

# ALGORITHMIQUE NUMÉRIQUE

Notes de Cours  
Jean-Yves JAFFRAY

Introduction

Analyse et algorithmique numériques

CHAPITRE 1

REPRESENTATION DES NOMBRES EN  
ORDINATEUR ET PRECISION  
DES CALCULS

CHAPITRE 2

RÉSOLUTION D'UN SYSTÈME  
D'ÉQUATIONS LINÉAIRES

# Introduction

## 0.1 Analyse et algorithmique numériques

L'analyse numérique est la discipline qui étudie *d'un point de vue mathématique* la résolution de problèmes tels que la recherche d'une solution d'un système d'équations ou du maximum d'une fonction. Le plus souvent, elle ne fournit qu'une solution "approchée" ; par exemple, elle donne un algorithme permettant d'engendrer les termes successifs d'une suite de nombres réels  $x^{(n)}$  telle que  $\lim_{n \rightarrow \infty} x^{(n)} = \bar{x}$ , où  $f(\bar{x}) = 0$  ; éventuellement, elle est capable de dire à partir de quelle valeur  $n_0$  de  $n$ , on est assuré que  $|f(x^{(n)})| < \epsilon$  ou que  $|x^{(n)} - \bar{x}| < \epsilon$ . Puisque l'on ne peut jamais procéder qu'à un nombre fini d'itérations, on ne peut espérer connaître qu'une valeur approchée de la solution  $\bar{x}$ .

A côté de cette cause mathématique d'approximation, en existe une seconde, qui est, elle, une cause informatique : l'ordinateur est incapable de traiter les nombres réels du mathématicien, qui, étant (par définition) des limites de suites de nombres rationnels, exigeraient généralement chacun une capacité de mémoire illimitée. L'*algorithmique numérique*, s'appuyant sur les résultats des analystes numériques, cherche à les valider dans le cadre du calcul sur ordinateur, malgré cette limitation informatique.

# CHAPITRE 1

## REPRESENTATION DES NOMBRES EN ORDINATEUR ET PRECISION DES CALCULS

### 1

#### 1.1 Représentation des nombres en virgule flottante normalisée

Un ordinateur lorsqu'il travaille avec une précision donnée (simple précision, double précision) alloue une place en mémoire bien définie à chaque nombre.

Pour les calculs scientifiques le mode de représentation le plus courant (mais pas le seul) est la représentation en *virgule flottante normalisée*.

Les nombres acceptés sont de la forme :

$$N = \pm m \times b^e,$$

où :

$b$  est la *base* (par ex. la base 2 ou la base 10)

$m$  est la *mantisse*, nombre de la forme  $0.\alpha_1\alpha_2\dots\alpha_{K-1}\alpha_K$ , avec, pour tout  $k$ ,  $\alpha_k \in \{0, 1, \dots, 9\}$  si la base est 10 ( $\alpha_k \in \{0, 1\}$  si la base est 2, etc..) et  $\alpha_1 \neq 0$  (la *partie décimale de la mantisse ne doit pas commencer par un zéro*)

$e$ , entier, est l'*exposant* ou *caractéristique*.

Dans ce qui suit nous nous placerons dans le cas :  $b = 10$ ;  $K = 7$ ;  $e \in \{-50, -49, \dots, -1, 0, 1, \dots, 48, 49\}$ . On peut alors représenter tous les nombres (à 7 décimales) de  $0.1 \times 10^{-50}$  à  $0.99\dots9 \times 10^{49}$ .

Plutôt que la valeur de  $e$  on préfère noter celle de  $e+50 \in \{00, 01, \dots, 98, 99\}$  d'où la représentation :

$$N \rightsquigarrow \boxed{\pm \quad \beta_1 \quad \beta_2 \quad | \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6 \quad \alpha_7}$$

soit  $N \rightsquigarrow (S_N, C_N, M_N)$  (signe, caractéristique augmentée de 50 et mantisse de  $N$ )

*Exemples.*

$$N = -9.3 = -0.93 \times 10^1 \rightsquigarrow \boxed{- \quad 5 \quad 1 \quad | \quad 9 \quad 3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}$$

$$N = +0.034 = 0.34 \times 10^{-1} \rightsquigarrow \boxed{+ \quad 4 \quad 9 \quad | \quad 3 \quad 4 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0}$$

## 1.2 Opérations arithmétiques et perte de précision

Multiplication :

Etant donné deux nombres  $A \rightsquigarrow (S_A, C_A, M_A)$  et  $B \rightsquigarrow (S_B, C_B, M_B)$  leur produit  $P = A \times B$  est un nombre  $P \rightsquigarrow (S_P, C_P, M_P)$  où :

- $S_P = S_A \times S_B$  ;
- si  $M_A \times M_B \geq 0.1$ ,  $M_P$  est égal à  $M_A \times M_B$  tronqué après son 7<sup>eme</sup> chiffre et  $C_P = C_A + C_B - 50$
- si  $M_A \times M_B < 0.1$ , alors  $M_P$  est égal à  $10 \times M_A \times M_B$  tronqué après son 7<sup>eme</sup> chiffre et  $C_P = C_A + C_B - 51$ .

