**Travail #1**

**Matrices et Systèmes d’équations**

**PIF1006**

**Mathématiques pour informaticiens II**

**Présenté à**

**Adam Joly**

**Par**

**Vincent Gélinas**

**Et**

**Dominic Lafrance**

**Université du Québec à Trois-Rivières**

**Novembre 2016**

**Instructions sur l’utilisation et le fonctionnement du programme**

Nous avons utilisé github pour la gestion de version de notre programme, même si nous avons travaillé ensemble lors de la durée du projet. Le lien github est le suivant : <https://github.com/vinki11/tpMatrice>. Le programme a été fait en console. Lors du lancement du programme, il y déjà 5 matrices et 3 systèmes d’équation de créés par défaut. Ces matrices nous servaient de matrices de tests durant le développement, nous n’avions pas besoin de créer manuellement des nouvelles matrices lors de chaque exécution. Nous avons laissé les 5 matrices lors de la remise pour te faciliter la tâche de test à toi aussi. Il y va de même pour les systèmes d’équations par défaut, ils sont là pour nous permettent de tester rapidement nos méthodes.

Voici les 5 matrices :

**Matrice #1**

[ 2, 0, 0 ]

[ 0, 3, 0 ]

[ 0, 0, 4 ]

Cette matrice est utile pour des tests de matrice triangulaire et diagonale.

**Matrice #2**

[ 0, 0, 0 ]

[ 4, 0, 0 ]

[ 0, 2, 0 ]

Cette matrice est utile pour des tests de matrice diagonale strict.

**Matrice #3**

[ 1, 2, -1]

[ -2, 1, 1]

[ 0, 3, -3]

Cette matrice est celle des notes de cours de la section de matrice inverse. Elle nous a permis principalement de tester si on avait le bon résultat pour la comatrice et la matrice inverse.

**Matrice #4**

[ 4, 2, 8, 3 ]

[ 5, 1, 7, 5 ]

[ 8, 0, 8, 5 ]

[ 3, 2, 3, 8 ]

Cette matrice est celle des notes de cours dans la section du déterminant. Elle nous a servi pour tester notre calcul du déterminant ainsi que faire divers tests pour valider que nos méthodes fonctionnaient avec des matrices plus grandes que 3 x 3.

**Matrice #5**

[ 4, 2, 8, 3, 3, 2, 1 ]

[ 5, 1, 7, 5, 3, 2, 1 ]

[ 8, 0, 8, 5, 4, 4, 2 ]

[ 3, 2, 3, 8, 4, 2, 1 ]

[ 6, 7, 3, 0, 1, 1, 3 ]

[ 3, 2, 3, 4, 4, 4, 4 ]

[ 3, 5, 3, 2, 7, 3, 2 ]

Cette matrice permet de tester les méthodes sur des matrices encore plus grosse que 4 x 4.

On peut évidemment créer de nouvelles matrices lors de l’exécution du programme. Cette partie sera couvert dans une section qui suivra.

Voici les systèmes d’équations par défaut :

**Système #1**

2x1 + x2 + 3x3 = 6

x1 - 2x2 + x3 = 2

x1 + x2 - 2x3 = 1

Ce système est celui de base dans les notes pour les exemples des méthodes de Cramer et de l’inversion matricielle. Nous l’avons utilisé pour tester rapidement ces méthodes.

**Système #2**

4x1 - x2 + 0x3 = 100

-x1 + 4x2 - x3 = 100

0x1 - x2 + 4x3 = 100

Ce système est un système qui nous a été utile pour facilement tester la méthode de résolution de Jacobi. Il respecte les critères nécessaires pour pouvoir utiliser la méthode de Jacobi.

**Système #3**

0x1 + 0x2 + 0x3 = 5

4x1 + 0x2 + 0x3 = 2

0x1 + 2x2 + 0x3 = 3

Ce système est un exemple de système dont le déterminant de la Matrice A est nul et qui n’est pas strictement dominante diagonalement. Le système va nous servir à tester que nos méthodes prennent en charge les exceptions à gérer.

**Problèmes et difficultés rencontrés**

Lors de la programmation du programme, nous avons fait face à certaines difficultés. Le premier défi qui nous a poser problèmes était la compréhension de certaines propriétés de la matrice. Précisément, ce sont le calcul du déterminant et celui de la matrice inverse qui nous ont demandé un bon moment de réflexion afin de bien comprendre le principe mathématique.

