Question individualisée n°7

Compte Rendu

Polytech Tours

MARLIN Alexis, GUILLOT Matthieu

[Analyse du sujet 2](#_Toc9126991)

[Ajout à la librairie CMatrice 3](#_Toc9126992)

[Code 3](#_Toc9126993)

[ CMatriceExtractor 3](#_Toc9126994)

[ CMatriceGaussJordan 11](#_Toc9126996)

[ Ajout à CMatrice 13](#_Toc9126997)

[Tests 17](#_Toc9127000)

[ CArc 17](#_Toc9127001)

[ CSmmet 18](#_Toc9127002)

[18](#_Toc9127003)

[ CGraphe 19](#_Toc9127004)

[Programme 21](#_Toc9127005)

# Analyse du sujet

La demande formulée par le client consiste en l’ajout à la librairie CMatrice d’une fonctionnalité permettant d’inverser une matrice si celle-ci est carrée. La méthode de calcul de l’inverse d’une matrice proposée est celle de Gauss-Jordan. Le rendu doit comporter un projet Visual Studio (application console win32) ainsi qu’une version compilée avec le profil « Release ».

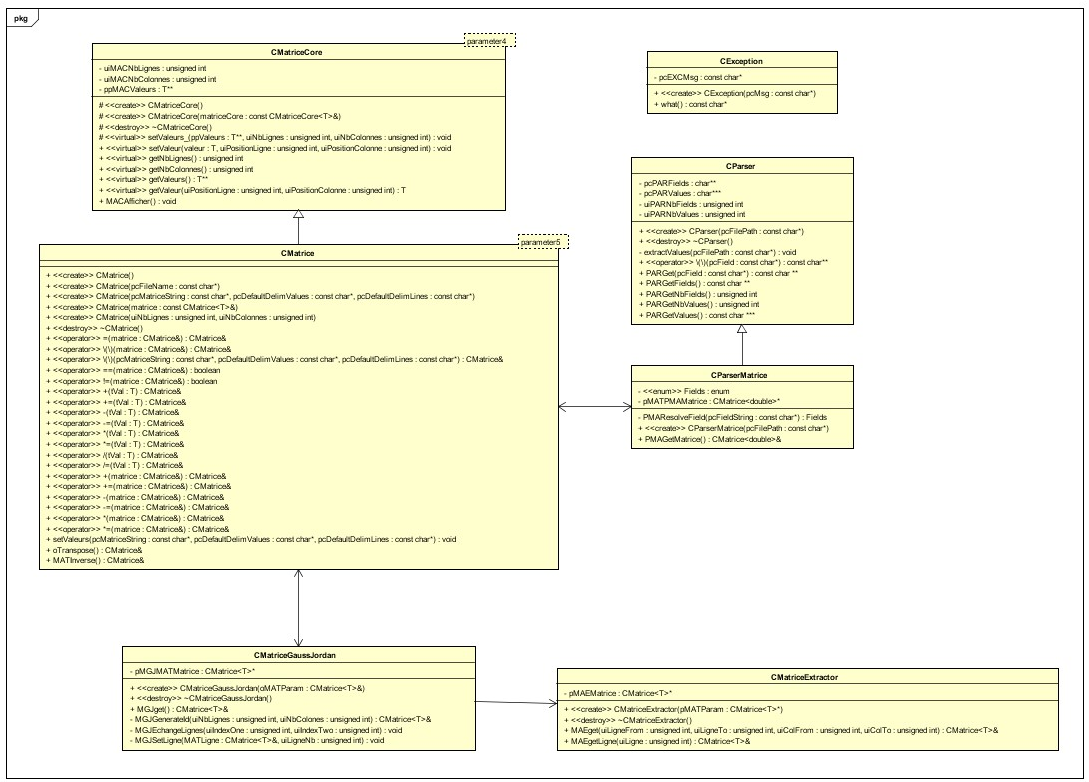
La méthode proposée consiste à prendre la matrice à inverser M et de la combiner à une matrice identité de même taille que M. On appelle cette matrice résultante une matrice augmentée. Il faut ensuite effectuer des opérations sur les lignes de la matrice augmentée de sorte à transformer la partie de M en matrice identité. On doit ensuite extraire la partie de la matrice identité qui est maintenant la matrice inverse M-1.