La troncature entraine une perte d'information : on ne peut plus retrouver  $A$  par division de  $A \times B$  par  $B$ .

Division :

Le quotient  $Q = A \div B$  de deux nombres  $A \rightsquigarrow (S_A, C_A, M_A)$  et  $B \rightsquigarrow (S_B, C_B, M_B)$  est un nombre  $Q \rightsquigarrow (S_Q, C_Q, M_Q)$  où :

- $S_Q = S_A \times S_B$  ;
- si  $M_A \div M_B < 1$ ,  $M_Q$  est égal à  $M_A \div M_B$  tronqué après son 7<sup>eme</sup> chiffre et  $C_P = C_A - C_B + 50$
- si  $M_A \div M_B > 1$ , alors  $M_Q$  est égal à  $M_A \div (10 \times M_B)$  tronqué après son 7<sup>eme</sup> chiffre et  $C_P = C_A - C_B + 51$ .

Ici aussi la troncature est génératrice de perte d'information.

Addition / Soustraction :

La somme  $\Sigma$  de deux nombres  $A$  et  $B$  de signes opposés (ou la différence de deux nombres de même signe) peut engendrer une forte imprécision.

En effet, supposons par exemple  $C_A \geq C_B$  ; dans ce cas  $M_\Sigma = |M_A - M_B \times 10^{C_B - C_A}| \times 10^r$  et  $C_\Sigma = C_A - r$  où  $r$  est tel que  $0.1 \leq M_\Sigma < 1$ . La multiplication par  $10^r$  correspond à l'introduction de  $r$  chiffres arbitraires dans  $M_\Sigma$ .

*exemple :*

$$\begin{array}{l}
 A = +0.8743257 \times 10^9 \rightsquigarrow \boxed{+ \ 5 \ 9 \ | \ 8 \ 7 \ 4 \ 3 \ 2 \ 5 \ 7} \\
 B = -0.8743252 \times 10^9 \rightsquigarrow \boxed{- \ 5 \ 9 \ | \ 8 \ 7 \ 4 \ 3 \ 2 \ 5 \ 2} \\
 \Sigma = +0.5 \times 10^3 \rightsquigarrow \boxed{+ \ 5 \ 3 \ | \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0}
 \end{array}$$

Les 6 derniers de la mantisse de  $\Sigma$  sont non-significatifs : ils ont été choisis arbitrairement par la machine.

### 1.3 Exemples d'erreurs engendrées par l'imprécision numérique

#### Exemple 1

La matrice  $A = \begin{bmatrix} 1 & 1 \\ 1 + 10^{-7} & 1 \end{bmatrix}$  est régulière de déterminant  $\det A = -10^{-7}$ .

La machine ne peut distinguer le nombre  $(1 + 10^{-7}) = 0.10000001 \times 10^1$  du nombre 1, car elle les représente tous deux par :  $\boxed{+ \ 5 \ 1 \ | \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0}$ ;

elle trouvera donc  $\det A = 1 - 1 = 0$  et conclura que  $A$  est singulière.

#### Exemple 2

Le système  $\begin{cases} 3x + 5y = 4 & (1) \\ 3x + 5y = 4 & (2) \end{cases}$ , ayant une équation redondante, a une infinité de solutions. Pourtant si on le résoud par la méthode de Gauss (voir ci-dessous), on doit diviser (1) par 3, d'où (1') d'une part, multiplier (1') par  $-3$  et l'ajouter à (2), d'où (2') d'autre part ; ces calculs donnent :

$$\begin{bmatrix} 3.000000 & 5.000000 & 4.000000 & (1) \\ 3.000000 & 5.000000 & 4.000000 & (2) \end{bmatrix}$$

$$\begin{bmatrix} 1.000000 & 1.666666 & 1.333333 & (1') \\ 0.000000 & 0.000002 & 0.000001 & (2') \end{bmatrix}$$

De (2') on tire alors  $y = 0.5$ , puis, de (1'),  $x = 0.5$ ; on a trouvé une solution mais pas toutes les solutions.



- i) multiplier tous les éléments de la ligne  $A_{i_0}$  par  $\frac{1}{a_{i_0 j_0}}$  ;  
 ii) ajouter à toute autre ligne  $A_i$  ( $i \neq i_0$ ) de  $A$ , élément par élément, la ligne  $A_{i_0}$  multipliée au préalable par  $-\frac{a_{ij_0}}{a_{i_0 j_0}}$ .

La matrice  $A'$  obtenue a pour  $j_0^{eme}$  colonne le vecteur-unité  $e^{j_0}$ .

Pour le calcul des autres colonnes, (i) exige  $(n-1)$  divisions et (ii), qui est répété  $(m-1)$  fois, exige en tout  $(m-1)$  divisions,  $(m-1) \cdot (n-1)$  multiplications et autant de soustractions.

