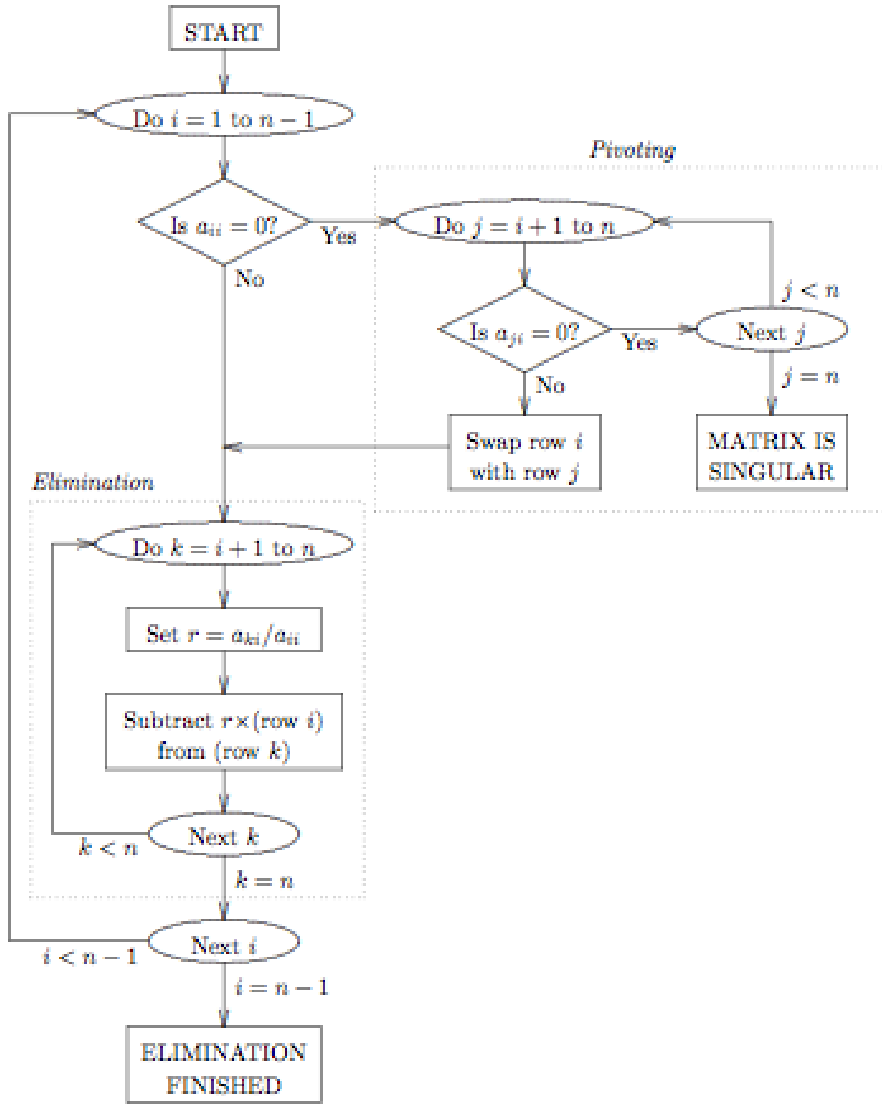
1.5-1.6. Matrices Echelonnées

April 21, 2020

1 Concept(s)-clé(s) et théorie

La **méthode d'élimination de Gauss** est un algorithme central en algèbre linéaire qui consiste à appliquer une séquence appropriée d'opérations élémentaires à une matrice (voir Notebook du chapitre 1.3-4: Notation Matricielle) jusqu'à ce qu'il soit réduit à une forme triangulaire supérieure. La structure de la méthode d'élimination de Gauss est la



suivante:
Considérez donc le système linéaire

$$Ax = b$$

étant A la matrice des coefficients dans $\mathcal{R}^{n\times n}$ et b le terme de vecteur de droite dans $\mathcal{R}^n$; en appliquant la méthode d'élimination de Gauss à la matrice augmentée $A|b$ on obtient:

$$\text{Système Original} \quad A|b \quad \Leftrightarrow \quad \text{Système Triangulaire Supérieur} \quad \tilde{A}|\tilde{b} \quad \Leftrightarrow \quad \text{Système Réduit} \quad I|\hat{b}$$