Afin de parvenir à cela, nous avons ajouté à la librairie CMatrice deux classes, CMatriceGaussJordan et CMatriceExtractor, ainsi qu’une méthode à la classe CMatrice.

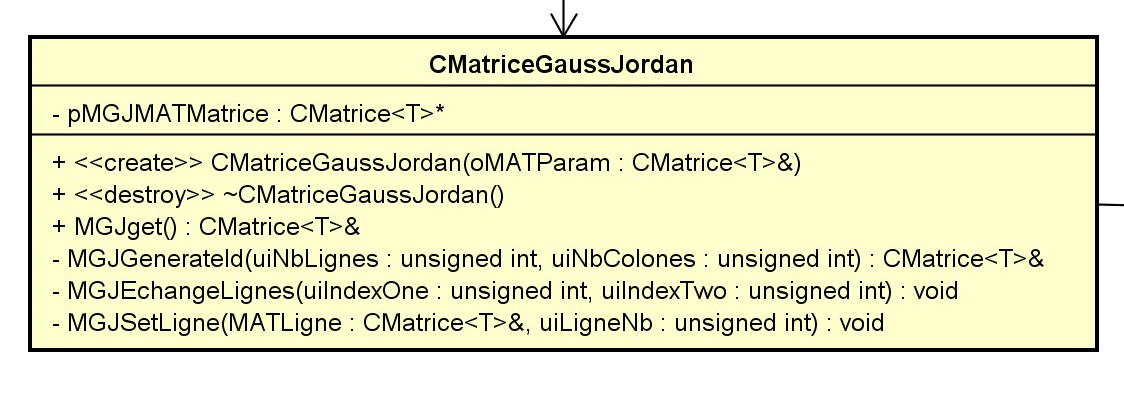
Afin d’extraire les informations d’un fichier une classe parser est requise. Afin de comprendre les données extraites du fichier un parser spécifique à ce projet est requis. Afin de gérer les exceptions une classe spécifique est également nécessaire. Ces classes étaient présentes dans la librairie CMatrice et ne sont pas des ajouts de ce projet en particulier.

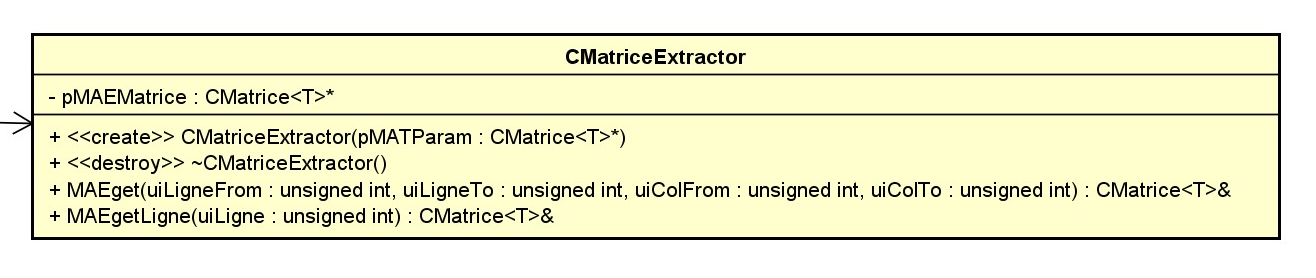
Dans le sujet il n’est pas spécifié les types de matrice que l’on peut inverser.

Le diagramme de classe (disponible dans le dossier « docs ») résultant de cette analyse est le suivant :



Les classes implémentés au cours de ce projet sont :





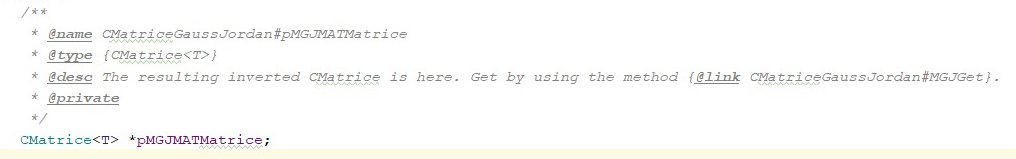
# Librairie CMatrice

## Code

Dans un premier temps les classe CMatriceGaussJordan et CMatriceExtractor :

