les tables et les graphes.



*Fig. I Face-avant avec différents commandes et indicateurs*

On construit la face avant en plaçant des éléments graphiques accessibles dans une palette.

Chaque objet déposé sur la face avant génère dans le diagramme un symbole appelé « Terminal ». Ce terminal contient la valeur de l'objet graphique correspondant. Le symbole représente le type de la donnée (par ex. DBL pour double), le sens du flot, (maigre avec une flèche à droite s'il s'agit d'un indicateur, gras avec une flèche à gauche s'il s'agit d'une commande).

<sup>2</sup> I2C NXP User Manual UM10204 : [http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)

- *Le diagramme* contient les terminaux, les sous VIs, les fonctions, les constantes, les structures ainsi que les fils qui relient les différents objets pour leur transmettre les données.

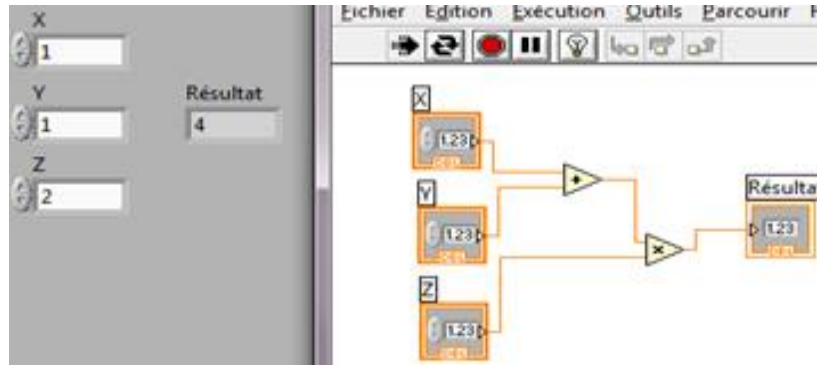


Fig. II Programme 1 : le bloc diagramme et face-avant qui calcule  $C = (x + y)z$

- *Icône et cadre connecteur* : identifie le VI pour être utilisé dans un autre VI. Un VI dans un autre VI est appelé un sous-VI. Un sous-VI correspond à un sous-programme dans des langages de programmation textuels.

### I.5 Programmation par flots de données <sup>3</sup>

Utilisant le principe du contrôle de flot de données, LabVIEW n'exécute un nœud que lorsque l'ensemble des données arrivant sur ses entrées est présent. Après exécution, le nœud transmet les données sur ses sorties, les passant ainsi au nœud suivant.

Considérons le diagramme suivant :

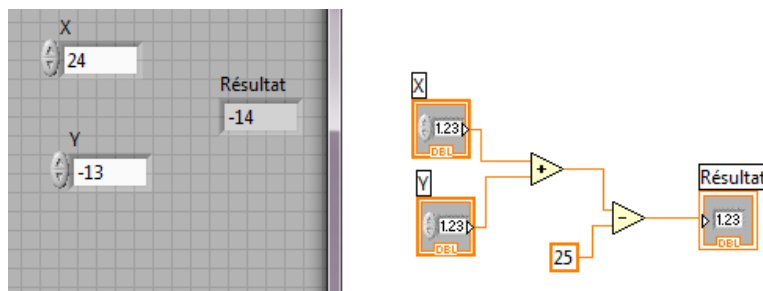


Fig. III Programme 2 : Programmation par flots de données  $C = (x + y) \times 25$

Il additionne deux nombres et soustrait 25 au résultat de l'addition. Dans ce cas, le diagramme s'exécute de la gauche vers la droite, non pas parce que les objets sont placés dans cet ordre, mais parce que l'une des entrées de la fonction « Soustraire » reste invalide tant que la fonction « Additionner » n'a pas fini son exécution.

<sup>3</sup> R. Decourt, *Op.cit.*

## II. MATRICES ET OPERATIONS MATRICIELLES SOUS LABVIEW

### II.1 Palette d'Algèbre linéaire de LabVIEW<sup>4</sup>

Résoudre les problèmes d'algèbre linéaire sous LabView revient à exploiter les ressources de la palette « Algèbre linéaire » accessible à partir du Diagramme en faisant clic droit puis **Fonction** → **Toutes les fonctions** → **Analyse** → **Mathématique** → **Algèbre linéaire**.



Fig. IV Palette d'Algèbre linéaire de LabVIEW

Labview utilise également des tableaux pour représenter des vecteurs et des matrices. Un vecteur est représenté comme un tableau à une dimension, tandis qu'une matrice comme un tableau à deux dimensions.

On trouve les principaux contrôleurs des tableaux et matrices dans la palette « Tableau, Matrice & Cluster » accessible par clic droit en face-avant du VI.

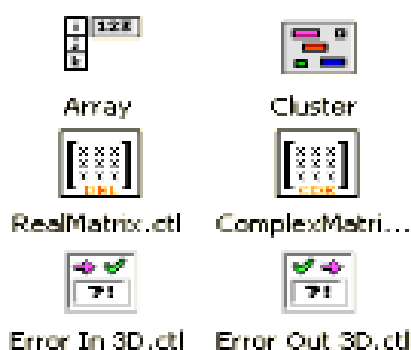


Fig. V Les principaux contrôleurs des tableaux et matrices

<sup>4</sup> <http://home.hit.no/~hansha/> University College of Southeast Norway

## II.2 Vecteur <sup>5</sup>

Etant donné un vecteur  $X$  tel que :  $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$ . On peut prendre comme exemple d'un vecteur de  $\mathbb{R}^3$  :  $X = \begin{pmatrix} 4 \\ -1 \\ 2 \end{pmatrix}$ . Sa représentation à la face-avant est :

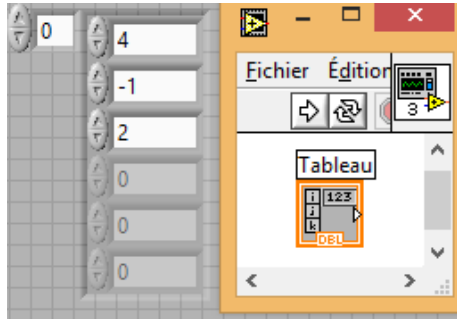


Fig. VI Tableau à 1D illustrant un vecteur  $X$

Application : Un programme qui calcule le produit scalaire, le produit vectoriel et le produit externe des deux vecteurs  $X$  et  $Y$  ainsi la racine carrée du vecteur  $Y$  et la norme de la matrice  $M$ .

