

Clément MILVILLE
Edouard SIMON

2ème année
Groupe N

Projet tutoré de Mathématiques

Résolution de matrices

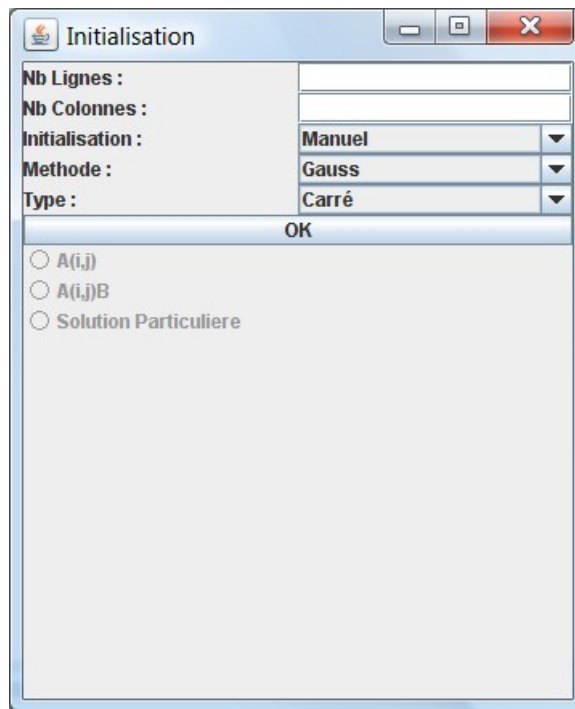


Sommaire

I. Manuel d'utilisation du logiciel.....	4
I. 1. L'interface graphique.....	4
I. 2. Utilisation du logiciel.....	5
II. Explications du code source.....	7
II. 1. Les différents objets.....	7
1. a La classe Matrix.....	7
1. b La classe Equation.....	8
1. c La classe Resolution.....	9
1. d La classe MainPanel.....	10
II. 2. Déroulement du programme.....	12

I. Manuel d'utilisation du logiciel

I. 1. L'interface graphique



Cette interface est composée de :

-Deux zones de saisie, qui vous permettent de saisir les paramètres de votre matrice à savoir sa longueur et sa largeur.



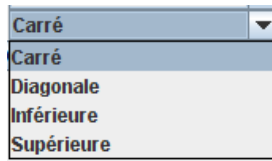
-D'un menu déroulant qui permet de choisir le mode de remplissage de la matrice : Manuel, Aléatoire (généralisé automatiquement) ou à l'aide d'une équation.



-D'un deuxième menu déroulant permettant de choisir le théorème de la résolution à savoir Gauss ou Jordan.

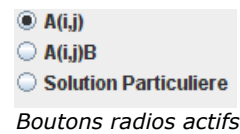
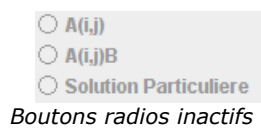


-D'un troisième et dernier menu déroulant laissant le choix à l'utilisateur du type de matrice qu'il désire utiliser: Carré, diagonale, triangulaire supérieure et triangulaire inférieure.



- D'un bouton « OK » permettant de valider vos choix et de générer la matrice.

- De 4 boutons radio qui seront utilisés une fois vos choix validés pour afficher $A(i, j)$, $A(i, j)B$ et la solution particulière. Si vous n'avez pas encore validé votre système, ces boutons ne sont pas actifs.



I. 2. Utilisation du logiciel

Une fois que vous avez initialisé votre matrice à l'aide du menu précédent, si le mode Manuel à été choisi vous obtiendrez une fenêtre de ce type:

Ce panel vous affiche une matrice de la taille désirée et vous permet d'effectuer une saisie manuelle de la matrice A et du vecteur B. Une fois remplie cliquez sur le bouton « Valider » pour effectuer les calculs.

L'affichage de votre système linéaire à résoudre apparaît alors à la place.

$$\begin{matrix}
 \mathbf{A} & \begin{bmatrix} 34.0 & 33.0 & 4.0 \\ 19.0 & 14.0 & 41.0 \\ 81.0 & 84.0 & 86.0 \end{bmatrix} & \cdot \mathbf{X} & = & \mathbf{B} & \begin{bmatrix} 10.0 \\ 24.0 \\ 85.0 \end{bmatrix}
 \end{matrix}$$

Vous pouvez maintenant utiliser les boutons radio :

A(i,j)

Sélectionnez ce bouton pour afficher la solution générale ainsi que la matrice $A(i,j)$, le rang de A, I, J, !I et !J.