Cette opération correspond au produit :

$$A' = \begin{bmatrix} a'_{11} & a'_{12} & \dots & 0 & \dots & a'_{1n} \\ a'_{21} & a'_{22} & \dots & 0 & \dots & a'_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a'_{i_0 1} & a'_{i_0 2} & \dots & 1 & \dots & a'_{i_0 n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a'_{m1} & a'_{m2} & \dots & 0 & \dots & a'_{mn} \end{bmatrix} = P.A =$$

$$\begin{bmatrix} 1 & 0 & \dots & -\frac{a_{1j_0}}{a_{i_0 j_0}} & \dots & 0 \\ 0 & 1 & \dots & -\frac{a_{2j_0}}{a_{i_0 j_0}} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{a_{i_0 j_0}} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\frac{a_{mj_0}}{a_{i_0 j_0}} & \dots & 1 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j_0} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j_0} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i_0 1} & a_{i_0 2} & \dots & a_{i_0 j_0} & \dots & a_{i_0 n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mj_0} & \dots & a_{mn} \end{bmatrix}$$

Noter que  $P$  est régulière, car  $\det(P) = \frac{1}{a_{i_0 j_0}}$ .

Dans le cas où le pivot est  $a_{11}$ ,  $P$  est de plus triangulaire inférieure; par

exemple, pour  $A_{(3,4)}$  :

$$A' = \begin{bmatrix} 1 & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \square & \square & \square \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{bmatrix} \times A.$$

**Exemple 1**  $A = \begin{bmatrix} 3 & 2 & -5 \\ 1 & 3 & 2 \\ 6 & 7 & 4 \end{bmatrix}$ . Pour le pivot  $a_{11} = 3$ ,

$$A \rightarrow A' = P.A = \begin{bmatrix} 1 & \frac{2}{3} & -\frac{5}{3} \\ 0 & \frac{7}{3} & \frac{11}{3} \\ 0 & 3 & 14 \end{bmatrix} \quad \text{avec } P = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}.$$

### 3.1 PROCÉDURE DE PIVOTAGE

NB : Les algorithmes sont écrits dans un pseudo-langage rudimentaire : les types ne sont pas déclarés (le contexte les indique); les éléments d'une matrice  $A$  sont désignés par  $a_{ij}$  (et non  $A[i][j]$ ); etc..

```
Fonction pivotage ( $A$  : matrice  $(m, n)$ ;  $i_0, j_0$  : entier) : matrice  $(m, n)$ 
// pivot  $a_{i_0 j_0} \neq 0$ .
// pivotage  $(A, i_0, j_0) = P.A$ , avec  $P$  régulière et  $(P.A)^{j_0} = e^{i_0}$ 
début
  pour  $i = 1 \rightarrow m$  et  $j = 1 \rightarrow n$  faire
    si  $i = i_0$  alors si  $j \neq j_0$  alors  $a'_{i_0 j} \leftarrow \frac{a_{i_0 j}}{a_{i_0 j_0}}$ ;
    sinon  $a'_{i_0 j_0} \leftarrow 1$ ;
    sinon si  $j \neq j_0$  alors  $a'_{ij} \leftarrow a_{ij} - \frac{a_{i j_0}}{a_{i_0 j_0}} a_{i_0 j}$ ;
    sinon  $a'_{i j_0} \leftarrow 0$ ;

  pivotage  $\leftarrow A'$ 
fin
```

## 4 MÉTHODE DE GAUSS

La méthode de résolution des systèmes d'équations linéaires par éliminations successives des variables, ou *méthode d'élimination*, de Gauss comprend une phase de transformation du système et une phase de résolution.

### 4.1 Pas de la phase de transformation

Soit, pour  $k < m$ , une matrice  $M^{(k)}$  de la forme :

$$M^{(k)} = \begin{bmatrix} T^{(k)} & C \\ 0 & D \end{bmatrix}$$

où  $T^{(k)}$  est triangulaire supérieure avec tous ses éléments diagonaux égaux  $\begin{matrix} (k,k) \\ (k,k) \end{matrix}$

à 1 et  $D$  est non-nulle.

Par permutation de ses  $(m - k)$  dernières lignes et  $(n - k)$  dernières colonnes, on transforme  $M^{(k)}$  en une matrice

$$M^{(k)} \equiv_{perm} \begin{bmatrix} T^{(k)} & C^* \\ 0 & \Delta \end{bmatrix}$$

où  $\Delta$  a son élément  $\delta_{11} \neq 0$ .