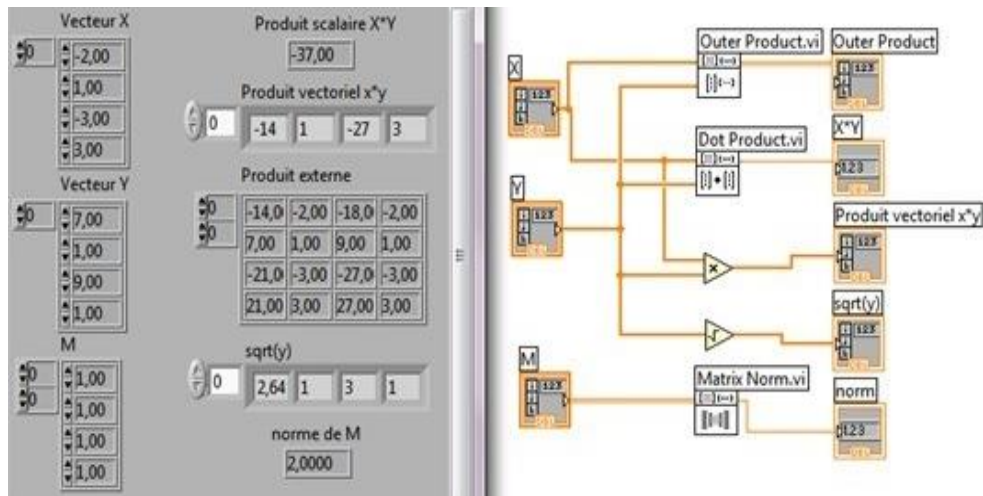


Fig. VII Programme 3 : Calcul du produit scalaire, produit vectoriel, produit externe, racine carrée et la norme vectorielle

<sup>5</sup> KUMBI MWAMBA, *Algèbre linéaire « KUMBINOV »*, Fac. Polytechnique, UNIKIN, Edition 2004.

### II.3 Matrice <sup>6</sup>

Soit une matrice  $A \in \mathcal{M}_{(n,m)}$ , où  $\mathcal{M}_{(n,m)}$  est un ensemble des matrices de n-linges et m-colonnes telle que :

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \in \mathcal{M}_{(n,m)}$$

Par exemple pour  $n = m = 3$ , on obtient une matrice carrée d'ordre 3 comme celle ci-dessous :

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sur la face avant, cette matrice se représente de la manière suivante :

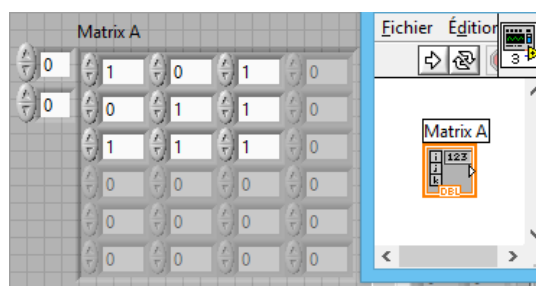


Fig. VIII Tableau 2D illustrant une matrice A d'ordre 3

**N.B.** : Les indices d'incrémentation à gauche d'un tableau détermine bien sa dimension et peuvent être visibles ou cachés selon le choix de l'opérateur.

#### II.3.1 Transposée d'une matrice carrée <sup>7</sup>

La transposée d'une matrice  $A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \in \mathcal{M}_{(n,m)}$  s'écrit par :

$$A^T = \begin{bmatrix} a_{11} & \dots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \dots & a_{nm} \end{bmatrix} \in \mathcal{M}_{(m,n)}$$

Par exemple lorsqu'on donne une matrice A telle que :

$$A = \begin{bmatrix} 1 & 3 & -1 \\ -4 & 2 & 1 \\ -2 & 1 & 1 \end{bmatrix} \text{ alors } A^T = \begin{bmatrix} 1 & 3 & -1 \\ -4 & 2 & 1 \\ -2 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & -4 & -2 \\ 3 & 2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

<sup>6</sup> Pierre BOUSQUET et al., *Mathématiques générales II-Algèbre linéaire*, Université Aix-Marseille 1, PEIP-L1, 2012.

<sup>7</sup> *Idem.*



**Remarque :** Les notations suivantes sont également utilisées pour la transposée de la matrice  $A$  :  ${}^tA$ ,  ${}^T A$ ,  $A'$ .

Sous Labview, on trouve :

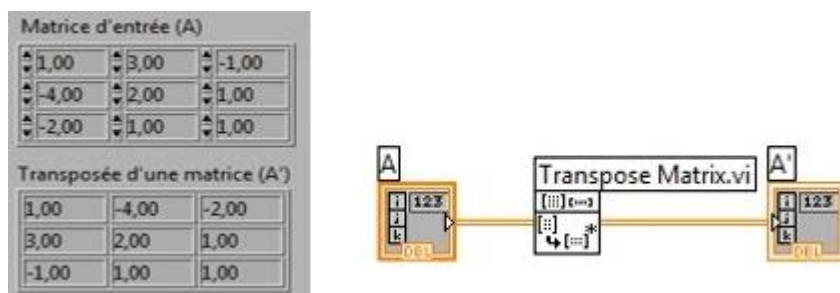


Fig. IX Programme 4 : calcul de transposée de la matrice  $A$

### II.3.2 Addition matricielle <sup>8</sup>

On peut définir la somme de deux matrices si elles sont de même taille. Soient  $A$  et  $B$  deux matrices de taille  $(n, m)$ . On définit leur somme  $C = A + B$ , de taille  $(n, m)$ , par :

$$c_{ij} = a_{ij} + b_{ij}$$

En d'autres termes, on somme composante par composante.

**N.B.** : Il n'existe pas des fonctions spéciales pour l'addition matricielle, on utilise juste la fonction addition standard présent dans la palette « Numérique » de Labview.

