Matrices Hiérarchiques et GMRES Inexacte

Étudiant:

GUIMARÃES Eduardo

Professeurs:

FARIA Luiz

MARCHAND Pierre

Enseignant-Référent:

CHAPOUTOT Alexandre

Palaiseau, le 13 mai 2024

Contents

Chapter 1

Méthodes itératives et Sous-espace de Krylov

1.1 Méthodes itératives et motivation

Les méthodes itératives apparaissent comme une alternative aux méthodes de solution directe , où la vrai solution n'est pas recherchée et une bonne approximation suffit.
L'idée consiste à trouver, après un nombre définit d'itérations, une suite x_k qui converge à x, la solution exacte du problème 1.1.

$$x = lim_{k\to\infty} x_k \tag{1.1}$$

La méthode est appliquée de façon à s'arrêter après k itérations, où x_k est le premier élément de la suite à satisfaire la condition 1.2.

$$\frac{||x_k - x||}{||x||} \leq \epsilon \tag{1.2}$$

Où ϵ est une tolérance définie par qui l'applique.
Normalement x n'est pas connue, de façon que 1.2 est changée pour 1.3, où A est la matrice du système linéaire et b le RHS(Right Hand Side).

$$\frac{||Ax_k - b||}{||b||} \leq \epsilon \tag{1.3}$$

Les premiers méthodes itératives utilisaient une décomposition de la matrice A comme une combinaison de deux matrices 1.4, où A_1 est inversible, et chaque itération serait définie comme 1.5.

$$A = A_1 - A_2 \tag{1.4}$$

$$A_1 x_{k+1} = b + A_2 x_k \tag{1.5}$$

Avec une substitution des autres x_k, 1.5 donne 1.6, qui converge n'importe quelle solution initiale ssi $\rho(A_2 A_1^{-1}) < 1$, où $\rho(X)$ est le rayon spectral de la matrice X [2].

$$x_{k+1} = A_1^{-1}(b + A_2 x_k) = A_1^{-1}(b + A_2 A_1^{-1}(b + A_2 x_{k-1}))... = A_1^{-1}\left[\sum_{i=0}^{k}(A_2 A_1^{-1})^i b\right] \tag{1.6}$$

Si $A_1 = I$ et $A_2 = I - A$ en 1.4, la suite trouvée en 1.6 est: $x_1 = b$,$x_2 = 2b - Ab$, $x_3 = 3b - 3Ab + A^2 b$, ...
Même que la condition $\rho(A - I) \leq 1$ soit restrictive [2], cela nous montre qu'une approximation x_k peut être représentée comme 1.7.

$$x_k \in span(b, Ab, A^2 b, ..., A^{k-1} b) \tag{1.7}$$

1.2 Sous-espace de Krylov

Soit $A \in \mathbb{K}^{n \times n}$ une matrice et $b \in \mathbb{K}^n$. Pour $k \leq n$ le sous-espace de Krylov $\mathcal{K}_k = \mathcal{K}_k(A, b)$ associé à A,b est défini comme 1.8.

$$\mathcal{K}_k(A, b) = span(b, Ab, A^2 b, \ldots, A^{k-1} b) \tag{1.8}$$

Ces sous-espaces suivent aussi la propriété: $k < l \rightarrow \mathcal{K}^k \subset \mathcal{K}^l$ [2].
Ce sous-espace $\mathcal{K}_k(A, b)$ est aussi le sous-espace de tous les vecteurs de $\mathbb{R}^m$ qui peuvent être écrits comme $x = p(A)b$, où $p(A)$ est un polynôme de degré inférieur à $k - 1$ dont $p(0) = 1$.
Le problème avec l'utilisation de $A^k b, k \in 0, 1, 2, \ldots$ comme une base vient du fait que les produits successifs de la matrice A font des vecteurs qui sont *presque colinéaires*, vu que ceux sont proches du vecteur propre du plus grand valeur propre de la matrice A.

1.3 Méthode d'Arnoldi

Dans le but d'obtenir une base orthonormale pour $\mathcal{K}_k(A, b)$, le méthode cherche une matrice unitaire Q tel que l'expression 1.9 est valide. $H_k = h_{ij}$ est une matrice de Hessenberg.

$$AQ_k = Q_{k+1} H_k \tag{1.9}$$

Pour chaque vecteur-colonne de Q, q_i, 1.9 peut être écrite comme 1.10, où la représentation de $\mathcal{K}_k(A, b)$ avec une base orthonormal devient plus claire. Dans une application pratique, Q est initialisée avec $q_1 = \frac{b}{||b||}$.

$$Aq_m = h_{1m} q_1 + h_{2m} q_2 + \ldots h_{m+1,m} q_{m+1} \tag{1.10}$$

Un algorithme pour la méthode peut être trouvée en 1.