Plus précisément,  $\Pi$  et  $\Pi'$  étant les matrices de permutation adéquates :

$$\begin{bmatrix} T^{(k)} & C^* \\ 0 & \Delta \end{bmatrix} = \begin{bmatrix} E & 0 \\ 0 & \Pi \end{bmatrix} \times \begin{bmatrix} T^{(k)} & C \\ 0 & D \end{bmatrix} \times \begin{bmatrix} E & 0 \\ 0 & \Pi' \end{bmatrix} = \begin{bmatrix} T^{(k)} & C.\Pi' \\ 0 & \Pi.D.\Pi' \end{bmatrix},$$

d'où

$$C^* = C.\Pi' \text{ et } \Delta = \Pi.D.\Pi'.$$

On opère ensuite un pivotage de  $\Delta$  avec  $\delta_{11}$  pour pivot, qui transforme  $\Delta$  en  $\Delta' = P.\Delta$ , matrice ayant pour première colonne le vecteur-unité  $e^1$ ;  $M^{(k)}$  est donc finalement transformée en une matrice

$$M^{(k+1)} = \begin{bmatrix} E & 0 \\ 0 & P \end{bmatrix} \times \begin{bmatrix} T^{(k)} & C^* \\ 0 & \Delta \end{bmatrix} = \begin{bmatrix} T^{(k)} & C^* \\ 0 & P.\Delta \end{bmatrix} \text{ avec } P.\Delta = \begin{bmatrix} 1 & c' \\ 0 & D' \end{bmatrix}$$

$M^{(k+1)}$  est donc bien de la forme :

$$M^{(k+1)} = \begin{bmatrix} T^{(k)} & c^* & \Gamma^* \\ 0 & 0 & 1 & c' \\ 0 & 0 & 0 & D' \end{bmatrix} = \begin{bmatrix} T^{(k+1)} & C' \\ 0 & D' \end{bmatrix}.$$

Notons que  $M^{(k+1)} = N^\circ.M^{(k)}. \Pi^\circ$ , avec  $N^\circ = \begin{bmatrix} E & 0 \\ 0 & P \end{bmatrix} \times \begin{bmatrix} E & 0 \\ 0 & \Pi \end{bmatrix}$  et  $\Pi^\circ = \begin{bmatrix} E & 0 \\ 0 & \Pi \end{bmatrix}$  qui sont toutes deux régulières.

## 4.2 Phase de transformation

Partant de la matrice  $M^{(0)} = A$ , on calcule successivement  $M^{(1)}, M^{(2)}, \dots$ , en utilisant le pas décrit ci-dessus. L'arrêt intervient dès qu'il ne peut plus être appliqué, c.-à-d. au plus tard après  $\min\{m, n\}$  itérations ( $D = \emptyset$ ) mais éventuellement plus tôt ( $D = \emptyset$ ).

On obtient donc  $M^{(k_0)} = \begin{bmatrix} T & C \\ 0 & 0 \end{bmatrix}$ , où  $T_{(k_0, k_0)}$  est triangulaire supérieure, avec tous ses éléments diagonaux égaux à 1 et  $k_0 \leq m$ .

En fait, en prévision de la phase de résolution, on applique les transformations successives à la matrice  $\begin{bmatrix} A & b \\ (m, n+1) \end{bmatrix}$ . On obtient donc finalement une

matrice  $\begin{bmatrix} M^{(k_0)} & b^{(k_0)} \end{bmatrix}$  avec  $M^{(k_0)} = \begin{bmatrix} T & C \\ 0 & 0 \end{bmatrix} = N.A.\Pi$ , et  $b^{(k_0)} = N.b$ , où  $N$  est une matrice régulière (comme produit de matrices de permutations et de matrices de pivotage) et  $\Pi$  est une matrice de permutation (comme produit de matrices de permutations).

### 4.2.1 Choix des pivots

Il y a une certaine latitude de choix du pivot : de quel élément  $m_{i_0 j_0}$  de la sous-matrice  $D$  de  $M^{(k)}$  doit-on faire le  $\delta_{11}$  de  $\Delta$  ?

Dans chaque pivotage, un élément  $m_{ij}$ ,  $i \neq i_0$ , est transformé en

$$m'_{ij} = m_{ij} - \frac{m_{i_0 j_0}}{m_{i_0 j_0}} m_{i_0 j}.$$

Si le terme soustrait de  $m_{ij}$  est d'un ordre de grandeur supérieur à celui de  $m_{ij}$ , on va perdre de la précision (par exemple,  $m_{ij}$  ne sera plus récupérable exactement à partir de  $m'_{ij}$ ). Il est donc intéressant d'avoir, en valeur absolue,  $\frac{m_{i_0 j_0}}{m_{i_0 j_0}}$  petit, c.-à-d.  $\frac{m_{i_0 j_0}}{m_{i_0 j}}$  grand, pour tout  $j$ , soit encore  $\frac{m_{i_0 j_0}}{\max_j m_{i_0 j}}$ ; d'où un critère de choix du pivot, qui dans la pratique donne de bons résultats :

$$(i_0, j_0) = \underset{i, j}{\operatorname{argmax}} \frac{|m_{ij}|}{\max_j |m_{ij}|}$$

Cependant la méthode est souvent implémentée en cherchant seulement à

éviter un pivot trop petit, donc en prenant  $(i_0, j_0) = \underset{i, j}{\operatorname{argmax}} |m_{ij}|$ .

Si l'on ne veut pas permuter de colonnes, on impose  $j_0 = k_0 + 1$  et ne maximise que sur  $i$ .