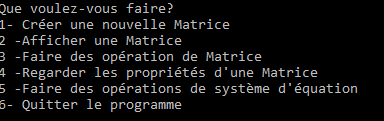
Une autre petite difficulté a été au niveau de l’inclusion de la récursivité nécessaire pour certain calcul. La récursivité n’est pas un concept que nous avons l’occasion de programmer régulièrement alors il y avait un petit manque de pratique de notre part, même si on comprend le concept.

Finalement, la programmation de la méthode de Jacobi nous a laissé un certain temps perplexe mais nous en somme venu à bout plus rapidement que nous pensions une fois que nous avons débloqué.

**Guide d’utilisation détaillé du programme**

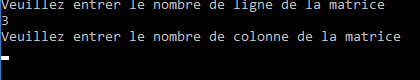
**Menu principal**

Lors du lancement du programme, nous arrivons au menu principal. Les choix disponibles sont affichés dans l’imprime écran qui suit et seront détaillé un a un dans le document. Le menu demande la saisie d’un paramètre integer pour savoir l’action que l’utilisateur souhaite faire. À noter qu’étant donné la portée réduite du projet, aucune validation à savoir quel type de caractère a été saisie n’as été faite. Par contre, si l’utilisateur entre un chiffre autre que ceux proposé, il est averti et doit recommencer son choix. Cela est commun à tous les menus qui suivront.

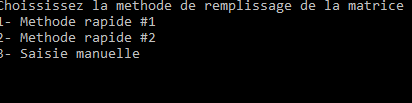


**Créer une nouvelle matrice**

En choisissant de créer une nouvelle matrice, l’utilisateur se voit demandé plusieurs questions. Dans l’ordre, il doit saisir le nombre de lignes et de colonnes de la nouvelle matrice.



Ensuite, l’utilisateur se voit demandé de choisir la méthode voulu pour remplir la matrice. En effet, nous avons fait deux méthodes qui remplit automatiquement la matrice. Les chiffres saisit dépendent de la ligne et la colonne de la position. Cela a été fait de manière à pouvoir créer rapidement de nouvelles matrices pour nos tests. Finalement, l’utilisateur peut saisir manuellement les données de sa matrice.

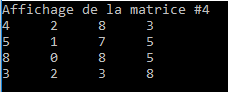
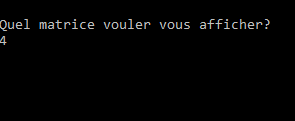


Une nouvelle matrice créé est assigné un numéro de matrice. Ce numéro incrémente selon le nombre de matrices existantes dans le système. Comme il y a par défaut 5 matrices au départ, la première nouvelle matrice créer est assigné le numéro 6. Cette id nous permet d’accéder à la matrice de notre choix lorsque on demande de saisir une matrice.



**Affichage d’une matrice**

L’utilisateur a la possibilité d’afficher le contenu de n’importe quelle matrice. Dans cet option, il suffit de saisir le numéro de la matrice que on souhaite afficher. Son contenu apparaitra alors dans la console. Si la matrice n’existe pas, l’utilisateur sera averti.



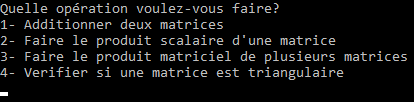
Cette méthode de saisie pour choisir une matrice est utilisé à de nombreux endroit dans le programme. Elle ne sera pas mentionnée à chaque section.

La méthode d’affichage de matrice est utile pour savoir le contenu initiale d’une matrice. Cette même méthode est appelée lors des diverses autres actions possibles pour afficher le résultat des opérations sur les matrices.

**Faire des opérations de matrices**

Cette section comprend les différentes méthodes de la classe matrice. Grossièrement, cette section comprend les calculs qui demande une saisie de paramètres. De ce fait, EstTriangulaire est dans cette section même si dans son identité, cela serait plus une propriété de la matrice. EstTriangulaire était également listé sous « Opérations » dans le document de l’énoncé du travail.

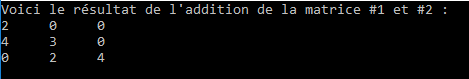
Voici le menu des opérations :



Nous pouvons voir les différentes opérations possibles. Elles seront analysées une à une de manière détaillée, dans leur fonctionnement au niveau du programme mais les concepts mathématiques ne seront pas expliqués dans ce document.