Solution générale: X =

-1.491175366128425
1.7574164476154692
0.6763049192639878

A(i,j) =	34.0	0.0	0.0	Rang(A) = 3 I = { 1 2 3 } J = { 1 2 3 } !I = { } !J = { }
	0.0	-4.4411764705882355	0.0	
	0.0	0.0	123.45033112582787	

A(i,j)B

Sélectionnez ce bouton pour afficher la matrice $A(i,j)B$, le rang(A,B) ainsi que l'existence des solutions

Solution Particuliere

Si vous selectionnez ce bouton, la solution particulière $A(i,j)X_j=B_i$ s'affiche ainsi que la solution X.

II. Explications du code source

II. 1. Les différents objets

1. a La classe Matrix

Cette classe est la représentation d'une matrice, contenant les méthodes principales de calculs matriciels détaillées ci-dessous.

- Constructeur aléatoire de la classe. Ce constructeur génère une matrice aléatoire à partir de sa taille. Il prend donc en paramètre le nombre de lignes de la matrice et le nombre de colonnes.
- Constructeur de la classe à partir d'une matrice qui prend en paramètre une matrice.
- *getIndices* Méthode d'accès sur les indices grâce au numéro d'index de l'indice désiré. *GetIndices* retourne l'indice correspondant.
- *getRow* Méthode d'accès donnant le nombre de lignes de la matrice.
- *getCol* Méthode d'accès donnant le nombre de colonnes de la matrice.
- *getVal* Méthode d'accès à une valeur de la matrice grâce au numéro de la ligne et au numéro de la colonne. Retourne la valeur à ces coordonnées.
- *setVal* Méthode d'initialisation d'une cellule de la matrice grâce au numéro de la ligne et au numéro de la colonne et à la nouvelle valeur à mettre à ces coordonnées.
- *getIndices* Méthode d'accès sur indices. Retourne les indices de la matrice.
- *setIndices* Méthode d'initialisation sur les indices prenant en paramètre les nouveaux indices à modifier.
- *getMatrice* Méthode d'accès sur la matrice, retournant la matrice courante.
- *getRang* Méthode d'accès sur rang, retournant le rang de la matrice courante.
- *setRang* Méthode d'initialisation du rang prenant en paramètre le nouveau rang à modifier.
- *getAIJ* Méthode d'accès sur la matrice $A(i, j)$.

- *setAIJ* Méthode d'initialisation sur la matrice $A(i, j)$ prenant la matrice $A(i, j)$ en paramètre à modifier.
- *coef* Méthode permettant de trouver le coefficient multiplicateur à 0 de 2 valeurs placées en paramètre.
- *multiplieMatrice* Méthode permettant de multiplier une matrice quelconque à une matrice d'une colonne. Retourne une matrice contenant la solution.
- *calculsPivotJordan* Méthode permettant de faire les calculs sur les différentes lignes de la matrice, en dehors de celle du pivot, avec la méthode de Jordan.
- *calculsPivotGauss* Méthode permettant de faire les calculs sur les différentes lignes de la matrice, en dehors de celle du pivot, avec la méthode de Gauss.
- *resoudSoluceParticuliere* Méthode permettant de résoudre un système d'équations du type $A(i, j) * X = B$ et retournant les solutions X .
- *ajouterVecteur* Méthode permettant d'ajouter le vecteur à la fin d'une matrice. Elle prend en paramètre le vecteur à ajouter à la matrice.
- *inverseCol* Méthode permettant d'inverser deux colonnes de matrice qui lui sont données en paramètres.
- *decalage* Méthode permettant d'inverser les colonnes en cas de pivot nul. le paramètre est la ligne du pivot.
- *toString* Méthode permettant d'avoir une représentation d'une matrice sur le terminal, en mode texte.

1. b La classe Equation

Cette classe permet de résoudre les systèmes d'équations.

- Ce constructeur crée un système d'équations. Les equations sont fournies par une matrice placée en paramètre. Celle-ci contient une équation par ligne, ayant chacune autant d'inconnus que de nombre de colonnes-1. La dernière colonne est l'égalité de chaque équation.
- *resoudreEquation* Méthode permettant de résoudre un système d'équation en utilisant un boolean en paramètre pour indiquer si c'est une résolution de matrice appliquée à Jordan ou Gauss. Elle retourne un tableau de double, correspondant aux solutions de chaque inconnus dans l'ordre.