### 4.3 Phase de résolution

Le système  $(S) : A.x = b$  est équivalent au système

$$N.A.x = N.A.\Pi.\Pi^{-1}.x = N.b, \text{ soit encore}$$

$$(S') : M^{(k_0)}.\Pi^{-1}.x = b^{(k_0)}$$

qui s'écrit, en utilisant les décompositions

$$M^{(k_0)} = \begin{bmatrix} T & C \\ 0 & 0 \end{bmatrix}, \quad \Pi^{-1}.x = \begin{bmatrix} x' \\ x'' \end{bmatrix} \quad \text{et} \quad b^{(k_0)} = \begin{bmatrix} b' \\ b'' \end{bmatrix},$$

$$(S') \begin{cases} T.x' + C.x'' = b' \\ 0 = b'' \end{cases}$$

on peut conclure que :

i) si  $k_0 < m$  et  $b'' \neq 0$ , ni  $(S')$  ni  $(S)$  n'ont de solution ;

ii) si  $k_0 = m$  ou  $k_0 < m$  et  $b'' = 0$ , on obtient toutes les solutions de  $(S')$  et donc de  $(S)$  en donnant aux composantes de  $x''$ , dites *variables libres*, des valeurs arbitraires,  $\hat{x}''$ , puis en résolvant de proche en proche, en remontant à partir de sa dernière équation, le système triangulaire

$$T.x' = b' - C.\hat{x}'' =_{def} d \quad (\text{donc pour des variables libres nulles : } d = b') :$$

$$\begin{aligned} x'_{k_0} &= d_{k_0}; & x'_{k_0-1} &= d_{k_0-1} - t_{k_0-1, k_0} \cdot x'_{k_0} \\ x'_{k_0-2} &= d_{k_0-2} - t_{k_0-2, k_0-1} \cdot x'_{k_0-1} - t_{k_0-2, k_0} \cdot x'_{k_0}; & \text{etc...} \end{aligned}$$

**Exemple 2** Soit à résoudre  $Ax = b$  avec  $[A \ b] = \begin{bmatrix} 4 & -12 & 20 & 28 \\ 2 & -6 & -2 & 3 \\ 6 & -18 & -6 & 9 \end{bmatrix} =$

$M^{(0)}$ . On calcule  $M^{(1)} = \begin{bmatrix} 1 & -3 & 5 & 7 \\ 0 & 0 & -12 & -11 \\ 0 & 0 & -36 & -33 \end{bmatrix}$  puis, après permutation des

colonnes 2 et 3,  $M^{(2)} = \begin{bmatrix} 1 & 5 & -3 & 7 \\ 0 & 1 & 0 & \frac{11}{12} \\ 0 & 0 & 0 & 0 \end{bmatrix}$ . Le système donné est donc équivalent

au système  $\begin{cases} x_1 + 5x_3 - 3x_2 = 7 \\ x_3 = \frac{11}{12} \end{cases}$  qui a une infinité de solutions :  $x_3 = \frac{11}{12}$ ;  $x_2 = \alpha$  ( $\alpha \in \mathbb{R}$ , quelconque);  $x_1 = \frac{29}{12} + 3\alpha$ .

#### 4.4 ALGORITHME DE GAUSS

N.B. : Lorsqu'une matrice  $M$  a pour ensembles d'indices de lignes et de colonnes  $\bar{I}$  et  $\bar{J}$ , respectivement, on notera  $M_I^J$  la sous-matrice de  $M$  constituée par les éléments  $m_{i,j}$ ,  $i \in I \subset \bar{I}$ ,  $j \in J \subset \bar{J}$ .

**Procédure GAUSS** ( $A$  : matrice  $(m, n)$ ;  $b$  : vecteur  $(m)$ ) : vecteur  $(n)$   
 // GAUSS ( $A, b$ ) est solution du système  $Ax = b$ . Les variables libres éventuelles  $y$  sont nulles.

**début**

$M \leftarrow [A \ b]$ ;  $x \leftarrow 0$ ;  $I \leftarrow \{1, \dots, m\}$ ;  $J \leftarrow \{1, \dots, n\}$ ;

$J' \leftarrow \{1, \dots, n+1\}$ ;  $k \leftarrow 0$ ;

**tant que**  $\max_{(i,j) \in I \times J} |m_{ij}| > 0$  **et**  $I \neq \emptyset$  **et**  $J \neq \emptyset$  **faire**

**début**

$(i^*, j^*) \leftarrow \operatorname{argmax}_{(i,j) \in I \times J} \left\{ \frac{|m_{ij}|}{\max_{j \in J} |m_{ij}|} \mid m_{ij} \neq 0 \right\}$ ;

// recherche du meilleur pivot

$k \leftarrow k + 1$ ;  $\operatorname{col}(k) \leftarrow j^*$ ;

// enregistrement des permutations des variables

$M_I^{J'} \leftarrow \operatorname{pivotage}(M_I^{J'}, i^*, j^*)$ ;

$I \leftarrow I \setminus \{i^*\}$ ;  $J \leftarrow J \setminus \{j^*\}$ ;  $J' \leftarrow J \cup \{n+1\}$ ;

