1 Méthode de Gauss

1.1 La méthode de Gauss sans pivotement

1.2 1. Algorithme MATLAB

L'algorithme se déroule en deux phases : la triangularisation (élimination) et la remontée.

```
function x = gauss_sans_pivot(A, b)
n = length(b);
% Matrice augmentée pour appliquer les opérations sur A et b simultanément
Ab = [A, b];
% — Phase 1 : Élimination (Triangularisation) —
for k = 1:n-1
for i = k+1:n
% Calcul du multiplicateur
m = Ab(i,k) / Ab(k,k);
% Mise à jour de la ligne i
Ab(i, k:n+1) = Ab(i, k:n+1) - m * Ab(k, k:n+1);
end
end
% — Phase 2 : Remontée (Backward Substitution) —
U = Ab(:, 1:n);
y = Ab(:, n+1);
x = zeros(n, 1);
for i = n:-1:1
x(i) = (y(i) - U(i, i+1:n) * x(i+1:n)) / U(i,i);
end
end
```

1.3 2. Les Phrases à retenir

★ **Le concept de "Nettoyage" :** "L'élimination de Gauss est un processus de nettoyage systématique : à chaque étape, on annule l'influence d'une variable dans toutes les équations suivantes pour ne garder qu'une seule inconnue à la fin."

★ **La Fragilité du Pivot :** "Sans pivotement, cette méthode est comme un château de cartes : si une seule valeur diagonale est nulle ou très petite, tout l'édifice numérique s'effondre par division par zéro."

★ **L'Organisation :** "C'est le passage du désordre d'un système plein à l'ordre parfait d'une structure triangulaire où la solution devient évidente."

Exemple 1.

```
% Coefficients du système d'équations linéaires
A = [2, 3, -1; 4, 1, 2; -3, 2, 4];
B = [1; 2; 3];
```

Exemple 2.

```
% Coefficients du système d'équations linéaires
A = [1, 2, 3, 4; 2, 1, 2, 3; 3, 2, 1, 2; 4, 3, 2, 1];
B = [10; 11; 13; 15];
```

1.4 Élimination de Gauss avec pivotement partiel

L'élimination de Gauss avec pivotement partiel est la version "professionnelle" de l'algorithme. Alors que la méthode naïve peut s'effondrer face à un simple zéro, le pivotement partiel sécurise le calcul en recherchant systématiquement l'élément le plus robuste pour servir de diviseur.

1.5 1. Algorithme MATLAB

L'astuce réside dans l'utilisation de la fonction max() pour identifier la ligne ayant la plus grande valeur absolue dans la colonne de travail actuelle.

```
function x = gauss_pivot_partiel(A, b)
n = length(b);
Ab = [A, b]; % Matrice augmentée
for k = 1:n-1
% — Étape de Pivotement —
% On cherche le maximum en valeur absolue dans la colonne k (sous la
diagonale)
[~, relative_idx] = max(abs(Ab(k:n, k)));
ipiv = relative_idx + k - 1; % Index réel dans la matrice
% Échange de la ligne actuelle avec la ligne du pivot max
Ab([k, ipiv], :) = Ab([ipiv, k], :);
% — Étape d'Élimination Classique —
for i = k+1:n
if Ab(k,k) == 0, error('Matrice singulière'); end
m = Ab(i,k) / Ab(k,k);
Ab(i, k:n+1) = Ab(i, k:n+1) - m * Ab(k, k:n+1);
end
end
% Phase de Remontée
x = zeros(n, 1);
for i = n:-1:1
x(i) = (Ab(i, n+1) - Ab(i, i+1:n) * x(i+1:n)) / Ab(i, i);
end
end
```

1.6 2. Les "Phrases" à retenir

★ **L'instinct de survie :** "Le pivotement partiel est le système d'autodéfense de l'analyse numérique : il refuse de diviser par un chiffre faible qui pourrait corrompre tout le résultat."

★ **La quête du leader :** "À chaque étape, on ne se contente pas de la ligne suivante ; on cherche le 'leader' de la colonne pour porter la responsabilité du calcul."

★ **La précision avant tout :** "Ce n'est pas seulement une question de zéro, c'est une question de stabilité : diviser par un grand nombre réduit les bruits de calcul, diviser par un petit nombre les explose."

Exemple 1.

```
% Coefficients du système d'équations linéaires
A = [2, 3, -1; 4, 1, 2; -3, 2, 4];
B = [1; 2; 3];
```

Exemple 2.

```
% Coefficients du système d'équations linéaires
A = [1, 2, 3, 4; 2, 1, 2, 3; 3, 2, 1, 2; 4, 3, 2, 1];
B = [10; 11; 13; 15];
```

2 La méthode de Gauss avec pivotement partiel et mise à l'échelle des lignes

Le pivotement partiel avec mise à l'échelle des lignes (ou Scaled Partial Pivoting) est une version encore plus robuste de l'élimination de Gauss.

Le pivotement partiel classique peut être "trompé" si une ligne contient des coefficients très grands par rapport aux autres. La mise à l'échelle consiste à normaliser chaque ligne par son plus grand élément (en valeur absolue) avant de choisir le pivot, garantissant ainsi que l'on compare ce qui est comparable.