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right) \Leftrightarrow \left(\begin{array}{cccc|c} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} & \tilde{b}_1 \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_{2n} & \tilde{b}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \tilde{a}_{nn} & \tilde{b}_n \end{array}\right) \Leftrightarrow \left(\begin{array}{cccc|c} 1 & 0 & \dots & 0 & \hat{b}_1 \\ 0 & 1 & \dots & 0 & \hat{b}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \hat{b}_n \end{array}\right)$$

Il est immédiat de remarquer que le système linéaire résultant est équivalent à celui d'origine (puisque seules des opérations matricielles élémentaires ont été utilisées!), mais beaucoup plus facile et plus rapide à résoudre, en procédant à reculons de la dernière équation à la première. La matrice résultante est dite **échelonnée**. De plus, il est possible de réduire additionnellement la matrice (via des opérations élémentaires) jusqu'à ce qu'elle coïncide avec la matrice d'identité I; ce processus est appelé **réduction matricielle** (ou élimination de Gauss-Jordan) et rend le système linéaire résultant trivial à résoudre car sa solution coïncide simplement avec le terme de vecteur de droite $\hat{b}$.

[ ]:
```
import Librairie.AL_Fct as al
import numpy as np
import ipywidgets as widgets
import random

from ipywidgets import interact, interactive, fixed, interact_manual
```

1.0.1 Exercice 1

À l'aide des opérations élémentaires, échelonner et réduire les matrices ci-dessous.

$$\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \qquad \begin{pmatrix} \frac{1}{2} & 3 & 0 \\ 2 & -4 & 6 \\ 1 & 3 & -1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix}$$

[ ]:
```
A=[[1,1], [1,1], [1,1]]
```

[ ]:
```
print('Vous allez échelonner la matrice')
al.printA(A)
[i,j,r,alpha]= al.manualEch(A)
MatriceList=[np.array(A)]
m=np.array(A)
print('\033[1mExecutez la ligne suivante pour effectuer l\'opération choisie␣
↪\033[0m')
```

[ ]:
```
m=al.echelonnage(i, j, r, alpha, A, m, MatriceList)
```

1.0.2 Exercice 2

À l'aide des opérations élémentaires, échelonner et réduire les matrices (augmentée) ci-dessous.

$$A = \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad A = \begin{pmatrix} \frac{1}{2} & 3 & 0 \\ 2 & -4 & 6 \\ 1 & 3 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} \qquad A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

```
[ ]: A=[[1,1,1], [1,1,1],[1,1,1]]
     b =[[1], [1], [1]]
```

```
[ ]: print('Vous allez échelonner la matrice augmenteé')
     al.printAAug(A,b)
     [i,j,r,alpha]= al.manualEch(A)
     MatriceList=[np.array(A)]
     RHSList = [np.array(b)]
     m=np.concatenate((A,b), axis=1)
     print('\033[1mExecutez la ligne suivante pour effectuer l\'opération choisie␣
      ↪\033[0m')
```

```
[ ]: m=al.echelonnage(i, j, r, alpha, A, m, MatriceList, RHSList)
```

1.0.3 VERIFICATION

À l'aide des cellules ci-dessous, vous pouvez entrer la matrice (des coefficients ou augmentée) de votre choix et obtenir une forme échelonnée et sa forme échelonnée réduite.

Pour **les formes échelonnées** on utilise la syntaxe suivante

1. Pour la matrice A : al.echelonMat('E', A)
2. Pour la matrice augmentée $(A|b)$: al.echelonMat('E', A, b)

Pour obenir **les formes échelonnées réduites** mettez 'ER' au lieu de 'E'

```
[ ]: A=[[2,1,1], [1,-1,1], [1,4,5]]
     b=[[1], [3], [1]]
```

```
[ ]: M=al.echelonMat('ER',A,b)
```

Passez au notebook du chapitre 1.7: Résolutions de système linéarires