**fin**

//fin des transformations et début de la résolution

**si**  $I = \emptyset$  **ou**  $I \neq \emptyset$  **et**  $[i \in I \implies m_{i,n+1} = 0]$

**alors pour**  $i = 1 \rightarrow k$  **faire**  $x_{\operatorname{col}(k+1-i)} \leftarrow m_{k+1-i, n+1}$ ;

**si**  $i \neq k$  **pour**  $l = 1 \rightarrow k - i$  **faire**

$m_{l, n+1} \leftarrow m_{l, n+1} - m_{l, k+1-i} \times x_{\operatorname{col}(k+1-i)}$ ;

$\text{GAUSS} \leftarrow x$ ;

// les variables libres auront la valeur 0 dans la solution fournie

**sinon sortie\_en\_erreur** "pas de solution"

**fin**

## 4.5 DÉCOMPTE DES OPÉRATIONS

Nous ferons ce décompte dans le cas où  $A$  est carrée  $(n, n)$  et régulière et donc  $[A \ b]$  est de format  $(n, n + 1)$ .

La phase de transformation comporte alors  $n$  pas :  $k = 0, 1, \dots, n - 1$ . D'après ce que nous avons vu, le pas  $k$  comporte  $(n - k - 1) + (n - k) = 2(n - k) - 1$  divisions,  $(n - k) \cdot (n - k - 1)$  multiplications et autant de soustractions. D'où en tout  $2 \sum_{k=0}^{n-1} (n - k) - n = n^2$  divisions,  $\sum_{k=0}^{n-1} (n - k) \cdot (n - k - 1) = \frac{n(n-1)(n+1)}{3}$  multiplications et le même nombre de soustractions. Le nombre total d'opérations (de toutes sortes) de cette phase est donc de l'ordre de  $\frac{2}{3}n^3$ .

La phase de résolution y ajoute, dans la  $l^{\text{eme}}$  équation à partir du bas,  $l$  multiplications et autant de soustractions, donc en tout  $2 \sum_{l=1}^n l = n(n + 1)$  - de l'ordre de  $n^2$  - opérations.

Au total des deux phases, le nombre d'opérations reste de l'ordre de  $\frac{2}{3}n^3$ .

## 5 MÉTHODE DE GAUSS-JORDAN

La méthode de Gauss-Jordan est une variante de celle de Gauss, où les pivotages successifs font apparaître une sous-matrice unité et non plus seulement une matrice triangulaire. La phase de résolution est alors immédiate : une fois fixées les valeurs des éventuelles variables libres, on lit directement les valeurs du reste des variables.

### 5.1 Pas de la phase de transformation

Soit, pour  $k < m$ , une matrice  $M^{(k)}_{(m,n)} = \begin{bmatrix} E_{(k)} & C \\ 0 & D \end{bmatrix}$

où  $E_{(k)}$  est une matrice-unité  $(k, k)$  et  $D$  est non-nulle.

Par permutation de ses  $(m - k)$  dernières lignes et  $(n - k)$  dernières colonnes, on transforme  $M^{(k)}$  en une matrice équivalente

$$M^{(k)} \equiv_{perm} \begin{bmatrix} E_{(k)} & C^* \\ 0 & \Delta \end{bmatrix}$$

où  $\Delta$  a son élément  $\delta_{11} \neq 0$ .

On opère un pivotage de  $M^{(k)}$  (et non seulement, comme dans la méthode de Gauss, de  $\Delta$ ) avec  $\delta_{11}$  pour pivot, qui transforme donc la première colonne de la matrice  $\begin{bmatrix} C^* \\ \Delta \end{bmatrix}$  en un vecteur  $\begin{bmatrix} 0 \\ e^1 \end{bmatrix}$  et donc  $M^{(k)}$  en une matrice

$$M^{(k+1)} = \begin{bmatrix} E^{(k)} & 0 & \Gamma^* \\ 0 & 0 & 1 & c' \\ 0 & 0 & 0 & D' \end{bmatrix} = \begin{bmatrix} E^{(k+1)} & C' \\ 0 & D' \end{bmatrix}.$$

## 5.2 Phase de transformation

Partant de la matrice  $M^{(0)} = A$ , on calcule successivement  $M^{(1)}, M^{(2)}, \dots$ , en utilisant le pas décrit ci-dessus. L'arrêt intervient dès qu'il ne peut plus être appliqué, c.-à-d. au plus tard après  $\min\{m, n\}$  itérations ( $D = \emptyset$ ) mais éventuellement plus tôt ( $D = 0$ ).

On obtient donc  $M^{(k_0)} = \begin{bmatrix} E^{(k_0)} & C \\ 0 & 0 \end{bmatrix} = N.A.\Pi$ , et, simultanément  $b^{(k_0)} = N.b$ , où  $N$  est régulière et  $\Pi$  est une matrice de permutation.