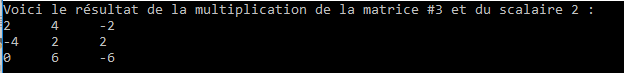
**Additionner deux matrices**

Cette opération effectue une addition de deux matrices. Celles-ci doivent être saisie par l’utilisateur, de la même manière que pour l’affichage d’une matrice. Le résultat est alors affiché à l’écran.



**Faire le produit scalaire d’une matrice**

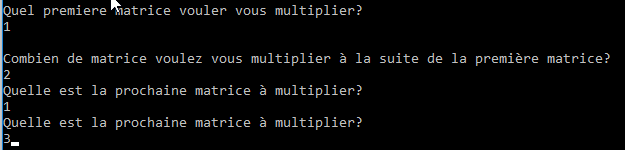
Pour cette opération, nous devons saisir la matrice qui sera multiplié. Ensuite l’utilisateur doit saisir le scalaire qui multipliera cette matrice. Le résultat est affiché ensuite à l’écran.



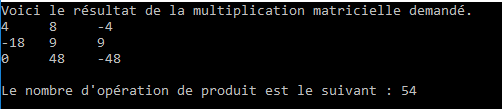
**Faire le produit matriciel de plusieurs matrices**

Dans cette section, il est offert à l’utilisateur de faire des multiplications matricielles entre plusieurs matrices. L’utilisateur peut choisir autant de matrice qu’il souhaite, tant que celles-ci respecte les règles pour les multiplications matricielles tel que le fait que le nombre de colonnes de la première matrice soit égal aux nombres de lignes de la deuxième matrice. Dans le cas d’un produit de plusieurs matrices, c’est le nombre de colonnes de la matrice résultante des opérations déjà fait qui doit correspondre au nombre de ligne de la nouvelle matrice à multiplier.

En premier lieu, l’utilisateur doit choisir la matrice de départ. Ensuite, il doit saisir le nombre de matrice qu’il souhaite multiplier à la suite. S’il saisit 1 il n’aura qu’à saisir une matrice de plus mais s’il saisit 2 ou plus le système lui demandera de saisir toutes les matrices souhaitées avant d’effectuer le calcul. Les matrices subséquentes sont ajoutées à mesure dans un tableau de matrices et c’est ce tableau qui est passé en paramètre à la méthode de la première matrice. La validation sur le format des matrices se fait à mesure de la saisie des matrices.

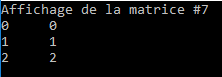


Sur l’imprime écran si haut on multiplie la matrice #1 par elle-même et le résultat de cette opération est alors multiplié par la matrice #3. Le résultat est le suivant :



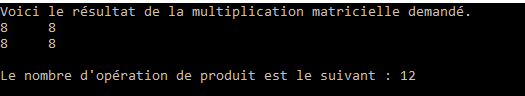
À noter, on affiche le nombre d’opération de produit pour la multiplication complète.

Pour bien démontrer la validation des nombres de lignes et colonnes, créons une matrice 2 x 3 et une autre 3 x 2. Respectivement matrice #6 et matrice #7.

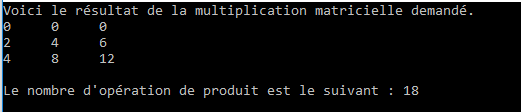


Maintenant, tentons les opérations suivantes :

Matrice #6 X Matrice #7

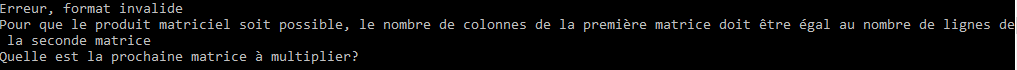


Matrice #7 X Matrice #6



Dans ses deux cas, la multiplication est permise mais les matrices résultantes ne sont pas du même format. Le nombre d’opération de produit est également différent.