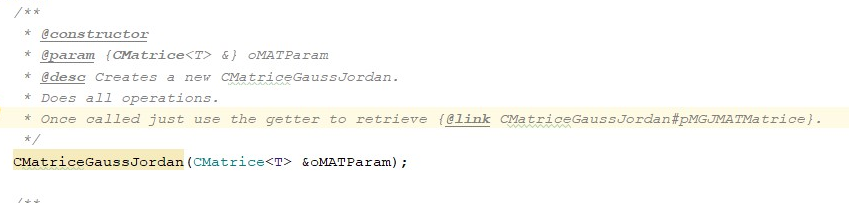
* CMatriceGaussJordan :

La classe CMatriceGaussJordan est la classe permettant d’inverser une matrice à partir d’un objet CMatrice. Elle possède un attribut privé, pMGJMATMatrice :



Cet attribut sert à stocker la matrice manipulée dans la classe CMatriceGaussJordan.

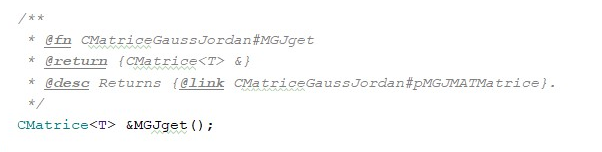
Concernant le constructeur :



C’est lui qui va créer la matrice augmentée et effectuer toutes les opérations permettant d’inverser l’objet CMatrice passé en paramètre par référence. Le choix de faire effectuer toutes les opérations au constructeur est souhaité, permettant ensuite l’accès direct à la matrice inversée.

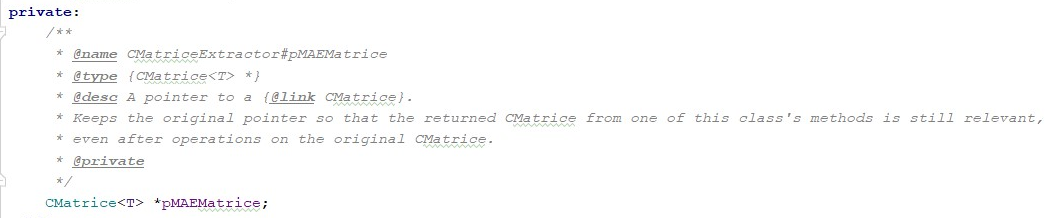
La classe possède des méthodes **privées** permettant d’échanger deux lignes et attribuer des nouvelles valeurs à une ligne de la matrice attribut de cette classe.

Finalement, la méthode MGJGet permet de récupérer la matrice inversée :

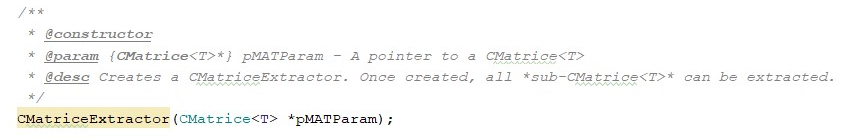


* CMatriceExtractor :

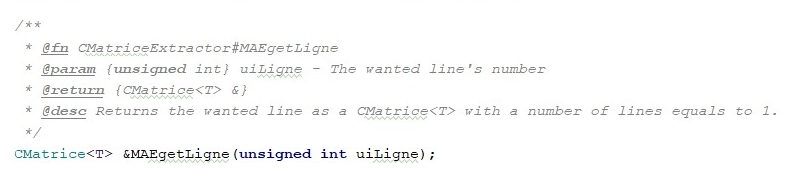
La classe CMatriceExtractor permet d’extraire un objet CMatrice d’un autre objet CMatrice. Dans notre cas elle va servir à extraire la matrice inverse M-1 de la matrice augmentée. Elle possède un attribut privé, pMAEMatrice, un pointeur sur un objet CMatrice:



