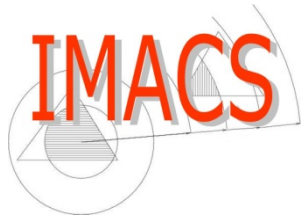


Journée Utilisateurs OpenTURNS #8

H-matrices Principes et applications

Toufic Abboud



$$K[u](x) = \int_{\Gamma} k(x,y)u(y)dy$$

MOTIVATIONS ET UN PEU D'HISTOIRE

Représentation intégrale

- Représentation intégrale des solutions (normalisées) des équations de Maxwell :

$$\begin{cases} \overrightarrow{\text{rot}} \vec{E} - ik\vec{H} = \vec{0} \\ \overrightarrow{\text{rot}} \vec{H} + ik\vec{E} = \vec{0} \end{cases} \quad \text{dans } \Omega$$

(vérifiant la condition de radiation à l'infini)

à l'aide des potentiels électriques et magnétiques :

$$\begin{cases} \vec{E} = -\nabla\phi_E + ik\vec{A}_E - \overrightarrow{\text{rot}} \vec{A}_M \\ \vec{H} = -\nabla\phi_M + ik\vec{A}_M + \overrightarrow{\text{rot}} \vec{A}_E \end{cases}$$

Potentiels

$$\phi_E(x) = \frac{1}{ik} \mathcal{S}(\text{div}_\Gamma \vec{j})(x) = \frac{1}{ik} \int_\Gamma G(x, y) \text{div}_\Gamma \vec{j}(y) d\Gamma(x)$$

$$\phi_M(x) = \frac{1}{ik} \mathcal{S}(\text{div}_\Gamma \vec{m})(x) = \frac{1}{ik} \int_\Gamma G(x, y) \text{div}_\Gamma \vec{m}(y) d\Gamma(x)$$

$$\vec{A}_E(x) = \mathbf{S} \vec{j}(x) = \int_\Gamma G(x, y) \vec{j}(y) d\Gamma(x)$$

$$\vec{A}_M(x) = \mathbf{S} \vec{m}(x) = \int_\Gamma G(x, y) \vec{m}(y) d\Gamma(x)$$

$$G(x, y) = \frac{e^{ik|x-y|}}{4\pi|x-y|}$$

le noyau de Green de l'équation de Helmholtz vérifiant la condition de radiation à l'infini

$$\vec{j} = \vec{H} \wedge \vec{n} \text{ et } \vec{m} = \vec{n} \wedge \vec{E} \quad \text{les courants électrique et magnétique resp.}$$

Équations intégrales - BEM



- On remplace la résolution d'un système d'EDP par celle d'un système d'équations intégrales

- Exemple simple (problème de Dirichlet scalaire) :

$$\int_{\Gamma} G(x, y) \lambda(y) d\Gamma(y) = u_0(x) \quad \text{pour } x \in \Gamma$$

- Approximation par éléments finis (BEM)

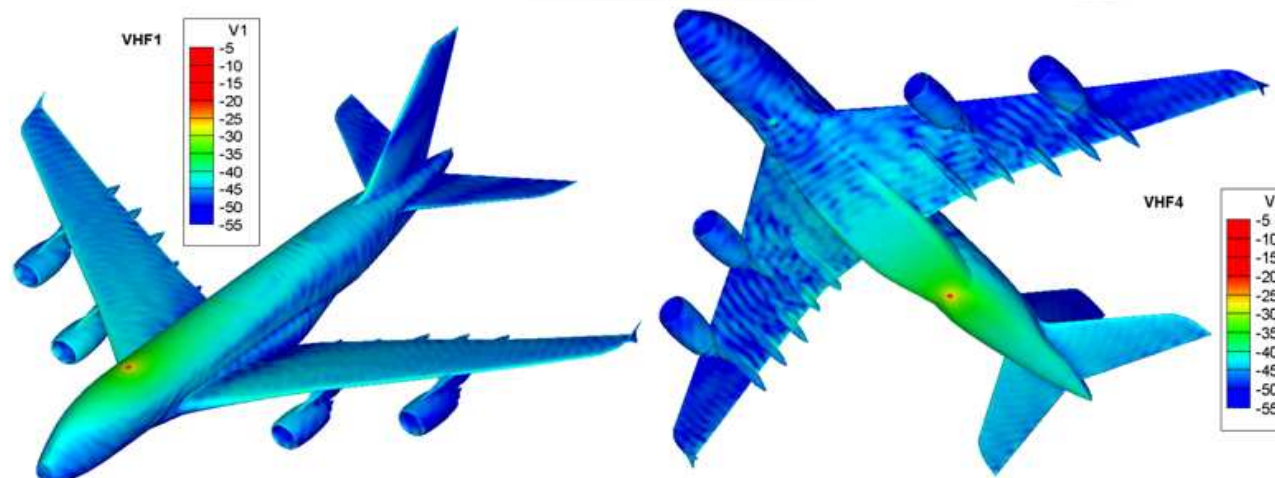
$$S\Lambda = U_0 \quad S \in \mathbb{C}^{N \times N}, \quad \Lambda, U_0 \in \mathbb{C}^N$$