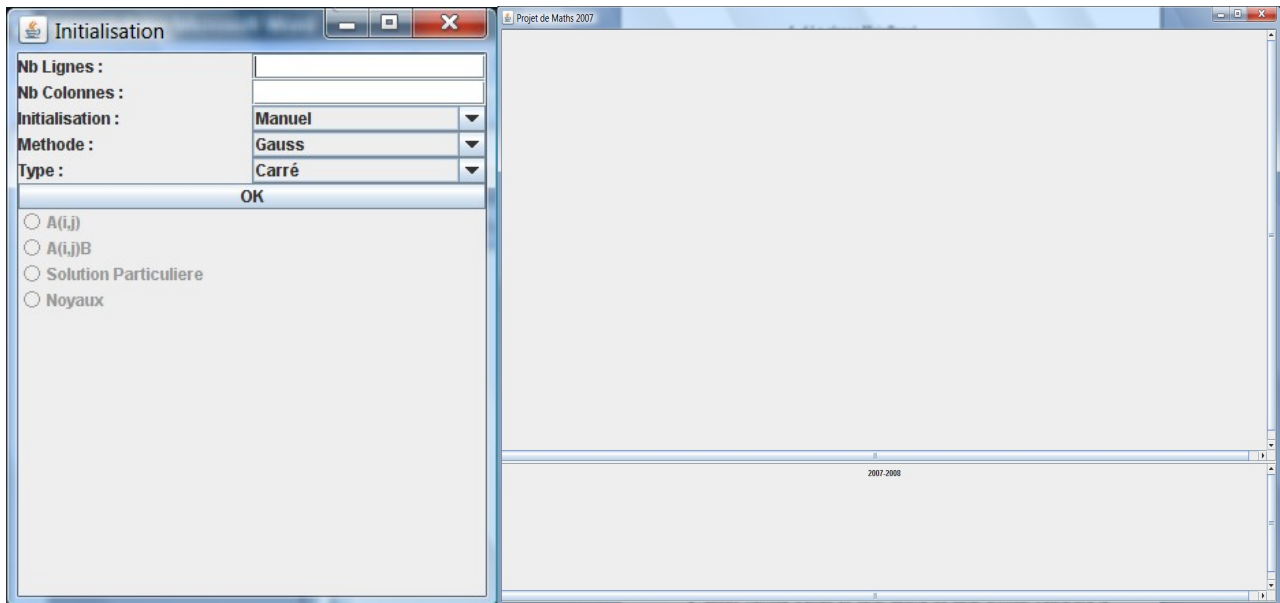
1. c La classe Resolution

Cette classe permet de résoudre les systèmes linéaires, avec la méthode de Jordan, ou Gauss.

- Le constructeur de la classe utilise comme paramètres la matrice A et le vecteur B de l'équation.
- *jordan* Méthode permettant d'effectuer Jordan sur une matrice. Elle retourne le système d'équations obtenu.
- *gauss* Méthode permettant d'effectuer Gauss sur la matrice (A,B). Elle retourne le système d'équations obtenu.
- *initIndices* Méthode permettant d'initialiser les tableaux d'entiers contenant les indices colonnes de la matrice A.
- *soustractionVecteurs* Méthode permettant de soustraire un vecteur à un autre. Elle prend 2 vecteurs en paramètre et retourne le résultat de la soustraction.
- *getMaxVector* Méthode permettant de récupérer le résidu d'un vecteur. Elle utilise le vecteur duquel on tirera le résidu et le retourne.
- *getAB* Méthode d'accès sur AB. Elle retourne une matrice contenant A et B.
- *getA* Méthode d'accès sur A, retournant la matrice A.
- *getB* Méthode d'accès sur B, retournant la vecteur B.

1. d La classe MainPanel

Classe du panel principal, celui qui contient la fenêtre d'initialisation ainsi que toutes les fenêtres permettant l'affichage des résultats.