Exemple : un programme qui calcule l'addition de deux matrices

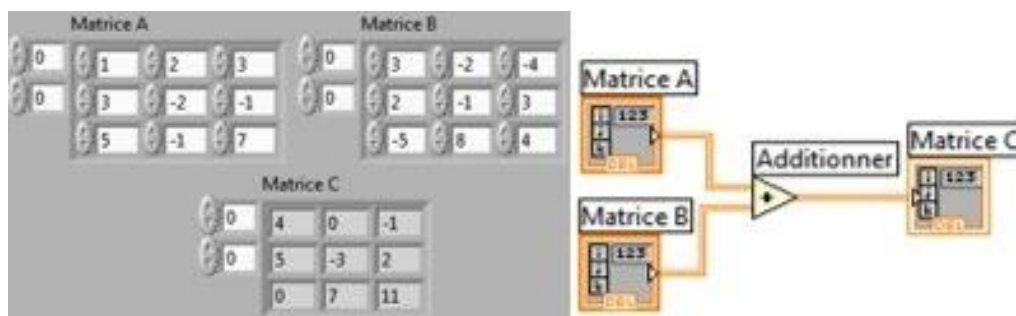


Fig. X Programme 5 : calcul de l'addition de deux matrices  $C = A + B$

<sup>8</sup> Arnaud BODIN et al., *Algèbre, Cours de mathématiques première année*, Université de Lille 1 et Unisciel, 2016.



**Propriétés de l'addition matricielle :**

Soient A, B et C trois matrices carrées d'ordre m, on peut donc vérifier les propriétés suivantes sous Labview :

- 1°) *La commutativité matricielle :  $A + B = B + A$*
- 2°) *La transitivité matricielle :  $A + (B + C) = (A + B) + C$*
- 3°) *L'élément neutre :  $A + 0 = 0 + A = A$*
- 4°) *Transposée de la somme des deux matrices :  $(A + B)^T = A^T + B^T$*



Fig. XI Programme 6 : Face-avant sur les propriétés de l'addition matricielle

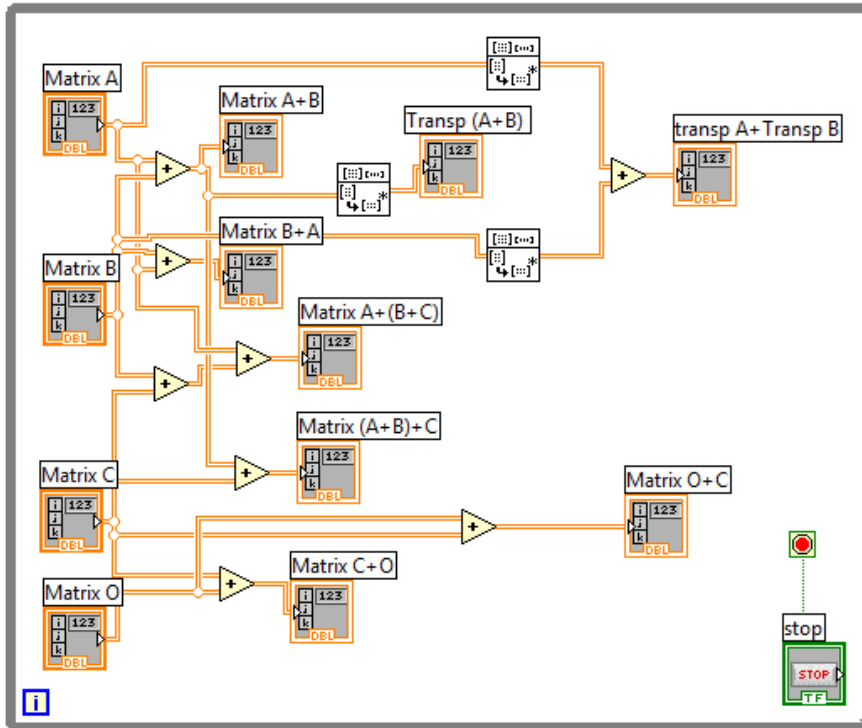


Fig. XII Bloc diagramme correspondant au programme 6

II.3.3 Multiplication matricielle<sup>9</sup>

Étant donné deux matrices  $A \in R^{n \times m}$  et  $B \in R^{m \times p}$ , alors on peut calculer la matrice  $C$  telle que :  $C = AB \in R^{n \times p}$ , où  $C_{jk} = \sum_{i=1}^n a_{ji}b_{ik}$ .

Exemple :

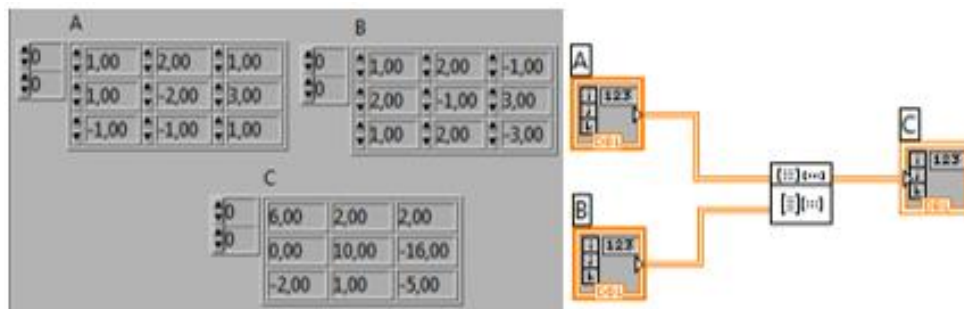


Fig. XIII Programme 7 : Calcul de la matrice  $C = A \times B$

<sup>9</sup> Arnaud BODIN et al., *op. cit.*

### II.3.4 Produit d'une matrice par une matrice colonne à droite

Le produit de la matrice  $M = (c_1 \dots c_i \dots c_n)$ , où  $c_i$  est la  $i^{\text{ème}}$  colonne de  $M$ , par la matrice colonne  $X$  est :

$$MX = (c_1 \dots c_n) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = x_1 c_1 + \dots + x_n c_n = \sum_{i=1}^n x_i c_i$$

Exemple:  $A = \begin{pmatrix} 2 & -2 & 6 \\ 1 & 1 & -4 \\ -2 & 3 & 3 \end{pmatrix}$  ;  $B = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$

$$AB = 1c_1 + 2c_2 + (-1)c_3$$

$$= 1 \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} + 2 \begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 6 \\ -4 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 - 4 - 6 \\ 1 + 2 + 4 \\ -2 + 6 - 3 \end{pmatrix} = \begin{pmatrix} -8 \\ 7 \\ 1 \end{pmatrix}$$