2.1 1. Algorithme MATLAB

Voici un exemple de code MATLAB qui utilise **la méthode de Gauss avec pivotement partiel et mise à l'échelle des lignes** pour résoudre un système d'équations linéaires :

```
function x = gauss_pivot_echelle(A, b)
n = length(b);
Ab = [A, b];
% — Étape 1 : Calcul des facteurs d'échelle (Scaling factors) —
% s(i) contient la plus grande valeur absolue de la ligne i
s = max(abs(A), [], 2);
if any(s == 0), error('La matrice a une ligne nulle.'); end
```

```
for k = 1:n-1
% — Étape 2 : Choix du pivot mis à l'échelle —
% On cherche l'indice i qui maximise |Ab(i, k)|/s(i)
[~, relative_idx] = max(abs(Ab(k:n, k)) ./ s(k:n));
ipiv = relative_idx + k - 1;
% Échange des lignes dans Ab ET dans le vecteur d'échelle s
Ab([k, ipiv], :) = Ab([ipiv, k], :);
s([k, ipiv]) = s([ipiv, k]);
% — Étape 3 : Élimination —
for i = k+1:n
m = Ab(i,k) / Ab(k,k);
Ab(i, k:n+1) = Ab(i, k:n+1) - m * Ab(k, k:n+1);
end
end
% Phase de Remontée
x = zeros(n, 1);
for i = n:-1:1
x(i) = (Ab(i, n+1) - Ab(i, i+1:n) * x(i+1:n)) / Ab(i, i);
end
end
```

2.2 2. Les Phrases à retenir

★ **L'équité numérique :** "La mise à l'échelle est une question de justice : elle empêche les lignes ayant de gros coefficients de dominer injustement le choix du pivot."

★ **La vision relative :** "Dans cet algorithme, on ne regarde pas seulement si un nombre est grand, on regarde s'il est grand par rapport au reste de sa ligne."

★ **Le bouclier anti-arrondi :** "C'est l'ultime protection contre les erreurs d'arrondi avant de passer au pivotement complet, beaucoup plus coûteux."

Exemple 1:

```
A = [3 -0.1 -0.2; 0.1 7 -0.3; 0.3 -0.2 10];
b = [7.85; -19.3; 71.4];
x = gaussPivotScale(A, b);
disp(x);
```

3 La méthode de Gauss avec pivot complet

La méthode d'élimination de Gauss avec pivotement complet est une variante de l'élimination de Gauss avec pivotement partiel. Alors que le pivotement partiel ne cherche que dans la colonne actuelle, le pivotement complet balaie toute la sous-matrice restante pour trouver l'élément le plus grand en valeur absolue.C'est la méthode la plus robuste contre les erreurs d'arrondi, mais elle

est plus complexe à coder car elle nécessite d'échanger non seulement les lignes, mais aussi les colonnes, ce qui change l'ordre des variables x.

3.1 1. Algorithme MATLAB

Pour cet algorithme, nous devons garder une trace de l'ordre des variables (vecteur p) afin de remettre la solution finale dans le bon ordre $(x_1, x_2, ..., x_n)$.

```
function x = gauss_pivot_complet(A, b)
n = length(b);
p = 1:n; % Vecteur pour suivre l'ordre des variables (colonnes)
Ab = [A, b];
for k = 1:n-1
% — Étape 1 : Recherche du pivot max dans toute la sous-matrice —
[val, idx] = max(abs(Ab(k:n, k:n)), [], 'all', 'linear');
[row_relative, col_relative] = ind2sub([n-k+1, n-k+1], idx);
i_pivot = row_relative + k - 1; % Ligne du pivot
j_pivot = col_relative + k - 1; % Colonne du pivot
% — Étape 2 : Échanges —
% Échange des lignes
Ab([k, i_pivot], :) = Ab([i_pivot, k], :);
% Échange des colonnes (uniquement dans la partie matrice A)
Ab(:, [k, j_pivot]) = Ab(:, [j_pivot, k]);
% On mémorise l'échange des colonnes pour le vecteur x
p([k, j_pivot]) = p([j_pivot, k]);
% — Étape 3 : Élimination classique —
for i = k+1:n
m = Ab(i,k) / Ab(k,k);
Ab(i, k:n+1) = Ab(i, k:n+1) - m * Ab(k, k:n+1);
end
end
% — Phase de Remontée —
temp_x = zeros(n, 1);
for i = n:-1:1
temp_x(i) = (Ab(i, n+1) - Ab(i, i+1:n) * temp_x(i+1:n)) / Ab(i, i);
end
% — Réordonner la solution —
x = zeros(n, 1);
x(p) = temp_x;
end
```

3.2 2. Les Phrases à retenir

★ **La sécurité absolue :** Le pivotement complet est la ceinture et les bretelles de l'analyse numérique : il sacrifie un peu de temps pour garantir que l'erreur ne pourra pas se cacher dans un coin de la matrice.

★ **Le prix de la perfection** : Chercher le meilleur pivot partout demande $O(n^3)$ comparaisons ; c'est le luxe de la précision payé par le temps de calcul.

★ **Le grand chamboulement :** Cette méthode est une danse complexe où lignes et colonnes changent de place continuellement pour maintenir l'équilibre du calcul."

Pour illustrer cela avec deux exemples, prenons le système suivant :

Exemple 1:

```
% Définition de la matrice A et du vecteur b
A = [3 2 1; 1 3 2; 1 1 3];
b = [10; 13; 14];
% Résolution du système Ax = b
x = gaussPivotComplet(A, b);
% Affichage de la solution
disp('La solution du système est :');
disp(x);
```

Exemple 2:

```
% Définition de la matrice A et du vecteur b
A = [2 3 -1 4; 4 4 2 -2; -2 3 -1 3; 5 -3 2 2];
b = [20; 28; 8; 10];
% Résolution du système Ax = b
x = gaussPivotComplet(A, b);
% Affichage de la solution
disp('La solution du système est :');
disp(x);
```