Algorithm 1 Itération k dArnoldi

$A \in \mathbb{K}^{n\times n}$ et $b \in \mathbb{K}^n$
$x = 0, \beta = \|b\|, q_1 = \frac{b}{\beta}$
for $j = 1, 2, \dots k$ **do**
    $q_{j+1} = Aq_j$
    for $i = 1, 2, \dots j$ **do**
        $h_{ij} = q_{j+1}^t q_i$
        $q_{j+1} = q_{j+1} - h_{ij} q_i$
    end for
    $h_{j+1,j} = \|q_{j+1}\|$
    $q_{j+1} = \frac{q_{j+1}}{h_{j+1,j}}$
end for

Chapter 2

GMRES

Un méthode de projetion en $\mathcal{K}_k(A, b)$, où les différentes approximations sont prises comme en 2.1, où Q_m est le vecteur défini en 1.9.

$$x = x_0 + Q_m y \tag{2.1}$$

Avec 2.1 et 1.9 le résidu devient 2.2, où $x_0 = 0$, $\beta = \|b\|$ et $Q_{m+1}^t b = (\|b\| \, 0 \, 0 \ldots)^t$ puisque les colonnes de Q_{m+1} sont des vecteurs orthonormals et $q_1 = \frac{b}{\|b\|}$.

$$\begin{aligned} r(y) &= \|b - Ax\| \\ &= \|b - A(Q_m y)\| \\ &= \|b - Q_{m+1} H_m y\| \\ &= \left\|Q_{m+1}(Q_{m+1}^t b - H_m y)\right\| \\ &= \|\beta e_1 - H_m y\| \end{aligned} \tag{2.2}$$

Ainsi, y qui apparaît en 2.1, est trouvé comme la solution du problème de minimisation du résidu en 2.2.

$$y = min_y \, \|\beta e_1 - H_m y\| \tag{2.3}$$

Une version initiale du GMRES est en 2. Les lignes entre 4 et 12 apportent la Méthode d'Arnoldi présentée en 1.
Cependant, 2 n'apporte pas une façon efficace de trouver le résidu en chaque itération. Pour le résoudre et trouver aussi une mieux façon de traiter le problème des moindres carrés en 2.3, une transformation est appliquée en H_m, la transformant dans une matrice triangulaire.

2.1 Transformation de Givens

L'opérateur de Givens, $G(i, i+1)$, est une matrice unitaire telle que le vecteur colonne résultant $a = Gb$ a les éléments $a(i) = r \in \mathbb{R}$ et $a(i+1) = 0$. Il est une matrice de struture comme en 2.4. Les coefficients c_i, s_i n'apparaissent que dans les lignes i et $i+1$.

Algorithm 2 GMRES Initial

1: $A \in \mathbb{K}^{n\times n}$ et $b \in \mathbb{K}^n$
2: $x = 0, \beta = \|b\|, q_1 = \frac{b}{\beta}$
3: **for** $k = 1, 2, \ldots$ **do**
4: **for** $j = 1, 2, \ldots k$ **do**
5: $q_{j+1} = Aq_j$
6: **for** $i = 1, 2, \ldots j$ **do**
7: $h_{ij} = q_{j+1}^t q_i$
8: $q_{j+1} = q_{j+1} - h_{ij} q_i$
9: **end for**
10: $h_{j+1,j} = \|q_{j+1}\|$
11: $q_{j+1} = \frac{q_{j+1}}{h_{j+1,j}}$
12: **end for**
13: Trouver $y = min_y \|\beta e_1 - H_m y\|$
14: $x = Q_k y$
15: **Arrêter** si le résidu est inférieur à la tolérance
16: **end for**

$$G(i, i+1) = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_i & s_i & & & \\ & & & -s_i & c_i & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix} \tag{2.4}$$

L'opérateur est une façon de transformer les colonnes de H_m, annulant l'élément dehors la diagonale. Comme un produit d'opérateurs unitaires est encore unitaire, cela nous permet récrire 2.3 comme 2.5, où R_m et g_m sont les résultats de l'application des opérateurs de Givens à H_m et βe_1.