Sous LabVIEW le problème se résous de la manière ci-après :

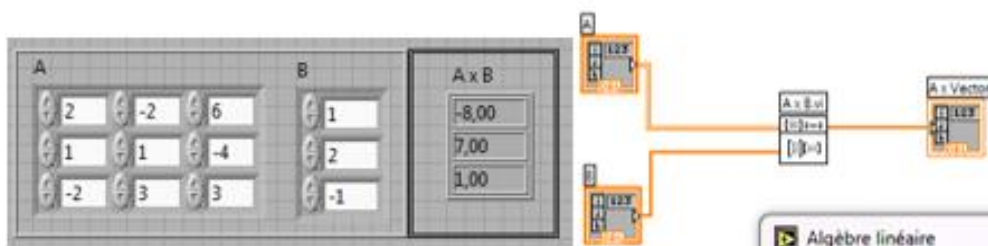


Fig. XIV Programme 8 : Produit d'une matrice par une matrice colonne à droite

### II.3.5 Produit d'une matrice par une matrice ligne (ou vecteur ligne) à gauche<sup>10</sup>

Soit une matrice  $M = \begin{pmatrix} L_1 \\ \vdots \\ L_m \end{pmatrix}$  et soit un vecteur  $Y = (y_1 \dots y_m)$ .

Le produit de la matrice  $M$  par le vecteur ligne  $Y$  est :

$$Y.M = (y_1 \dots y_m) \begin{pmatrix} L_1 \\ \vdots \\ L_m \end{pmatrix} = y_1 L_1 + \dots + y_i L_i + \dots + y_m L_m \rightarrow Y.M$$

$$= \sum_{i=1}^m y_i L_i$$

<sup>10</sup> KUMBI MWAMBA, *op. cit.*

Exemple :  $M_1 = \begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & 0 \end{pmatrix}$ ;  $Y = (2 \ 1)$

$$Y.M_1 = (2 \ 1) \begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & 0 \end{pmatrix} = (3 \ 0 \ -2)$$

Ainsi, sous LabVIEW le problème se résout de la manière ci-après :

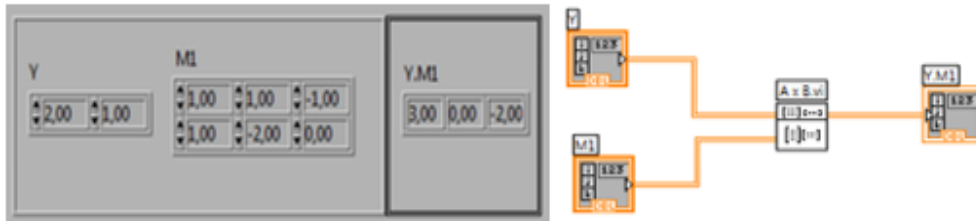


Fig. XV Programme 9 : Produit d'une matrice par une matrice ligne à gauche

### Propriétés

- 1°) L'Anti-commutativité de la multiplication matricielle :  $AB \neq BA$
- 2°) La transitivité de la multiplication matricielle :  $A(BC) = (AB)C$
- 3°) La distributivité de la multiplication matricielle par rapport à l'addition :  $(A + B)C = AC + BC$

The front panel is titled 'Propriétés de Multiplication Matricielle'. It is divided into two main sections: 'Tableaux Matrices d'Entrée' and 'Tableaux Matrices de sortie'.  
 Under 'Tableaux Matrices d'Entrée':  
 - Matrix A:  $\begin{bmatrix} 3 & -2 & 12 \\ 2 & 0 & -7 \\ -1 & 2 & -4 \end{bmatrix}$   
 - Matrix B:  $\begin{bmatrix} -3 & 2 & 1 \\ -2 & -1 & 3 \\ 3 & 1 & 1 \end{bmatrix}$   
 - Matrix C:  $\begin{bmatrix} -1 & 1 & 5 \\ 4 & 2 & -2 \\ 3 & -2 & 1 \end{bmatrix}$   
 Under 'Tableaux Matrices de sortie':  
 - Matrix A x B:  $\begin{bmatrix} 31,00 & 20,00 & 9,00 \\ -27,00 & -3,00 & -5,00 \\ -13,00 & -8,00 & 1,00 \end{bmatrix}$   
 - Matrix (A x B)C:  $\begin{bmatrix} 76,00 & 53,00 & 124,00 \\ 0,00 & -23,00 & -134,00 \\ -16,00 & -31,00 & -48,00 \end{bmatrix}$   
 - Matrix (A+B)C:  $\begin{bmatrix} 39,00 & -26,00 & 13,00 \\ -16,00 & 6,00 & -2,00 \\ 1,00 & 14,00 & 1,00 \end{bmatrix}$   
 - Matrix B x A:  $\begin{bmatrix} -6,00 & 8,00 & -54,00 \\ -11,00 & 10,00 & -29,00 \\ 10,00 & -4,00 & 25,00 \end{bmatrix}$   
 - Matrix A(B x C):  $\begin{bmatrix} 76,00 & 53,00 & 124,00 \\ 0,00 & -23,00 & -134,00 \\ -16,00 & -31,00 & -48,00 \end{bmatrix}$   
 - Matrix AC+BC:  $\begin{bmatrix} 39 & -26 & 13 \\ -16 & 6 & -2 \\ 1 & 14 & 1 \end{bmatrix}$   
 - B x C:  $\begin{bmatrix} 14,00 & -1,00 & -18,00 \\ 7,00 & -10,00 & -5,00 \\ 4,00 & 3,00 & 14,00 \end{bmatrix}$   
 - Matrix A x C:  $\begin{bmatrix} 25,00 & -25,00 & 31,00 \\ -23,00 & 16,00 & 3,00 \\ -3,00 & 11,00 & -13,00 \end{bmatrix}$   
 A 'STOP' button is located at the bottom right.