- On gagne une dimension d'espace, mais
- S est une matrice pleine
- Le maillage doit résoudre la longueur d'onde

$$\lambda = 2\pi/k, \quad h \lesssim \min(\lambda/5, h_{geom}) \quad N = O(k^2) \quad \text{en HF}$$

Applications (1)

- Implantation d'antenne sur Airbus A350XWB :
 - GPS (1.5GHz) : 5 M DL
 - Radioaltimètre (4.2-4.4 GHz) : 100 M DL
- Simulation wifi (2-6 GHz) dans un fuselage A350 : 10 à 100 M DL
- SER d'avion militaire en bande X : 100 M DL



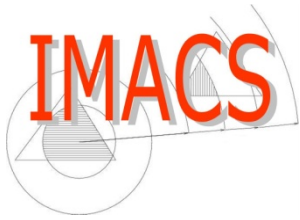
Calcs ASERIS-BE™

Application (2)

- Approche BEM dans d'autres physiques :
acoustique, élastodynamique, électrostatique,
thermique...
- Grands systèmes linéaires
- Matrices pleines, termes génériques du type :

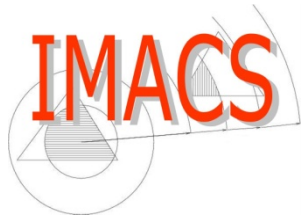
$$\int_{T_i \times T_j} G(x, y) w_j(y) w_i(x) dT_j(y) dT_i(x)$$

- Besoin de méthodes de résolution rapides



Remarque

- Transformée de Fourier discrète
- Matrice pleine
- Algorithme FFT permet :
 - d'effectuer un produit matvec en $O(N \log N)$ opérations
 - de résoudre un système linéaire en $O(N \log N)$ opérations
 - Pas de stockage en $O(N^2)$...



$$K[u](x) = \int_{\Gamma} k(x,y)u(y)dy$$

Rokhlin-Greengard...

MÉTHODE MULTIPOLAIRE RAPIDE – FMM

Problème à N corps

- Dynamique de N particules soumises à des interactions gravitationnelles ou électromagnétiques
- À chaque pas de temps, on doit évaluer le potentiel gravitationnel en chaque particule :

$$V(x_i) = \sum_{j \neq i} G_{i,j} m_j$$

$$G_{i,j} = G(x_i, x_j) = \frac{1}{4\pi |x_i - x_j|}$$

- N^2 interactions (produit matrice-vecteur)



Cas d'un noyau de rang faible

- Considérons le cas d'un noyau qui s'écrit : $K(x, y) = \sum_{l=1}^L \phi_l(x) \psi_l(y)$
- Le potentiel en x du nuage de points (x_j) se calcule rapidement grâce aux moments M_l :

$$V(x) = \sum_{l=1}^L M_l \phi_l(x) \quad \text{avec} \quad M_l = \sum_{j=1}^N m_j \psi_l(x_j)$$

- Calcul matriciel :

$$V = K m \quad \text{avec} \quad K = \Phi \Psi^t \quad \Phi, \Psi \in \mathbb{C}^{N \times L}$$

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_L(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_N) & \cdots & \phi_L(x_N) \end{bmatrix} \quad \Psi = \begin{bmatrix} \psi_1(x_1) & \cdots & \psi_L(x_1) \\ \vdots & \ddots & \vdots \\ \psi_1(x_N) & \cdots & \psi_L(x_N) \end{bmatrix}$$

Coût assemblage / stockage / produit matrice-vecteur : $O(NL)$

C'est intéressant si $L \ll N$



Développement multipolaire

- Si $|y| < |x|$:

$$G(x, y) = \sum_{l,m} \frac{1}{2l+1} \frac{|y|^l}{|x|^{l+1}} Y_{l,m}(\theta_x, \varphi_x) Y_{l,m}^*(\theta_y, \varphi_y)$$

- Ainsi le potentiel d'un sous-ensemble de n points x_j de masse m_j dans une boule $y_j = (\rho_j, \theta_j, \varphi_j)$, $\rho_j < a$

$$V(x) = \sum_{l,m} \frac{M_{l,m}}{2l+1} \frac{1}{r^{l+1}} Y_{l,m}(\theta, \varphi) \quad x = (r, \theta, \varphi), \quad r > a$$

avec

$$M_{l,m} = \sum_{j=1}^n \frac{\rho_j^l}{2l+1} Y_{l,m}^*(\theta_j, \varphi_j)$$

Approximation de rang faible

- La série converge exponentiellement bien en champ lointain :

$$\left| V(x) - \sum_{l=0}^L \sum_{m=-l}^l \frac{M_{l,m}}{2L+1} \frac{1}{r^{l+1}} Y_{l,m}(\theta, \varphi) \right| \leq \frac{M}{r-a} \left(\frac{a}{r} \right)^{L+1}$$