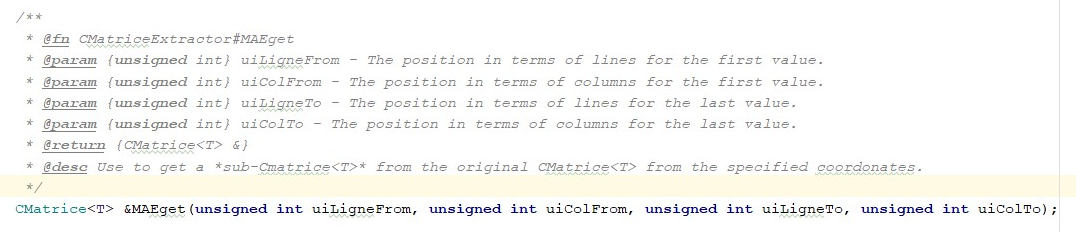
Le constructeur prend en paramètre un pointeur sur un objet CMatrice, qui sera enregistré dans l’attribut de cette classe après allocation. Si le pointeur est souhaité ici c’est pour permettre de lier l’instance de CMatriceExtractor à un objet CMatrice, permettant de faire appel aux méthodes de cette classe de façon relevante même après opérations sur ladite matrice.



Les méthodes définies dans cette classe correspondent à des opérations d’extraction de contenu de la matrice. MAEgetLigne permet d’extraire une ligne de la matrice attribut de cette instance, à partir du numéro de ligne passé en paramètre:

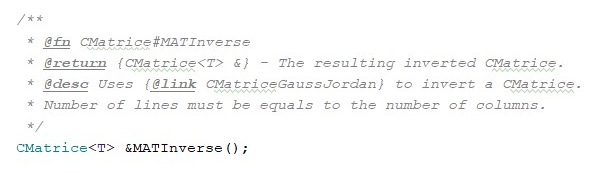


Finalement, la méthode MAEget permet d’extraire une matrice de la matrice stockée dans l’attribut pMAEMatrice. En paramètre de la méthode sont passés les numéros de ligne « from » et « to », de même pour les numéros de colonne :



* Ajout à CMatrice :

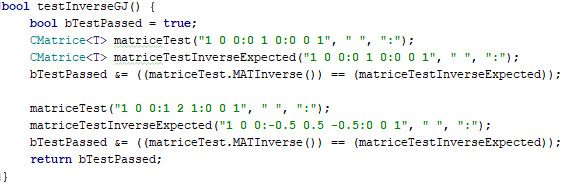
L’ajout effectué dans la classe CMatrice est la création de la méthode permettant d’inverser la matrice, MATInverse. C’est cette méthode qui doit être appelé afin d’inverser la matrice :



## Tests

Dans les tests ont été rajoutés une partie permettant le test de l’inversion d’une matrice.

* CMatriceGaussJordan :

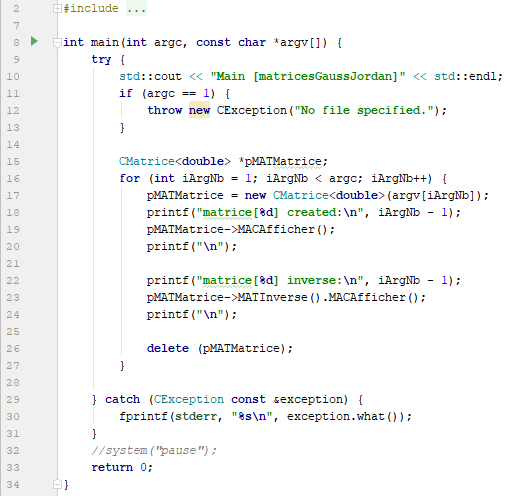


Documentation 27

Le code teste la bonne inversion des matrices passés en paramètre. Etant donné CMatrice est testé, on considère que toute initialisation de l’objet CMatrice est équivalente, et que toutes les méthodes associées fonctionnent.