Fig. XVI Programme 10 : Face-avant simulant les propriétés de la multiplication matricielle

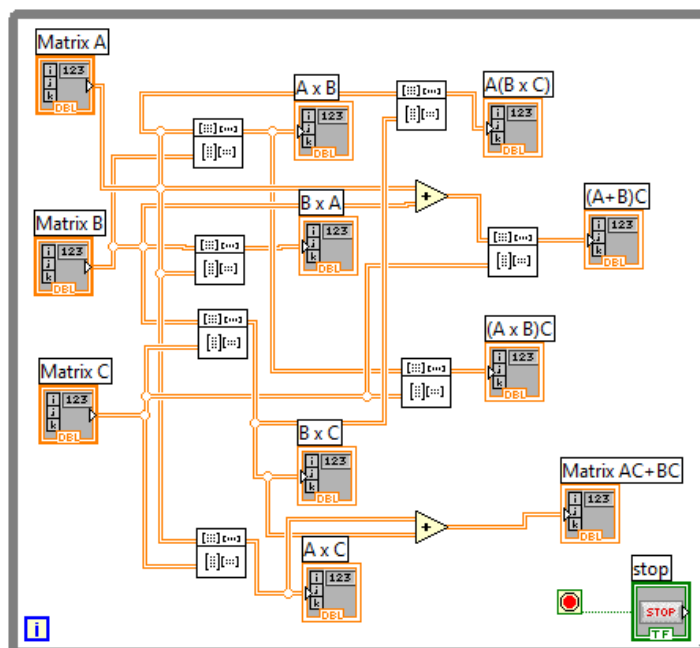


Fig. XVII Bloc diagramme correspondant au programme 10

#### II.4 Déterminant d'une matrice carrée <sup>11</sup>

Étant donné une matrice carrée  $A \in \mathcal{M}_n$ , alors le déterminant est donné par la relation suivante :

$$\det(A) = |A|$$

Pour une matrice  $2 \times 2$  donnée par :

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \in \mathcal{M}_2$$

Alors le déterminant se calcule par :

$$\det(A) = |A| = a_{11}a_{22} - a_{21}a_{12}$$

Par exemple :

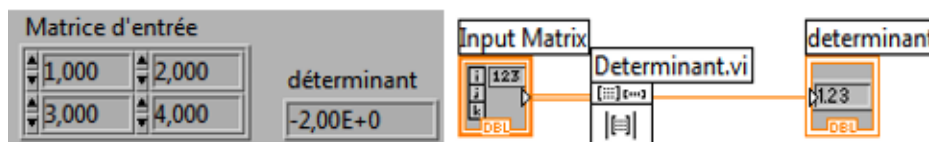


Fig. XVIII Programme 11 : Calcul du déterminant d'une matrice

<sup>11</sup> Ronald W. Larsen, *Labview for engineers*, Montana State University, 2015.

**Propriétés :**

1°)  $\det(AB) = \det(A) \det(B)$

2°)  $\det(A^T) = \det(A)$

En effet, on aura donc :

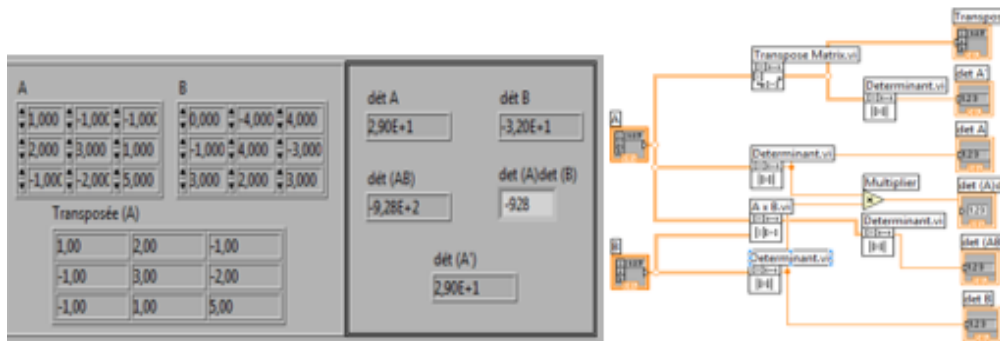


Fig. XIX Programme 12 : Propriétés du déterminant d'une matrice

**II.5 Calcul de l'inverse d'une matrice carrée <sup>12</sup>**

Soit  $A$  une matrice carrée d'ordre  $n$  :  $A = (a_{ij})$ ,  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, n$ . Si  $\det(A) \neq 0$ , alors  $A$  est inversible ou régulière et sa matrice inverse est :

$$A^{-1} = \frac{1}{\det(A)} A^\#$$

où  $A^\#$  est la transposée de la matrice des cofacteurs de  $A$  (aussi appelée matrice adjacente) :  $A^\# = C^T$  où  $C = (C_{ij})$

Exemple :  $A = \begin{pmatrix} -1 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 1 \end{pmatrix}$

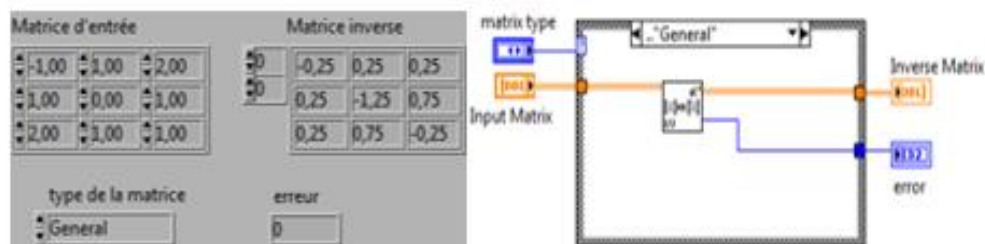


Fig. XX Programme 13 : Calcul de l'inverse d'une matrice carrée

<sup>12</sup> Ronald W. Larsen, *op. cit.*

Une autre solution possible et plus rapide serait d'utiliser la fonction « inverser une matrice » directement accessible à la sous-palette « algèbre linéaire » telle que :

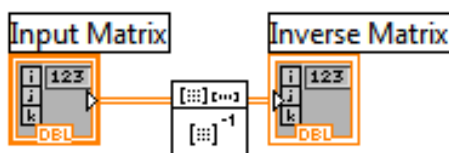


Fig. XXI Diagramme alternatif au programme 13

## II.6 Matrices diagonales

On appelle matrice diagonale une matrice carrée dont tous les termes sont nuls sauf éventuellement ceux qui sont situés sur sa diagonale principale :

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\}, D = (a_{ij})_{i=1, \dots, n, j=1, \dots, n} \text{ avec } a_{ij}=0 \text{ si } i \neq j$$