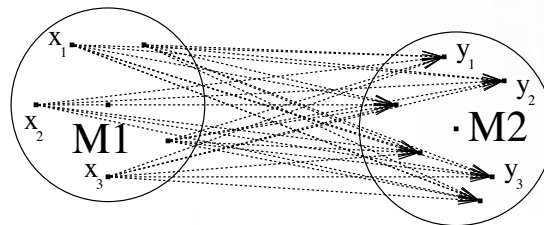
où

$$M = \sum_{j=1}^n |m_j|$$

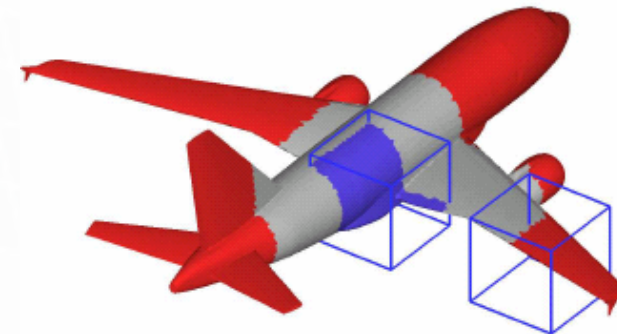
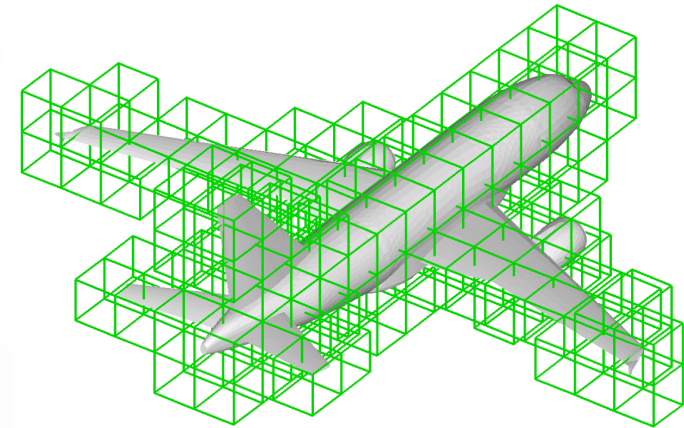
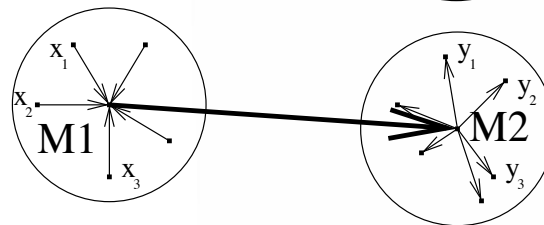
Méthode mono-niveau

- Interaction entre deux 2 boîtes B_i et B_j :
 - Bloc plein si B_i et B_j sont identiques ou voisines
 - Bloc de rang faible
- Complexité en $O(N^{3/2})$