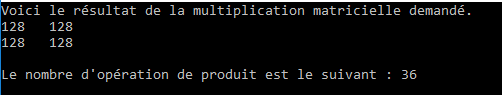
Matrice #7 X Matrice #7



Dans un cas où la matrice saisit n’est pas conforme aux règles de la multiplication, on nous demande de saisir à nouveau la prochaine matrice à multiplier. C’est une des seuls endroits que on redemande la saisit suite à une erreur au lieu de retourner au menu de base. La raison étant que si on a une longue multiplication de matrice à saisir et qu’on se trompe vers la fin, on ne voulait pas avoir à tout ressaisir à nouveau. On demande la saisie jusqu’à ce que le format de la nouvelle matrice soit valide.

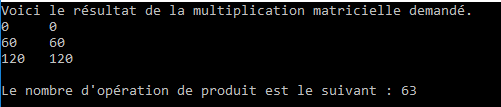
Regardons finalement un exemple de multiplication à plus de 3 matrices avec différents scénarios d’associativités possibles. Considérons en premier lieu le scénario suivant :

Matrice #6 X Matrice #7 X Matrice #6 X Matrice #7



Considérons maintenant le scénario suivant :

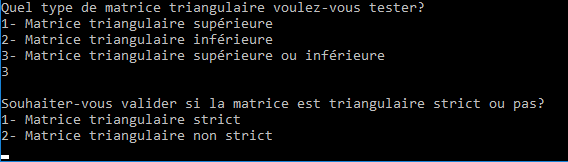
Matrice #7 X Matrice #6 X Matrice #1 X Matrice #7



On constate que diverses combinaisons de format de matrices est possible pour la multiplication.

**Matrice triangulaire**

Ici, l’utilisateur doit saisir la matrice qu’il souhaite vérifier de la même façon que lors des autres saisies. Ensuite, il se voit demandé de choisir différentes options de vérification. Ces options sont affichées dans l’imprime écran suivant.

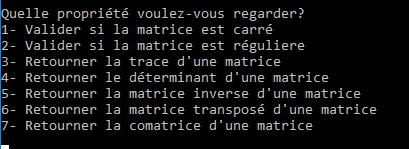
****

Le programme prend ensuite en compte les paramètres saisis et retourne si la matrice est triangulaire selon ces paramètres.



**Regarder les propriété d’une matrice**

Cette section comporte les différentes propriétés des matrices. On peut conclure que les propriétés sont les questions que l’on souhaite vérifier sur les matrices qui ne demande pas de saisie de paramètre.



**Matrice carrée et matrice régulière**

Ces propriétés retournent simplement un boolean qui dit si la matrice saisie respecte ou non la propriété.



Évidemment, il y a une validation à savoir si la matrice est carrée dans le cas de la matrice régulière. Un message d’erreur apparait si la matrice saisie n’est pas carrée.

**Trace de la matrice**

L’utilisateur doit saisir la matrice dont il veut calculer la trace. La trace est alors affichée à la console. La matrice saisie doit être carrée.

C:\Users\Vincent\AppData\Local\Microsoft\Windows\INetCache\Content.Word\23 trace.png

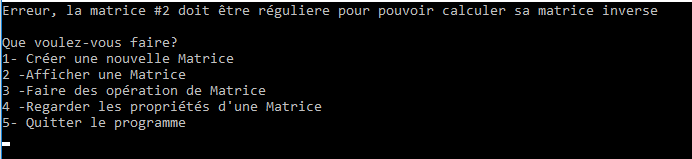
**Déterminant de la matrice**

Tout comme la trace, l’utilisateur n’a qu’à saisir la matrice dont il veut savoir le déterminant. La matrice saisie doit être carrée.

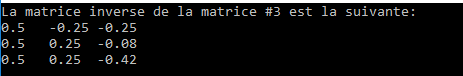
C:\Users\Vincent\AppData\Local\Microsoft\Windows\INetCache\Content.Word\24 det.png

**Matrice inverse**

Pour pouvoir calculer la matrice inverse d’une matrice, celle-ci doit être régulière. Si elle ne l’est pas, il y aura un message d’erreur, de même que si elle n’est pas carrée.

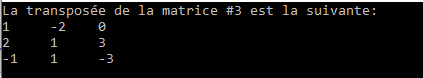


Si toutefois elle est régulière, le programme calculera et affichera sa matrice inverse.



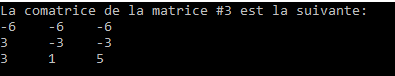
**Transposée d’une matrice**

Dans cette section, l’utilisateur doit saisir de la même manière que partout ailleurs la matrice dont il souhaite afficher la transposée. Le programme calculera et affichera cette transposée dans la console.