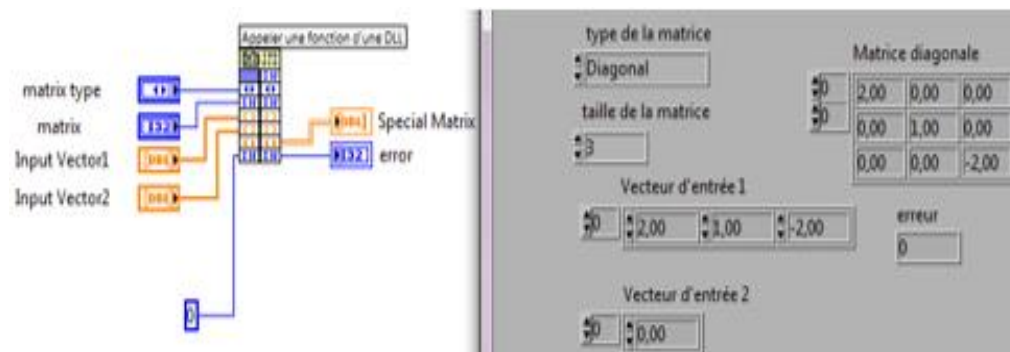


Fig. XXII Simulation d'une matrice diagonale

## II.7 Matrice identité/unité

Une matrice unité ou matrice identité d'ordre  $n$ ,  $I_n$  (ou  $I$ ), est une matrice carrée diagonale d'ordre  $n$  dont tous les termes situés sur la diagonale principale sont égaux à 1 tandis que les autres termes sont nuls.

$$I_n \in \mathcal{M}_n(\mathbb{R}) \text{ et } \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\}, \begin{cases} a_{ii} = 1 \\ a_{ij} = 0 \text{ si } i \neq j \end{cases}$$

$$I_n = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 1 \end{pmatrix}$$

Exemple dans labVIEW : pour  $\mathcal{M}_n(\mathbb{R}^3)$



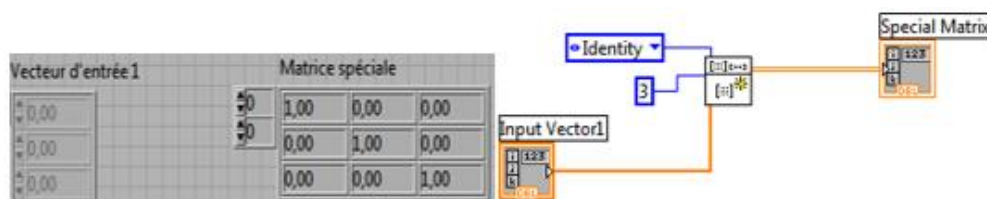


Fig. XXIII Détermination de la matrice identité

## II.8 Trace, Rang d'une matrice

La trace de la matrice carrée  $A$  est le nombre noté  $tr(A)$  :

$$tr A = a_{11} + a_{22} + \dots + a_{nn}$$

tandis que le rang d'une matrice  $A$  représente le nombre maximum des vecteurs colonnes linéairement indépendants non nuls.

Sous LabVIEW, on obtient :

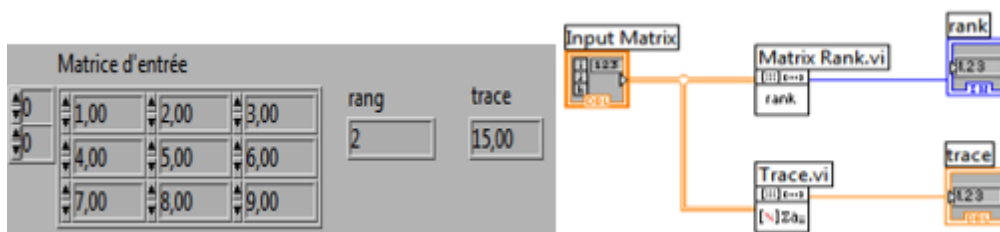


Fig. XXIV Programme 14 : Calcul de la trace et du rang d'une matrice

## II.9 Puissance d'une matrice

Pour tout  $A \in \mathcal{M}_n(K)$ , on définit les puissances successives de  $A$  par :  $A^0 = I_n$  et  $A^{p+1} = A^p \times A$  pour tout  $p \in \mathbb{N}$ .

Exemple : soit la matrice  $A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{pmatrix}$  calculer  $A^2$ ,  $A^3$  et  $A^5$

En effet, 2, 3, 4, 5, ... ne constituent que l'ordre de l'opération de puissance, d'où :

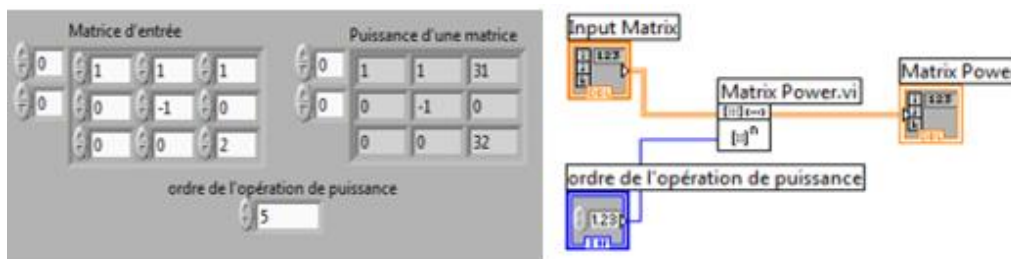


Fig. XXV Programme 15 : Détermination de la puissance d'une matrice

### III. SYSTEMES D'EQUATIONS LINEAIRES <sup>13</sup>

#### III.1 Système linéaire de $n$ équations à $p$ inconnues

On appelle système linéaire de  $n$  équations à  $p$  inconnues  $(x_1, x_2, \dots, x_p)$  tout système (S) de la forme :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p = b_1 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{np}x_p = b_n \end{cases}$$

Matriciellement ce système s'écrit :

$$Ax = b$$

avec comme solution :

$$x = A^{-1}b$$

tout en supposant que la matrice inverse de A existe.