Interactions
classique

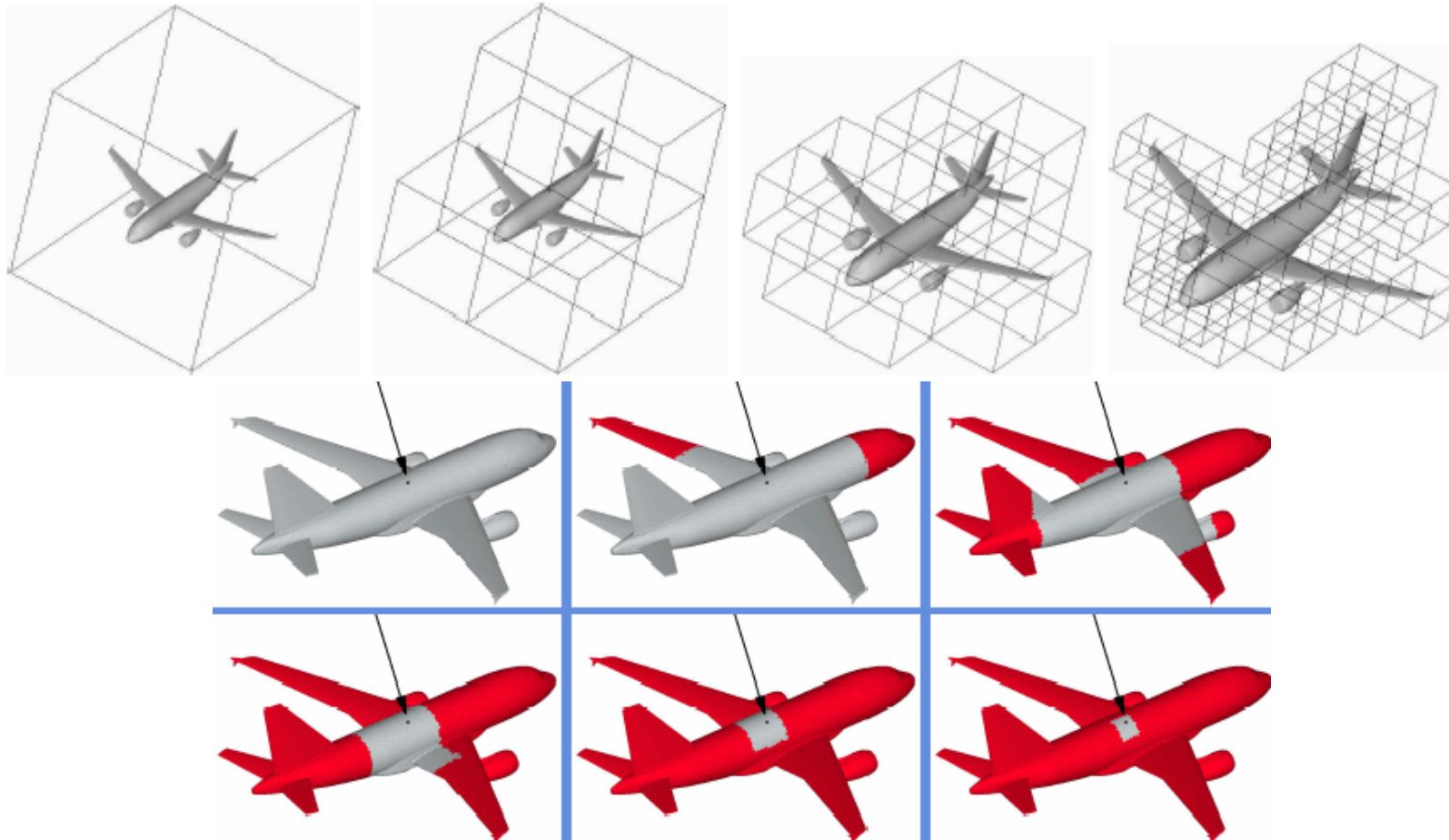


Interactions
FMM



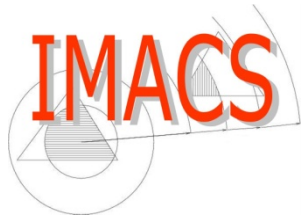
Thèse G. Sylvand (2002)

Méthode multi-niveau



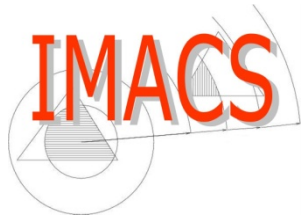
Thèse G. Sylvand (2002)

Assemblage & produit matvec en $O(N \log N)$



Suite...

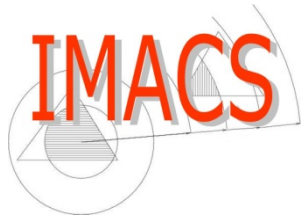
- Idées développées fortement pendant les années 1990-2000, à la suite des travaux de Rokhlin-Greengard :
 - Extension à d'autres noyaux, principalement aux noyaux oscillants
 - Parallélisation
 - Application à des problèmes industriels
- On accélère le produit matrice-vecteur → résolution par méthodes itératives. Problèmes avec :
 - Mauvais conditionnement
 - Difficultés avec les développements stables toutes fréquences
 - Le nombre élevé de seconds membres
 - De plus, c'est une méthode analytique : développement explicite à trouver pour tout nouveau noyau



$$K[u](x) = \int_{\Gamma} k(x,y)u(y)dy$$

W. Hackbusch + B. Khoromskij, S. Sauter, M. Bebendorf...

H-MATRICES



En deux mots

- Cadre systématique et rigoureux pour travailler avec les matrices compressibles
- Algorithmes rapides pour
 - produit matrice-vecteur et matrice-matrice
 - inverse ou factorisation LU ou LDL^t (direct vs. Itératif)
- Découpe hiérarchique de la matrice : bisection vs. octree
- Compression des blocs admissibles : approche algébrique vs. analytique
- Assemblage en $O(N \log N)$, **factorisation en $O(N \log^2 N)$**
- Application aux matrices creuses



Compression d'un bloc

- Matrice $m \times n$ A de rang r

$$A = UV^* \quad U \in \mathbb{C}^{m \times r}, V \in \mathbb{C}^{n \times r}$$

- Compression d'un bloc A ($m \times n$) : c'est trouver U ($m \times r$) et V ($n \times r$) avec $r \ll (m, n)$ tq

$$\|A - UV^*\| \leq \varepsilon \|A\|$$

- Problème : trouver U et V
 - en garantissant une précision donnée
 - avec le minimum d'opérations



Méthode de compression

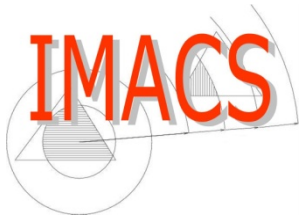
- Utilisation de la décomposition en valeur singulière (SVD) : si $\sigma_1 \geq \dots \geq \sigma_p$ sont les valeurs singulières de A , le rang r à ε près est donné par

$$\sigma_r > \varepsilon \sigma_1 \text{ et } \sigma_{r+1} \leq \varepsilon \sigma_1$$

- En gardant les r premiers vecteurs singuliers à droite et à gauche, on a

$$\|A - U_r \Sigma_r V_r\|_2 \leq \varepsilon \|A\|_2$$

- Méthode robuste mais **chère**. On cherche à calculer U et V « à la volée »



Deux familles de méthodes de compression



- Méthodes à base d'interpolation du noyau : méthode générale ne réclamant pas de développement en série...
- Méthodes algébriques applicables à des matrices quelconques (qui ignorent la forme du noyau)