- Constructeur de la classe utilisant comme paramètres la largeur et la hauteur du panel. Ici nous utilisons la dimension de l'écran pour l'affichage graphique en plein écran.
- *setRes* Méthode d'initialisation du résultat calculé.
- *getRes* Méthode d'accès au résultat. Elle retourne le résultat pour que l'ensemble des panels y ait accès.
- *getLeft* Méthode d'accès au panel gauche du panel principal c'est-à-dire le menu utilisateur.
- *getRight* Méthode d'accès au panel droit du panel principal c'est-à-dire le panneau d'affichage.
- *getMatrice* Méthode d'accès sur la matrice qui retourne la matrice initiale.
- *getVecteur* Méthode d'accès sur le vecteur qui retourne le vecteur initiale.
- *dimensionMatriceAleatoire* Méthode permettant de demander la taille (sans limitations) de la matrice. Cette méthode initialisera les matrices aléatoirement. Elle retourne la liste qui contiendra la matrice A et le vecteur B.
- *afficheMatrice* Méthode permettant de créer un panel contenant une matrice à afficher. Elle prend en paramètre la matrice à afficher et retourne le panel contenant la matrice.
- *afficheSolutionComparatif* Méthode permettant d'afficher les solutions de Jordan et de Gauss lors du comparatif. Elle prend en paramètre un tableau contenant

les solutions de Jordan et un pour les solutions de Gauss.

- *affichageInconnu* Méthode permettant d'afficher la matrice des inconnus pour le vecteur X. Elle prend en paramètre le nombre d'inconnus et retourne le panel d'affichage de la matrice des inconnus.
- *afficheVecteur* Méthode d'affichage du vecteur B du système. Elle utilise le vecteur B à afficher et retourne le panel contenant le vecteur.
- *afficheInfos* Méthode permettant de récupérer les informations du système linéaire, c'est-à-dire la forme $A \cdot X = B$. Elle récupère la matrice et le vecteur saisis pour retourner le panel contenant les informations.
- *afficheInfoSoluceParticuliere* Méthode permettant de créer un panel contenant les informations pour la recherche d'une solution particulière. Elle retourne le panel contenant les informations.
- *afficheSoluceParticuliere* Méthode permettant de créer un panel contenant la solution particulière. Elle prend en paramètre la matrice (A,B) et le tableau des valeurs des solutions. Elle retourne le panel contenant le resultat.
- *afficheSolutionGenerale* Méthode permettant de récupérer un panel contenant l'affichage de la solution finale. Elle utilise :
 - La matrice A
 - La liste contenant la solution particulièreEt retourne le panel contenant les solutions générales.
- *noyau* Méthode permettant d'ajouter les équations du noyau, cette méthode utilise une matrice A. et un tableau des indexes colonnes de la matrice A. Elle retourne la liste de tous les vecteurs noyaux.
- *afficheVecteurNoyau* Méthode permettant de récupérer un panel contenant les vecteurs noyau en utilisant la matrice A et la liste contenant les vecteurs noyau.
- *afficheRang* Méthode permettant de récupérer le panel contenant la matrice $A(i,j)$ et le rang d'une matrice A en utilisant la matrice A. et $A(i,j)$
- *afficheIndices* Méthode permettant d'avoir le panel contenant les informations sur les indices lignes et colonnes. Elle utilise une matrice sur laquelle on souhaite avoir les indices.
- *rechercheSolution* Méthode permettant de créer un panel vérifiant l'existence de solutions en utilisant la matrice (A,B) et le vecteur B.
- *existenceSolutions* Méthode permettant de créer un panel contenant l'information sur l'existence de solutions.
- *infoPane* Méthode permettant de créer un panel contenant $A \cdot X = B$.
- *jordanAPane* Méthode permettant de créer un panel contenant le résultat de jordan appliqué sur la matrice $A(i, j)$ passée en paramètre.
- *jordanABPane* Méthode permettant de créer un panel contenant le résultat de la méthode de jordan appliquée sur la matrice $A(i, j)B$ passée en paramètre.

- *solutionParticulierePane* Méthode permettant de créer un panel contenant une solution particulière.
- *noyauPane* Méthode permettant de créer un panel contenant les équations du noyau.
- *solutionGeneralePane* Méthode permettant de créer un panel contenant les solutions générales.

II. 2. Déroulement du programme

Après avoir généré la matrice à l'aide de la fenêtre d'initialisation et après l'avoir remplie, les calculs sont effectués par les méthodes expliquées précédemment.

L'affichage des résultats se fait dans le panneau droit (RightPane), vous pouvez choisir le type de résolution à l'aide des boutons situés dans la fenêtre d'initialisation.