Exemple : soit à résoudre le système suivant : 
$$\begin{cases} 2x_1 + 3x_2 + 4x_3 = 7 \\ 2x_1 + x_2 + 4x_3 = 8 \\ 3x_1 + 5x_2 + 7x_3 = 4 \end{cases}$$

par la méthode de la matrice inverse :  $x = A^{-1}b$ . On calcule d'abord  $A^{-1}$  :

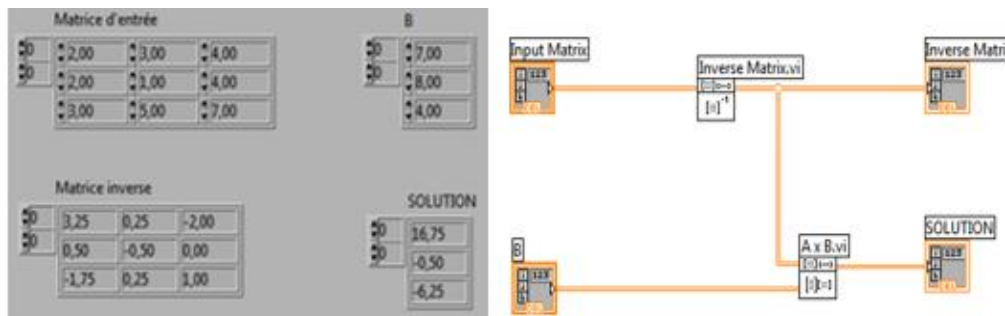


Fig. XXVI Programme 16 : Résolution du système par calcul de la matrice inverse

Il est possible d'aborder le problème en faisant intervenir la fonction « Résolution d'équations linéaires » figurant sur la palette Algèbre linéaire, soit :

<sup>13</sup> FPGA module training : <http://zone.ni.com/devzone/cda/tut/p/id/3555>

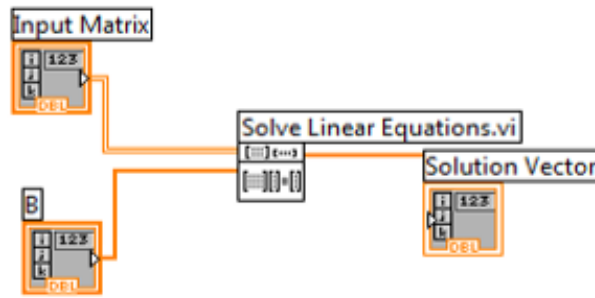


Fig. XXVII Alternative au programme 16 par une seule fonction SLE.

### III.2 Vecteurs propres-valeurs propres <sup>14</sup>

On dit que le complexe  $\lambda$  est une *valeur propre de la matrice A* s'il existe un vecteur colonne  $P$  non nul tel que :  $AP = \lambda P$

Le vecteur  $P$  est alors appelé *vecteur propre de A associé à  $\lambda$* . Autrement dit : Les valeurs propres de  $A$  sont les solutions de l'équation :  $\det(A - \lambda I) = 0$

L'ensemble des vecteurs propres d'une matrice carrée  $A$  associés à  $\lambda$  est l'ensemble des vecteurs  $P$  tels que  $AP = \lambda P$ , autrement dit tels que :  $(A - \lambda I)P = \vec{0}$