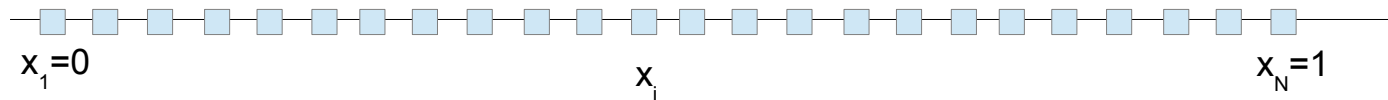


Exemple simple

- Noyau du Laplacien en 2D

$$G(x, y) = \begin{cases} -\ln |x - y| & \text{si } x \neq y \\ 0 & \text{sinon} \end{cases}$$

$$x_1 < x_2 < \dots < x_N = 1$$

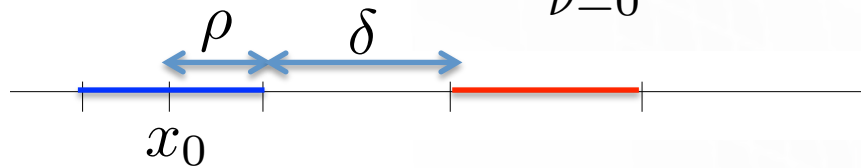


$$G_{i,j} = G(x_i, x_j) \quad i, j = 1, \dots, N$$

Exemple simple

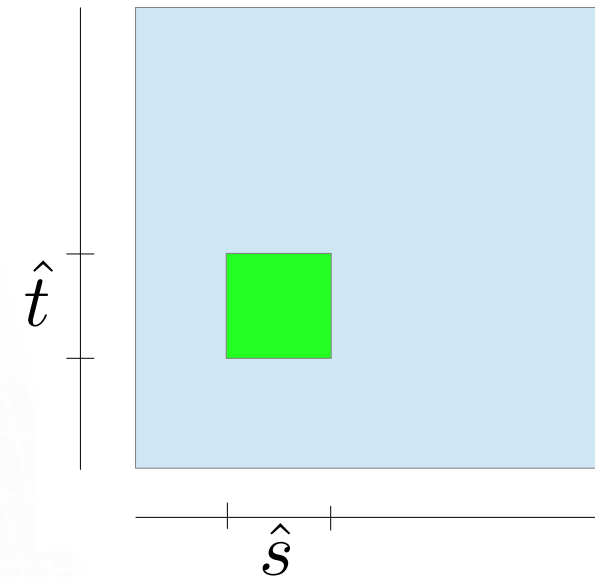
- Approximation de rang r du noyau basée sur la formule de Taylor :

$$G(x, y) \approx \tilde{G}(x, y) = \sum_{\nu=0}^{r-1} \frac{(x - x_0)^\nu}{\nu!} \frac{\partial^\nu G}{\partial x^\nu}(x_0, y) \quad x \in t, y \in s$$



$$G_{\hat{s} \times \hat{t}} \approx UV^T$$

$$U_{i,\nu} = \frac{(x_i - x_0)^\nu}{\nu!}, \quad V_{j,\nu} = \frac{\partial^\nu G}{\partial x^\nu}(x_0, y_j)$$





Exemple simple

- Erreur d'approximation (cv exp. si $\delta > 0$) :

$$|G(x, y) - \tilde{G}(x, y)| \leq \left(\frac{\rho}{\delta + \rho} \right)^{r-1} \ln(1 + \rho/\delta) \quad \text{pour } x \in t, y \in s$$

- Condition d'admissibilité :

$$\text{diam}(t) \leq 2\eta \text{dist}(t, s)$$

- On a alors :

$$\|G - \tilde{G}\|_{\infty, t \times s} \leq \left(\frac{\eta}{1 + \eta} \right)^{r-1} \ln(1 + \eta)$$

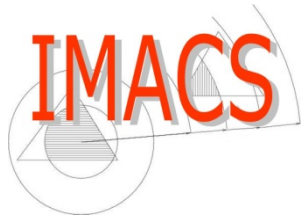
- Choix de r en fonction de l'erreur cible.



HCA

- Au lieu d'un développement de Taylor, on peut utiliser une approximation de G sur $t \times s$ en interpolant avec les polynômes de Lagrange en utilisant les points de Tchebytchev

$$G(x, y) \approx \sum_{\nu} \sum_{\mu} G(\xi_{\nu}, \eta_{\mu}) \mathcal{L}_{\nu}(x) \mathcal{L}_{\mu}(y)$$



Adaptive Cross Approximation

- Résultats théoriques de Goreinov-Tyrtysnikov (*skeleton*)
- Si le bloc matriciel A est déjà assemblée, une méthode rapide pour le compresser consiste à appliquer la méthode d'élimination de Gauss avec pivot total jusqu'à ce qu'on obtienne un pivot nul ou plutôt petit !

