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)

[ Ajout à CMatrice 3](#_Toc9126994)

[ CMatriceGaussJordan 11](#_Toc9126996)

[ CMatriceExtractor 13](#_Toc9126997)

[ CParser 16](#_Toc9126998)

[ CException 17](#_Toc9126999)

[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.

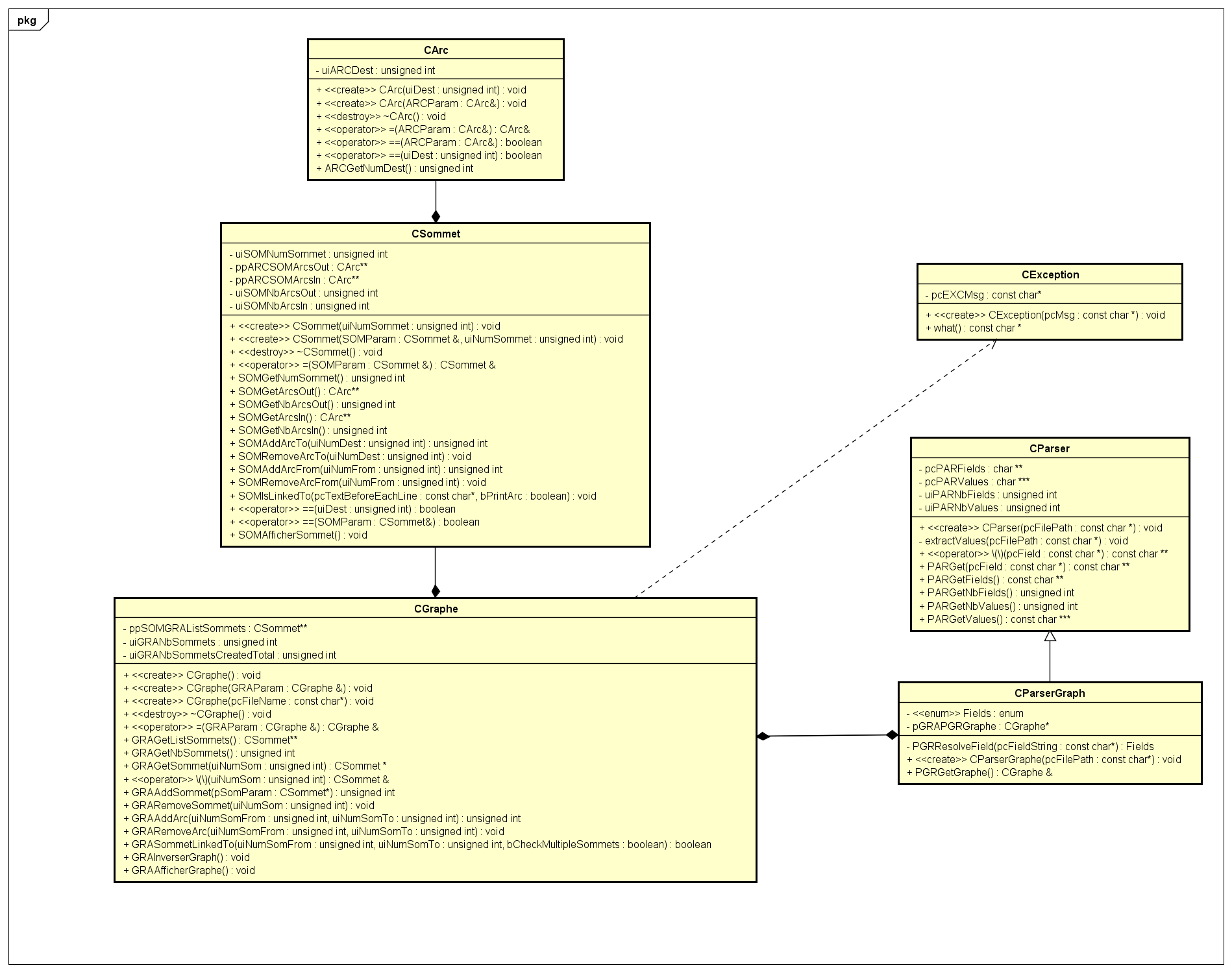
Afin de parvenir à cela, nous avons ajouté à la librairie CMatrice deux classes, CMatriceGaussJordan et CMatriceExtractor, ainsi qu’une méthode à la classe CMatrice.

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 :

**A CHANGER**



# Librairie CMatrice

## Code

Dans un premier temps les classe CMatriceGaussJordan et CMatriceExtractor :

* CMatriceGaussJordan :

La classe CMatriceGaussJordan est la classe permettant … . Elle possède un attribut privé, pMGJMATMatrice :

SCREEN

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

Concernant le constructeur :

SCREEN

C’est lui qui va créer une matrice augmentée et …..

Le destructeur quant à lui ne fait rien.

Afin de gérer les opérations sur les lignes, nous avons les méthodes MGJSetLigne et MGJEchangeLignes :

SCREEN

La méthode MGJSetLigne sert à remplacer une ligne donnée par une matrice ligne passée en paramètre. Elle est utilisée afin de réaliser les opérations sur les lignes. MGJEchangeLignes quant à elle permet d’échanger deux lignes d’une CMatriceGaussJordan.

La méthode MGJGenerateId permet de créer une matrice identité, c’est-à-dire une matrice carrée avec des 1 sur chaque entrée de sa diagonale principale et des 0 sur le reste de la matrice :

SCREEN

Finalement, la méthode MGJGet fait appel à la méthode MACAfficher de CMatriceCore, afin d’afficher la matrice de l’attribut pMGJMACMatrice, puis de renvoyer cette matrice :

SCREEN

* CMatriceExtractor :

## Tests

Dans les tests ont été rajoutés une partie permettant le test de la classe CMatrieGaussJordan.

* CMatriceGaussJordan :

IMAGE

Documentation 27

Le code teste …

# Programme

Le programme généré utilisant la librairie CMatrice est le suivant :

IMAGE

Documentation 33

En utilisant un terminal, on obtient en profil « debug »:

Et en profil « release » :