## 5.3 Phase de résolution

Le système  $(S) : A.x = b$  est équivalent au système

$$(S') : M^{(k_0)}. \Pi^{-1}.x = b^{(k_0)}$$

qui s'écrit encore, en utilisant les décompositions

$$\Pi^{-1}.x = \begin{bmatrix} x' \\ x'' \end{bmatrix} \text{ et } M^{(k_0)}.b^{(k_0)} = \begin{bmatrix} b' \\ b'' \end{bmatrix},$$

$$(S') \begin{cases} x' + C.x'' = b' \\ 0 = b'' \end{cases}$$

on peut conclure que :

i) si  $k_0 < m$  et  $b'' \neq 0$ , ni  $(S')$  ni  $(S)$  n'ont de solution ;

ii) si  $k_0 = m$  ou  $k_0 < m$  et  $b'' = 0$ , on obtient toutes les solutions de  $(S')$  et donc de  $(S)$  en donnant aux variables libres des valeurs arbitraires,  $\hat{x}''$ , les variables  $x'$  étant alors déterminées par  $x' = b' - C.\hat{x}''$ .

**Exemple 3** Soit  $A.x = b$  avec  $[ A \quad b ] = \begin{bmatrix} 1 & -1 & 3 & 0 \\ 2 & 4 & -5 & 8 \\ 6 & 12 & -15 & 24 \end{bmatrix} = M^{(0)}$ .

$$M^{(1)} = \begin{bmatrix} 1 & -1 & 3 & 0 \\ 0 & 6 & -11 & 8 \\ 0 & 18 & -33 & 24 \end{bmatrix}; M^{(2)} = \begin{bmatrix} 1 & 0 & \frac{7}{6} & \frac{4}{3} \\ 0 & 1 & -\frac{11}{6} & \frac{4}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad \text{Système}$$

équivalent :  $\begin{cases} x_1 + \frac{7}{6}x_3 = \frac{4}{3} \\ x_2 - \frac{11}{6}x_3 = \frac{4}{3} \end{cases}$  qui a une infinité de solutions :  $x_3 = \alpha (\in \mathbb{R}, \text{quelconque})$ ;  $x_1 = \frac{4}{3} - \frac{7}{6}\alpha$ ;  $x_2 = \frac{4}{3} + \frac{11}{6}\alpha$ .

## 5.4 ALGORITHME DE GAUSS-JORDAN

La procédure de *GAUSS-JORDAN* ne diffère de celle de *GAUSS* que sur les deux points suivants :

- 1) L'instruction d'appel à la procédure *pivotage* est :  
 $M \leftarrow \text{pivotage}(M, i^*, j^*)$
- 2) Les instructions de résolution du système se réduisent à :  
**si**  $I = \emptyset$  **ou**  $J \neq \emptyset$  **et**  $[i \in I \implies m_{i,n+1} = 0]$   
**alors pour**  $i = 1 \longrightarrow k$  **faire**  $x_{\text{col}(k+1-i)} \leftarrow m_{k+1-i, n+1}$ ;  
 $\text{GAUSS-JORDAN} \leftarrow x$ ;  
**sinon sortie\_en\_erreur** "pas de solution"  
**fin**

## 5.5 INVERSION D'UNE MATRICE PAR LA MÉTHODE DE GAUSS-JORDAN

La méthode de Gauss-Jordan demande environ deux fois plus d'opérations que celle de Gauss. Son principal intérêt est de permettre d'obtenir facilement l'inverse d'une matrice régulière :

Soit  $A_{(n,n)}$ , régulière ; une condition suffisante pour qu'une matrice  $N$  soit

l'inverse de  $A$  est que  $N.A = E$ . Or la phase de transformation de l'algorithme de Gauss-Jordan appliqué à un système  $A.x = b$  ( $b$  quelconque) le transforme en un système :

$$N.A.\Pi.\Pi^{-1}x = N.b, \text{ où } N.A = E.$$

(En effet, si l'algorithme s'arrête au bout de  $k < n$  pas, avec  $N.A = \begin{bmatrix} E^{(k)} & 0 \\ 0 & 0 \end{bmatrix}$ , c'est que  $A$  n'est pas régulière, car sinon,  $N$  l'étant toujours,  $N.A$  devrait l'être aussi).

$N$  est un produit de matrices de permutation et de pivotage, mais elle peut être calculée automatiquement par la résolution simultanée des  $n$  systèmes  $A.X^j = e^j$ ,  $j = 1, \dots, n$ , puisque :

$$A.X = A \times [ X^1 \quad \dots \quad X^j \quad \dots \quad X^n ] = [ e^1 \quad \dots \quad e^j \quad \dots \quad e^n ] = E$$

devient

$$N.A.\Pi.\Pi^{-1}X = N.E = N.$$

Il suffit donc d'appliquer la procédure de GAUSS-JORDAN à la matrice

$M = \begin{bmatrix} A & E_{(n)} \end{bmatrix}$ , avec choix du pivot limité aux  $n$  premières colonnes, pour obtenir  $\begin{bmatrix} E_{(n)} & N \end{bmatrix}$  et donc  $A^{-1} = N$ .