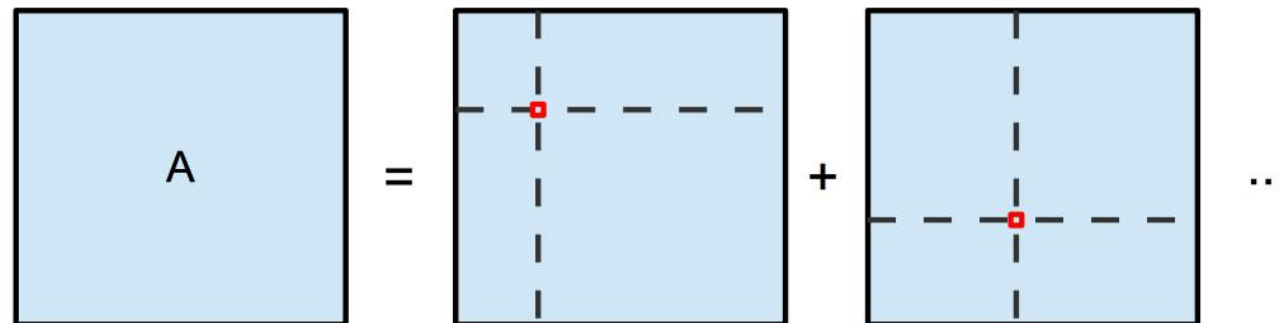
Tant que $A \neq 0$

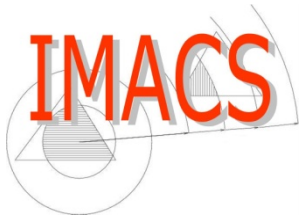
$k++$

trouver le plus grand $|a_{ij}|$

$$A = A - \frac{1}{A(i,j)} * A(:,j) * A(i,:)$$

- C'est l'algorithme **ACA total**





ACA partiel

- Remarquons qu'à chaque itération, nous effectuons une opération de rang 1, et nous avons besoin d'une ligne et d'une colonne de la matrice
- C'est uniquement la recherche du pivot total qui nécessite de connaître a priori toute la matrice.
- Heuristique pour le choix du pivot
- L'algorithme peut échouer (tout comme la factorisation de Gauss). → Algorithme ACA+ est plus robuste



Version fonctionnelle (HCA2)

- ACA de la matrice croisant les points de Tchebytchev :

$$G_{i,j} = G(\xi_i, \eta_j)$$

- Approximation de rang 1 (premier pivot) :

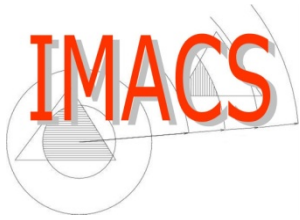
$$G(x, y) \approx G_1(x, y) = \frac{G(x, \eta_{j_1})G(\xi_{i_1}, y)}{G(\xi_{i_1}, \eta_{j_1})}$$

- On approche successivement

$$G(x, y) - \sum_{l=1}^{k-1} G_l(x, y)$$

- Utilisant les pivots du ACA de $G(\xi_{i_1}, \eta_{j_1})$

$$G(x, y) \approx \sum_{l=1}^k \left(\sum_{q=1}^l G(x, \eta_{j_q}) C_{l,q} \right) \left(\sum_{q=1}^l G(\xi_{i_q}, y) D_{l,q} \right)$$