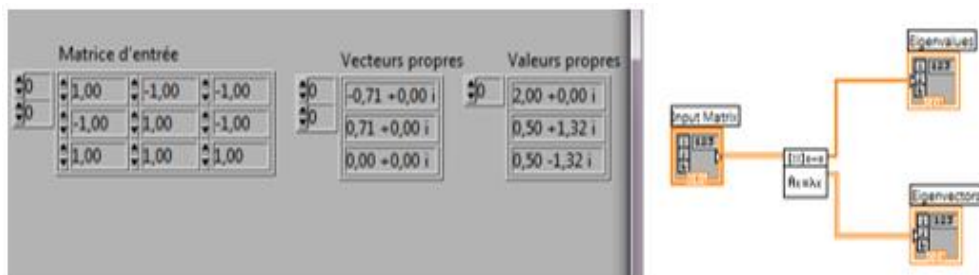


Fig. XXVIII Programme 17 : calcul des vecteurs et valeurs propres d'une matrice

<sup>14</sup> Ronald W. Larsen, *op. cit.*

### Application : Mini Calculateur Algébrique Numérique

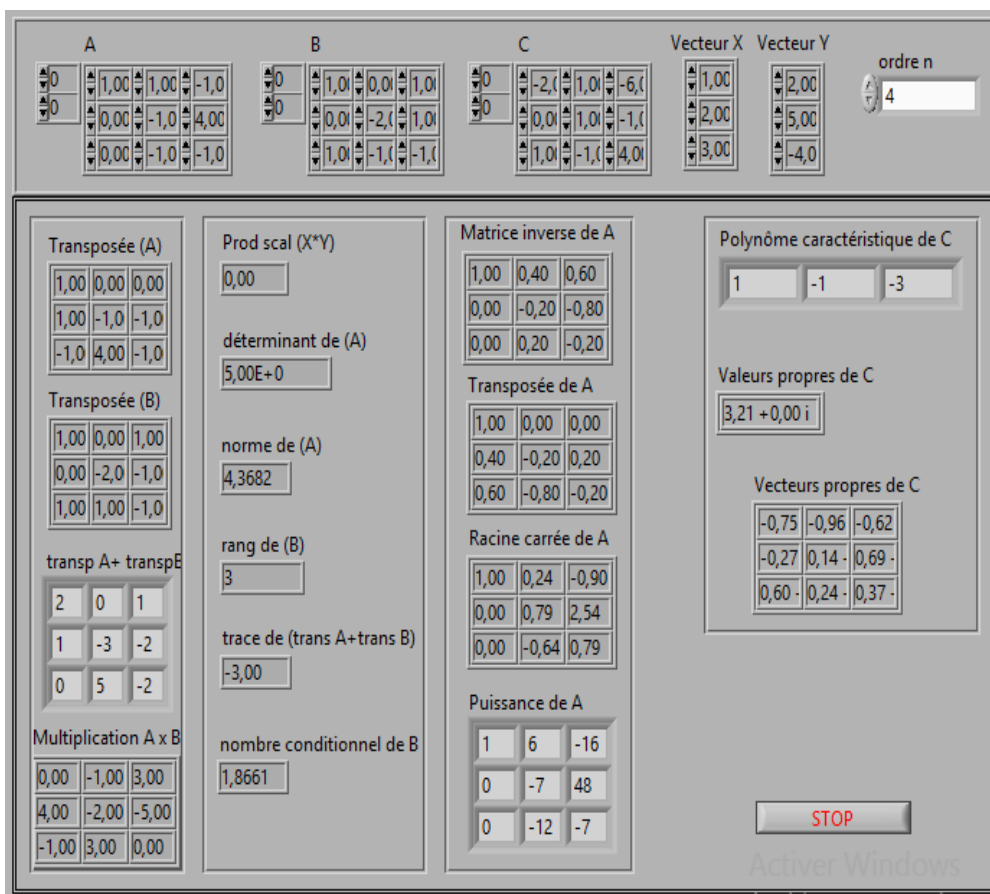


Fig. XXIX Programme 18 : Face-avant d'un Mini Calculator Algébrique

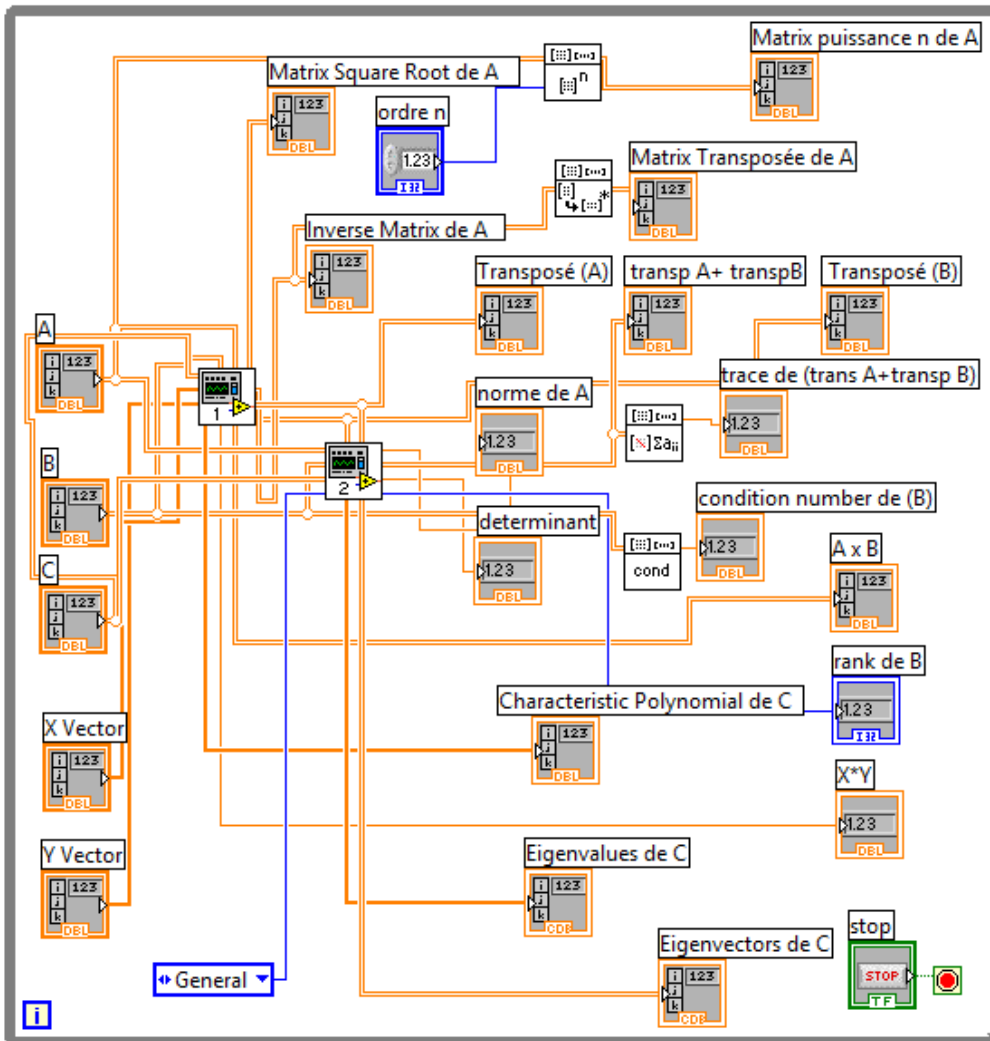


Fig. XXX Programme 18 : Bloc diagramme du Mini Calculator Algébrique

## CONCLUSION

Loin de faire toute l'algèbre linéaire dans cet article, nous avons voulu montrer l'importance de l'outil informatique, en particulier le logiciel LabView, dans la résolution des problèmes d'algèbre matricielle surtout la rapidité et la fiabilité qu'il offre dans la recherche des solutions.

Cependant, LabView a trois forces. D'abord, c'est un langage de programmation au sens informatique du terme, au même rang que Matlab, C, etc. On peut bien y faire les mêmes programmes même s'il est sans doute moins facile de le faire avec LabView. Puis, il propose une interface de programmation extrêmement simple d'utilisation. Enfin, il est associé à cet outil logiciel toute une batterie de cartes électroniques qui facilitent la mise en place d'une application.

Le logiciel LabView 7.1, de par sa simplicité de programmer, nous a permis d'implémenter des interfaces lesquelles pouvant être réutilisées par d'autres utilisateurs par implantation après encapsulation sur une autre machine ou même dans un laboratoire ou un centre de recherche.

**BIBLIOGRAPHIE**

- BODIN Arnaud et al., *Algèbre, Cours de mathématiques première année*, Université de Lille 1 et Unisciel, 2016.
- BOUSQUET Pierre et al., *Mathématiques générales II-Algèbre linéaire*, Université Aix-Marseille 1, PEIP-L1, 2012.
- DECOURT R., *LabView*, Notes de cours, Edition 2011.
- KAKUDJI E., *LabView*, Notes de cours, L1 Physique, Fac. Sciences, UNIKIN, 2011
- KUMBI MWAMBA, *Algèbre linéaire « KUMBINOV »*, Fac. Polytechnique, UNIKIN, Edition 2004.
- RONALD W. Larsen, *LabView for engineers*, Montana State University, 2015.

**Sites internet :**

- <http://home.hit.no/~hansha/> University College of Southeast Norway
- FPGA module training :
- <http://zone.ni.com/devzone/cda/tut/p/id/3555>
- I2C NXP User Manual UM10204 :  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)