$$y = min_y \|\beta e_1 - H_m y\| = min_y \|g_m - R_m y\| \tag{2.5}$$

Il peut être montré que g_m contient aussi la valeur du résidu de chaque itération [4].
Ainsi, le nouveau problème 2.5 peut être résolu avec une simple substitution.
(écrire le nouvel algorithme)

2.2 GMRES Inexact

La plus grande partie des calculs lourds est dans le produit matrice-vecteur contenue en 1, dans la ligne 4. Ainsi, un des approches pour accélerer les itérations est ne pas réalizer Aq, mais approximer son résultat avec 2.6.

$$\mathcal{A}q = (A + E)q \tag{2.6}$$

Oú E en 2.6 est une *matrice erreur* qui change à chaque itération et sera écrite comme E_k pour

l'itération k.
Quand le produit matrice-vecteur n'est pas exact, la gauche de 1.9 doit être changée par 2.7.

$$\begin{aligned}[(A+E_1)q_1,(A+E_2)q_2,\ldots,(A+E_k)q_k] &= Q_{k+1}H_k \\ (A+\mathcal{E}_k)Q_k &= Q_{k+1}H_k, \quad \mathcal{E}_k = \sum_{i=1}^{k} E_i q_i q_i^t \\ \mathcal{A}Q_k &= Q_{k+1}H_k\end{aligned} \tag{2.7}$$

Maintenant le sous-espace généré par les vecteurs de Q_k n'est plus un sous-espace de Krylov, mais ils sont encore orthonormals. La relation 2.7 montre aussi que Q_k devient une base pour un sous-espace de Krylov, $\mathcal{K}_k(A+\mathcal{E}_k, b)$, spanné par une grande perturbation de A, qui est mise à jour dans chaque itération.
En utilisant r_k, le résidu exact, et $\tilde{r}_k$, le résidu de la k-ème itération du GMRES inexact, il peut être montré [5] que si E_k suit la relation 2.8, alors les deux résidus suivent 2.9.

$$\|E_k\| \leq \frac{\sigma_k\left(\hat{H}_k\right)}{k}\frac{1}{\|\tilde{r}_k\|}\epsilon \tag{2.8}$$

$$\|r_k - \tilde{r}_k\| \leq \epsilon \tag{2.9}$$

En 2.8, $\hat{H}_k$ est la matrice triangulaire obtenue aprés l'application des rotations de Givens en H_k et σ_k le k-ème valeur singulière d'une matrice quelconte.

Chapter 3

Matrices Hiérarchiques et méthode ACA

3.1 Matrices low-rank

En pratique, les matrices sont de grande taille, de façon que stocker chaque élément n'est pas efficace ou même faisable. Si $A \in \mathbb{C}^{n\times m}$ a un rang k tel que $k \leq m$ et que $k(n+m) < n*m$ (A est low-rank), A peut être représentée comme un produit entre deux matrices $U \in \mathbb{C}^{n\times k}$ et $V \in \mathbb{C}^{m\times k}$, ce qui est vu en 3.1, où u_i, v_i sont les vecteurs colonnes de U et V.

$$A = UV^H = \sum_{i=1}^{k} u_i v_i^* \tag{3.1}$$

Ainsi, en stockant $k(n+m)$ éléments pour représenter A, au lieu de $n \times m$. Une matrice A qui peut être représentée comme 3.1 est dite un élément de $\mathbb{C}_k^{n\times m}$

La représentation en 3.1 facilite aussi des autres opérations possibles avec A, comme les produits matrice-vecteur Ab qui sont toujours présents dans les méthodes itératives comme GMRES [1] et les différentes normes, comme $\|A\|_F$, $\|A\|_2$ [1].

Cependant, même matrices de 'full rank', c-à-d matrices peuvent être approximées par matrices de rang plus petit. Un théorème [1] établit que la plus proche matrice en $\mathbb{C}_k^{n\times m}$ d'une matrice en $\mathbb{C}^{n\times m}$ peut être obtenue de la SVD $A = U\Sigma V^H$, où Σ contient les valeurs singulières $\sigma_1 \geq \sigma_2 \ldots \sigma_m \geq 0$ et U, V sont unitaires.