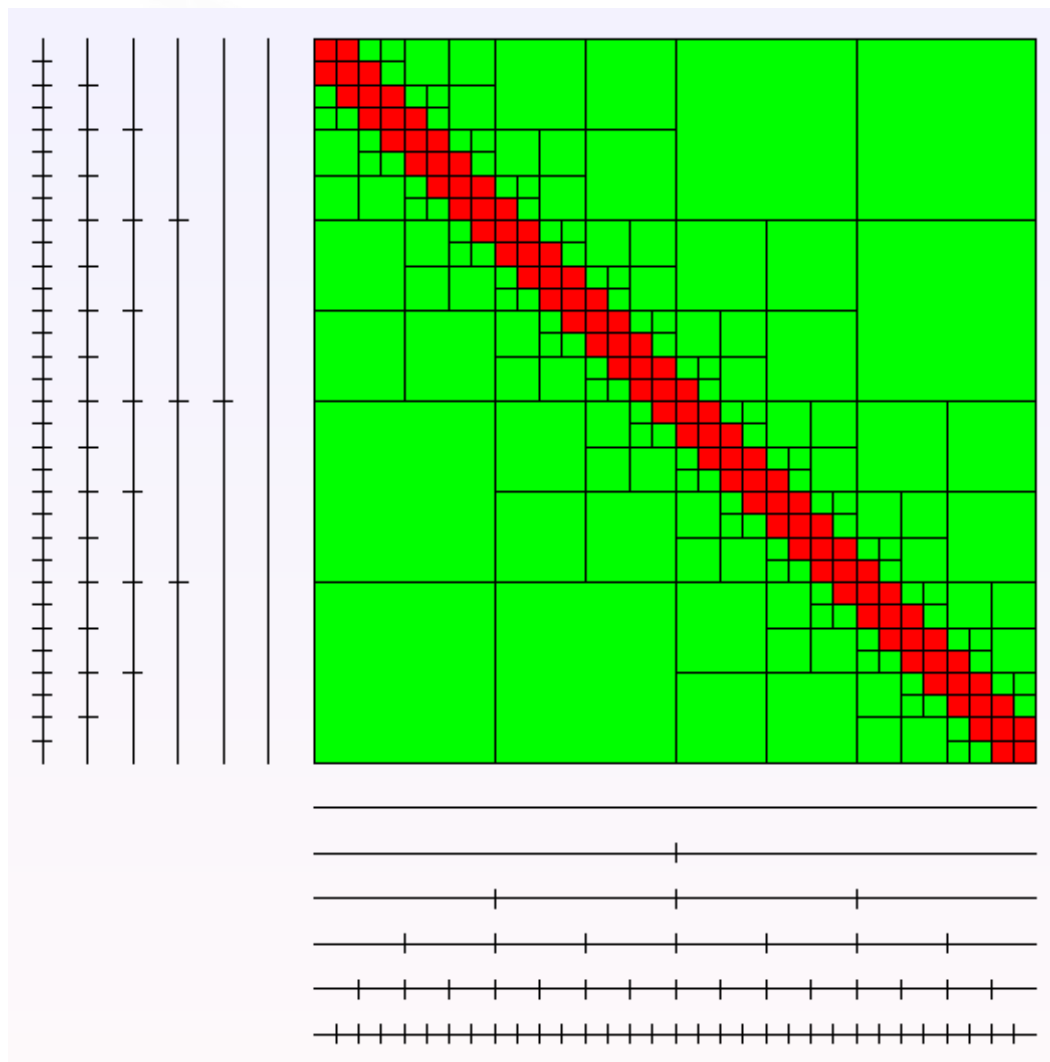


Matrices hiérarchiques

- Partitionner récursivement les DL (bissection) : « *cluster tree* »
- On croise les clusters tree ligne et colonne en commençant par le haut de l'arbre : on s'arrête aux couples de noeuds admissibles de même niveau (Rk matrice) ou aux couples de feuilles non admissibles (blocs pleins) : « *block cluster tree* »

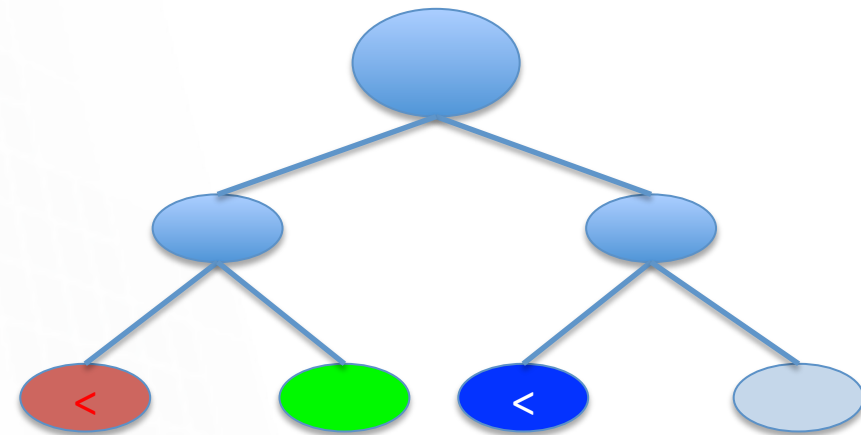
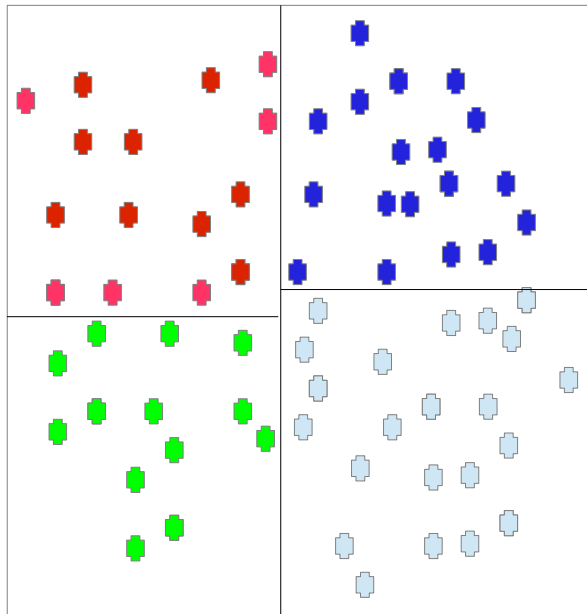
$$K[u](x) = \int k(x,y)u(y)dy$$