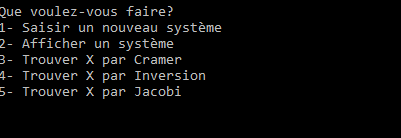
**Comatrice**

Finalement, ici l’utilisateur se doit de saisir la matrice dont il souhaite retourner la comatrice. Le programme s’assurera que la matrice est carrée et si oui il affichera la comatrice voulue.



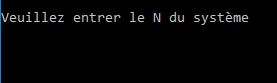
**Système d’équations**

Dans cette section nous allons couvrir toutes les options sur les systèmes d’équations. Toutes les opérations sont regroupées ici pour plus de simplicité. Pour accéder à ce sous-menu, l’utilisateur doit saisir « 5 » lorsqu’on lui demande ce qu’il veut faire. On accède alors au menu suivant.

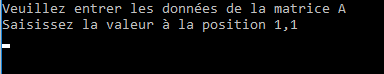


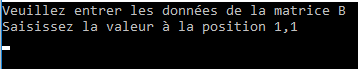
**Saisir un nouveau système**

Dans cette section, l’utilisateur à l’option de saisir un nouveau système d’équation. Un système d’équation est composé d’une matrice carré nommé matrice A de format [N,N] et d’une matrice B de format [N,1].



Premièrement, on doit saisir le N du système. Ce N déterminera la taille de la matrice carrée A ainsi que le nombre d’élément dans la matrice B.



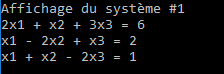


Ensuite l’utilisateur se doit de saisir les données contenues dans les matrices du système tel qu’illustré ci-dessus. Le système est alors créé.



**Afficher un système**

L’utilisateur a la possibilité d’afficher le contenu de n’importe quel système. Dans cet option, il suffit de saisir le numéro du système que on désire afficher. Son contenu apparaitra alors dans la console sous forme d’expression. Si la matrice système demandé n’existe pas, l’utilisateur sera averti.

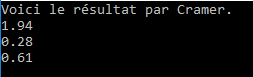


**Trouver X par Cramer**

Si l’utilisateur décide de trouver X par Cramer, le programme demandera à celui-ci de saisir le numéro du système qu’il désir résoudre.



Une fois le numéro du système saisi, le résultat de l’équation s’affiche dans la console si le déterminant de la matrice A du système saisi n’est pas nul.



Dans le cas où le système saisi comprend une matrice A dont le déterminant est nul, le programme va avertir l’utilisateur et on ne pourra trouver le X par Cramer.

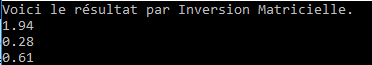
****

**Trouver X par Inversion Matricielle**

L’utilisateur peut également trouver le X par inversion matricielle. Si on choisit cette option, cela suit le même principe que pour la méthode de Cramer. On se fait tout d’abord demander de saisir le système voulu.



Ensuite, selon si le déterminant de la matrice A de notre système saisi est nul ou pas, le programme nous affiche soit le résultat, soit un message d’erreur.



****

**Trouver X par Jacobi**

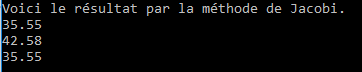
Pour conclure, l’utilisateur a comme dernière option de trouver X par la méthode de Jacobi. Encore une fois, on lui demande de saisir le système souhaité.



Par la suite, on lui demande de saisir l’epsilon qui sera utilisé dans la méthode pour déterminer quand cesser le calcul. L’epsilon est de format Double et se doit d’être écrit avec une virgule et non un point si c’est un nombre à virgule.



Le résultat de la méthode est alors affiché dans la console. L’affichage se fait par la même méthode d’affichage qui est utilisé ailleurs dans le programme. C’est pourquoi que seulement 2 chiffre après la virgule est affiché, mais nous jugeons que c’est représentatif.



Si la matrice A du système n’est pas strictement dominante et que nous tentons d’utiliser la méthode de Jacobi, un message d’erreur est alors affiché.



Cela conclu notre guide d’utilisation. Nous sommes d’avis que le programme est facile de prise en main et qu’avec ce document, un utilisateur devrait pouvoir sans problème effectuer les opérations qu’il souhaite en lien avec les matrices et les systèmes d’équations.