## 5.6 ALGORITHME D'INVERSION DE MATRICE PAR LA MÉTHODE DE GAUSS-JORDAN

**Procédure** *inverseGJ*( $A$  : matrice  $(n, n)$ ) : matrice  $(n, n)$   
 // *inverseGJ*( $A$ ) est l'inverse  $N$  de la matrice  $A$   
**début**  
 $M \leftarrow [ A \quad E_{(n)} ]$ ;  $I \leftarrow \{1, \dots, n\}$ ;  $J \leftarrow \{1, \dots, n\}$ ;  $J^0 \leftarrow \{1, \dots, 2n\}$ ;  
**tant que**  $\max_{(i,j) \in I \times J} |m_{ij}| > 0$  **et**  $I \neq \emptyset$  **faire**  
**début**  
 $(i^*, j^*) \leftarrow \operatorname{argmax}_{(i,j) \in I \times J} \left\{ \frac{|m_{ij}|}{\max_{j \in J} |m_{ij}|} \mid m_{ij} \neq 0 \right\}$ ;  
 $M^{J^0} \leftarrow \text{pivotage} (M^{J^0}, i^*, j^*)$ ;  
 $I \leftarrow I \setminus \{i^*\}$ ;  $J \leftarrow J \setminus \{j^*\}$ ;  $J^0 \leftarrow J^0 \setminus \{j^*\}$ ;  
**fin**  
**si**  $I = \emptyset$  **alors pour**  $i = 1 \rightarrow n$  **et**  $j = 1 \rightarrow n$  **faire**  $n_{i,j} \leftarrow m_{i,n+j}$ ;  
 $\text{inverseGJ} \leftarrow N$   
**sinon sortie\_en\_erreur** "  $A$  n'est pas régulière"  
**fin**

## 6 MÉTHODES ITÉRATIVES

Les méthodes *itératives* calculent une suite de vecteurs  $(x^{(k)})$  qui, sous les bonnes conditions, vont tendre vers une solution du système à résoudre.

### 6.1 MÉTHODE DE JACOBI

*Remarque préliminaire.*

Tout système  $(S^\circ) : A^\circ \cdot x = b^\circ$  où tous les éléments diagonaux  $a_{ii}$  de  $A$  sont non-nuls peut être transformé, par pré-multiplication des deux membres par la matrice diagonal  $D$  avec  $d_{ii} = 1/a_{ii}$ , en un système équivalent  $(S) : A \cdot x = b$ , où tous les éléments diagonaux de  $A$  valent 1.

*Méthode de JACOBI pour le cas particulier :  $a_{ii} = 1, \forall i$*

On peut encore écrire  $(S) : x = (E - A) \cdot x + b$ .

Considérons alors la suite de vecteurs  $(x^{(k)})$  définie par :

$$x^{(0)} = b \text{ et } x^{(k+1)} = (E - A) \cdot x^{(k)} + b.$$

Si cette suite tend vers une limite  $\bar{x}$ , les deux membres de la relation de récurrence ci-dessus auront une limite commune, d'où :  $\bar{x} = (E - A) \cdot \bar{x} + b$ . On peut alors calculer une valeur approchée d'une solution de  $(S)$  par itérations.

*Cette limite n'existe pas toujours ; cependant on montre que :*

- une condition *nécessaire et suffisante* d'existence d'une limite est que toutes les valeurs propres de  $(E - A)$  soient en valeur absolue strictement inférieures à 1 ;

- une condition *suffisante* d'existence d'une limite est que

$$\sum_{j \neq i} |a_{ij}| < 1, \text{ pour tout } i.$$

*Méthode de JACOBI pour le cas général où  $a_{ii} \neq 0, \forall i$*

Puisque l'on se ramène au cas particulier en remplaçant  $A$  par  $D^{-1}A$  et  $b$  par  $D^{-1}b$ , la récurrence doit être ;  $x^{(k+1)} = [E - D^{-1}.A] .x^{(k)} + D^{-1}.b$ , soit encore ;

$$x^{(k+1)} = D^{-1} . [(D - A).x^{(k)} + b] .$$

Les conditions de convergence deviennent alors :

- une CNS est est que toutes les valeurs propres de  $(E - D^{-1}A)$  soient en valeur absolue strictement inférieures à 1 ;
- une CS est que :  $\sum_{j \neq i} |a_{ij}| < a_{ii}$ , pour tout  $i$

## 6.2 MÉTHODE DE GAUSS-SEIDEL

C'est une variante de la méthode de Jacobi qui accélère la convergence en modifiant le vecteur courant composante par composante et utilise pour obtenir la composante d'indice  $i$  de  $x^{(k+1)}$  les autres composantes déjà calculées de ce vecteur au lieu de celles de  $x^{(k)}$ .

Dans le cas particulier  $a_{ii} \neq 0, \forall i$ , on calcule donc successivement pour  $i = 1, \dots, n$  :

$$x_i^{(k+1)} = [(E - A).\tilde{x}^{(k)}]_i + b_i$$

avec  $\tilde{x}_j^{(k)} = x_j^{(k+1)}$  (qui est déjà calculé) pour  $j < i$

et  $\tilde{x}_j^{(k)} = x_j^{(k)}$  pour  $j \geq i$ .

La convergence est à peu près deux fois plus rapide qu'avec la méthode de Jacobi.