H-Matrice



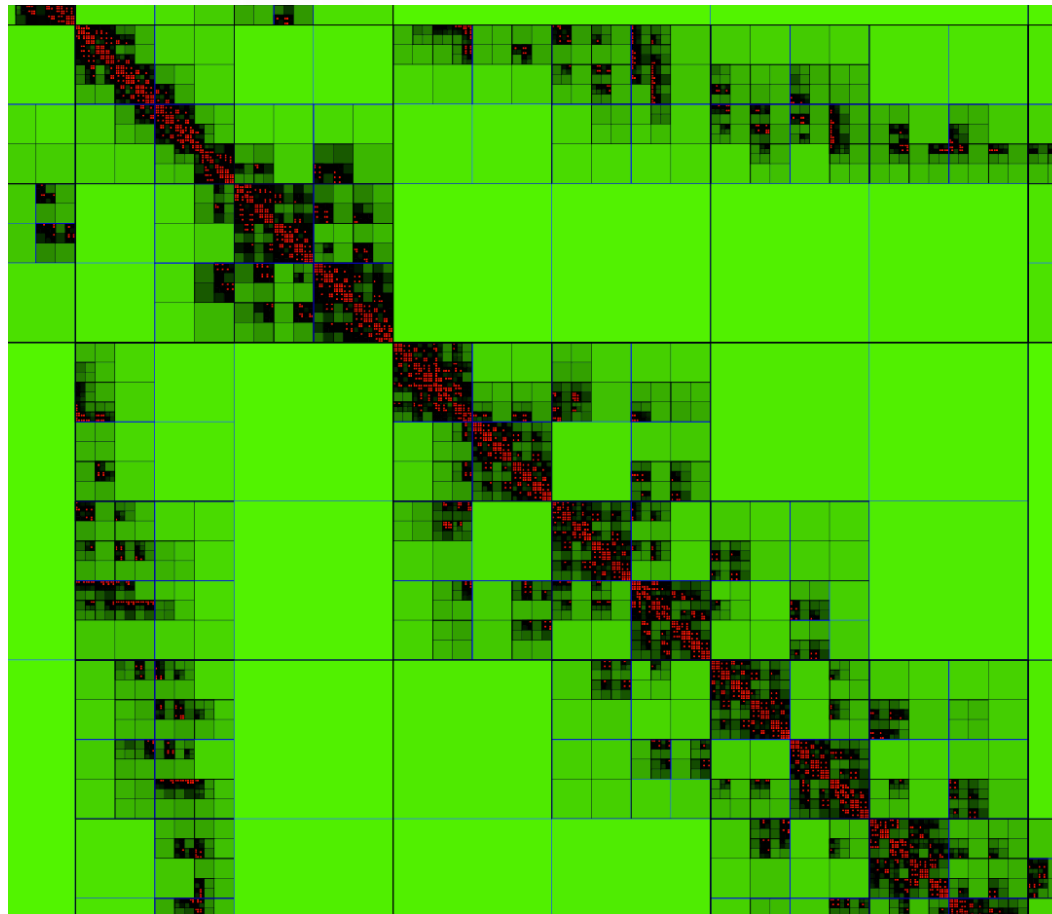
Dimensions supérieures

- Clustering par bisection (géométrique, médiane, hybride)

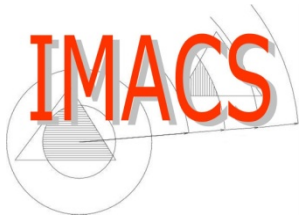




Exemple clustering géométrique



Thèse B. Lizé (2014)



Opérations

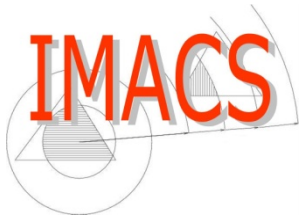
- *Opérations récursives* sur des matrices blocs 2x2
- Somme, produit de H-matrices (cluster trees compatibles)
- Produit matrice-vecteur
- Factorisation : Gauss, Cholesky
 - Lors de la factorisation, le taux de compression peut se détériorer. Le produit de deux R_k matrices est une R_k de rang inférieur au plus petit des 2 rangs. Par contre le rang de la somme peut être égal à la somme des deux rangs...

Factorisation

Exploitation de la structure 2X2 réursive

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$$

$$= \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}$$

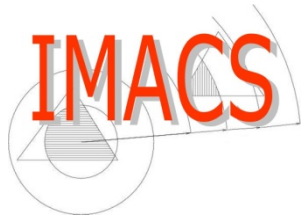


Résultats

- Beaucoup de résultats théoriques pour les noyaux asymptotiquement lisses

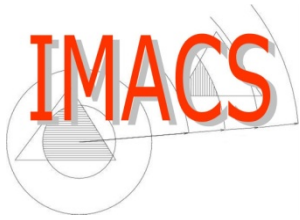
$$|D_y^\alpha G(x, y)| \leq cp! \gamma^{|\alpha|} |x - y|^{-s-|\alpha|} \quad \text{pour } x \neq y$$

- Le noyau de Helmholtz n'est pas asymptotiquement lisse, pourtant dans la pratique la méthode donne d'excellents résultats (cf. thèse B. Lizé '2014).
Éléments de justification théorique dans la thèse de K. Delamotte.



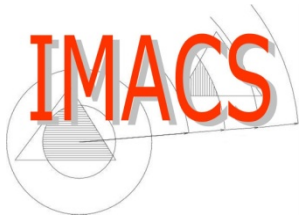
Résumé

- Clustering des DL
- Condition d'admissibilité : dépend des propriétés du noyau (paramètre ϵ). Cas particulier des matrices HODLR. Hypothèse : *noyau régulier en dehors de la diagonale*.
- Block cluster tree
- Méthodes d'assemblage d'un bloc compressé : ACA, HCA, HCA2... Et plein de variantes : autres factorisations, QR, Gram-Schmidt, factorisations randomisées
- Factorisation : c'est une opération purement algébrique sur une H-matrice. La détérioration de la compression dépend des propriétés de la matrice (noyau, distribution du nuage de points...)
- Bisection facilite la programmation récursive
- Paramètres contrôlant la précision : ϵ_{assemb} , $\epsilon_{\text{recompress}}$ à calibrer pour chaque application



Solveur direct vs. itératif

- Stratégie d'utilisation vue d'aujourd'hui :
 - Problèmes de petite taille : méthodes classiques robustes et déjà optimisées
 - Problèmes de grande taille bien conditionnés : méthodes itératives rapides (FMM)
 - Problèmes de grande taille mal conditionnés : méthodes directes rapides (H-matrices)
 - Utilisation des méthodes directes rapides comme pré-conditionneurs d'une méthode itérative
- Projet DGA/RAPID « Hi-BOX » en cours (IMACS + AGI + Inria) : bibliothèque logicielle de solveurs directs, itératifs et hybrides pour matrices BEM.



Applications dans OpenTURNS

- cf. exposé de D. Barbier
- Processus Gaussien : accélération de la factorisation de la matrice de covariance
- Krigeage