Si A_k est l'approximation obtenue en prenant les k premiers éléments de Σ (en créant la matrice Σ_k), l'erreur entre A et A_k est obtenu en 3.2.

$$\|A - A_k\| = \left\|U\Sigma V^H - U^{'}\Sigma_k V_{'H}\right\| = \|\Sigma - \Sigma_k\| \tag{3.2}$$

Si la norme spectrale, $\|.\|_2$ est utilisée, l'erreur en 3.2 est donné par σ_{k+1}. Pour la norme de Frobenius, $\|.\|_F$, l'erreur devient $\sum_{l=k+1}^{n} \sigma_l^2$.

Au lieu d'approximer des grandes matrices en une seule fois, c'est mieux penser dans les approximations faites pour leurs blocs. Des blocs qui vient d'une discrétisation d'opérateurs elliptiques ont aussi la possibilité d'être aproximés par matrices qui décroissent exponentiellement S_k, comme

en 3.3.

$$\|A - S_k\|_2 < q^k \|A\|_2 \tag{3.3}$$

Ainsi, le rang dépend de la précision d'une façon logarithme, et le rang nécessaire pour une certaine ϵ est 3.4.

$$k(\epsilon) = min\{k \in \mathbb{N} : \sigma_{k+1} < \epsilon\sigma_1\} \tag{3.4}$$

3.2 Méthode ACA(Adaptative Cross Approximation)

Comment montré dans la section antérieure, la méthode SVD donne une approximation de A à partir d'une erreur ϵ, à travers de la relation en 3.2. Néanmoins, c'est une méthode lourde, où la complexité la rend infaisable pour les grands calculs qui peuvent apparaître normalement. L'ACA arrive comme une alternative plus efficace pour les problemes où le noyau est asymptotiquement lisse pour au moins une variable. Il faut mentionner que le noyau lui même n'est pas nécessaire, juste l'information qu'il appartient à ce groupe de fonctions.
L'algorithme pour la méthode est en 3, où a_{ij} sont les éléments d'une matrice $A \in \mathbb{R}^{n\times m}$. L'objectif est d'approximer la matrice A pour $A = S_k + R_k$, $S_k = \sum_{l=1}^{k} u_l v_l^t$ et R_k est le résidu.

Algorithm 3 Méthode ACA

$k = 1$ et $\mathbf{Z} = \emptyset$
repeat
  Trouver i_k
  $\hat{v}_k = a_{i_k,1:m}$
  for $l = 1, \ldots, k-1$ **do**
    $\hat{v}_k = \hat{v}_k - (u_l)_{i_k} v_l$
  end for
  $Z = Z \bigcup \{i_k\}$
  if $\hat{v}_k$ ne disparaît pas **then**
    $j_k = argmax_j |(\hat{v}_k)_j|$; $v_k = (\hat{v}_k)_{j_k}^{-1} \hat{v}_k$
    $u_k = a_{1:n,j_k}$
    for $l = 1, \ldots, k-1$ **do**
      $u_k = u_k - (v_l)_{j_k} u_l$
    end for
    $k = k + 1$
  end if
until $\|u_k\| \|v_k\| \leq \epsilon$

Soit $I, J \in \mathbb{N}$ l'ensemble des index d'une matrice quelconque et $\mathbf{T}_{I\times J}$ l'arbre que contient une partition admissible P de $I \times J$ dans ses feuilles, $\mathfrak{L}(\mathbf{T}_{I\times J})$. L'ensemble des matrices hiérarchiques en $\mathbf{T}_{I\times J}$ rang k pour chaque bloc A_b est définit en 3.5.

$$\mathfrak{H}(\mathbf{T}_{I\times J}, k) = \left\{A \in \mathbb{C}^{I\times J} : rank A_b \leq k, \forall b \in P\right\} \tag{3.5}$$

Bibliography

[1] Mario Bebendorf. *Hierarchical matrices*. Springer, 2008.

[2] Marc Bonnet. *Lecture notes on numerical linear algebra*. ENSTA Paris - POEMS Group, 2021.

[3] Eric Darve and Mary Wootters. *Numerical linear algebra with Julia*. SIAM, 2021.

[4] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[5] Valeria Simoncini and Daniel B Szyld. Theory of inexact krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25(2):454–477, 2003.

[6] Lloyd N Trefethen, David Bau III, and Ricardo D Fierro. Numerical linear algebra. *SIAM Review*, 40(3):735–738, 1998.