# Programme

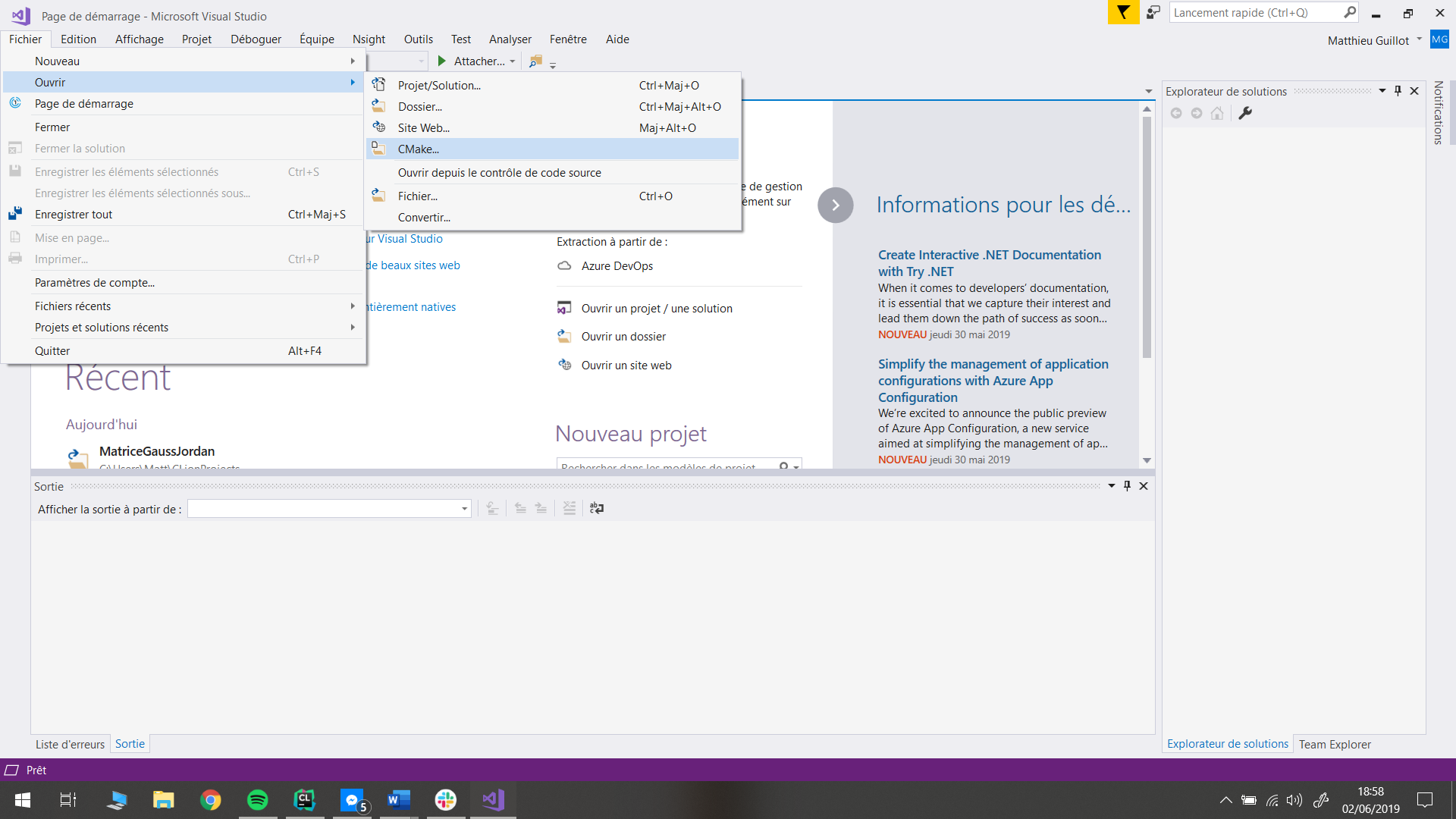
Le programme généré utilisant la librairie CMatrice est le suivant :



Il parcourt tous les fichiers passés en paramètre, en extrait la matrice correspondante et essaye de l’inverser. Si un seul des fichiers décrit une matrice non-carré ou aucun fichier n’est précisé, une exception sera lancée, le programme ne fonctionnera pas.

# Notes

Le projet à été réalisé sous CMake. Afin de l’ouvrir sous Visual Studio il suffit d’ouvrir le projet comme ci-dessous :



Les fichiers .exe se trouvent respectivement dans les dossiers « cmake-build-debug » et « cmake-build-release » et se nomment tous deux « matricesGJ.exe ».