Eléments finis spectraux en maillage hexaédrique pour l'équation de Helmholtz

M. Duruflé, G. Cohen

INRIA Rocquencourt

Problème modèle

$$-\omega^2 \frac{1}{\rho(x)\, c^2(x)} u(x) - \mathrm{div}\left(\frac{1}{\rho(x)} \nabla u(x)\right) = f(x) \qquad x \in \Omega$$

Problème modèle

$$-\omega^2 \frac{1}{\rho(x)\, c^2(x)} u(x) - \mathrm{div}\left(\frac{1}{\rho(x)} \nabla u(x)\right) = f(x) \qquad x \in \Omega$$

$$u(x) = -u^{inc}(x) \qquad x \in \Gamma$$

Problème modèle

$$-\omega^2 \frac{1}{\rho(x)\, c^2(x)} u(x) - \mathrm{div}\left(\frac{1}{\rho(x)} \nabla u(x)\right) = f(x) \qquad x \in \Omega$$

$$u(x) = -u^{inc}(x) \qquad x \in \Gamma$$

$$\frac{\partial u}{\partial n}(x) - i\omega\, u(x) = 0 \qquad x \in \Sigma$$

Problème modèle

$$-\omega^2 \frac{1}{\rho(x)\, c^2(x)} u(x) - \text{div}\left(\frac{1}{\rho(x)} \nabla u(x)\right) = f(x) \qquad x \in \Omega$$

$$u(x) = -u^{inc}(x) \qquad x \in \Gamma$$

$$\frac{\partial u}{\partial n}(x) - i\omega\, u(x) = 0 \qquad x \in \Sigma$$

$u^{inc}(x) = exp(ik\, x)$: onde plane incidente

Problème modèle

$$-\omega^2 \frac{1}{\rho(x)\, c^2(x)} u(x) - \text{div}\left(\frac{1}{\rho(x)} \nabla u(x)\right) = f(x) \qquad x \in \Omega$$

$$u(x) = -u^{inc}(x) \qquad x \in \Gamma$$

$$\frac{\partial u}{\partial n}(x) - i\omega\, u(x) = 0 \qquad x \in \Sigma$$

$u^{inc}(x) = exp(ik\, x)$: onde plane incidente

$c(x)$: vitesse de propagation de l'onde

Approximation

Maillage hexaédrique : $\mathcal{M}_h = \bigcup_{j=1}^{N_e} K_j$

Approximation

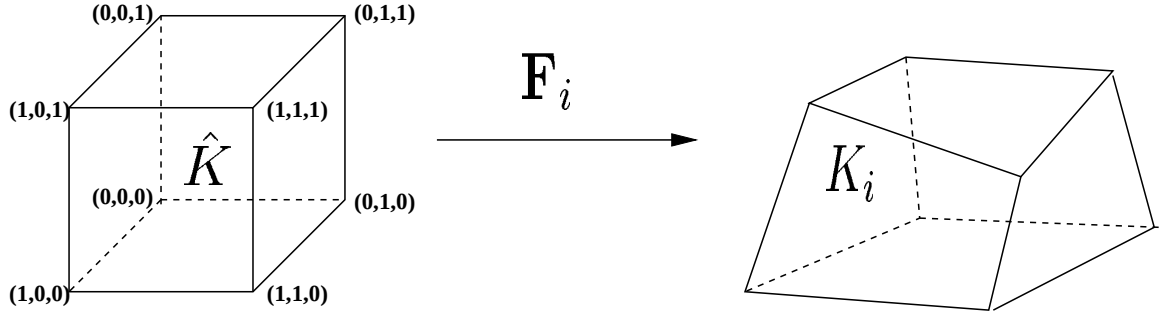
Maillage hexaédrique : $\mathcal{M}_h = \bigcup_{j=1}^{N_e} K_j$

$\widehat{K} = [0,1]^3$, K_j : un hexaèdre j

Approximation

Maillage hexaédrique : $\mathcal{M}_h = \bigcup_{j=1}^{N_e} K_j$

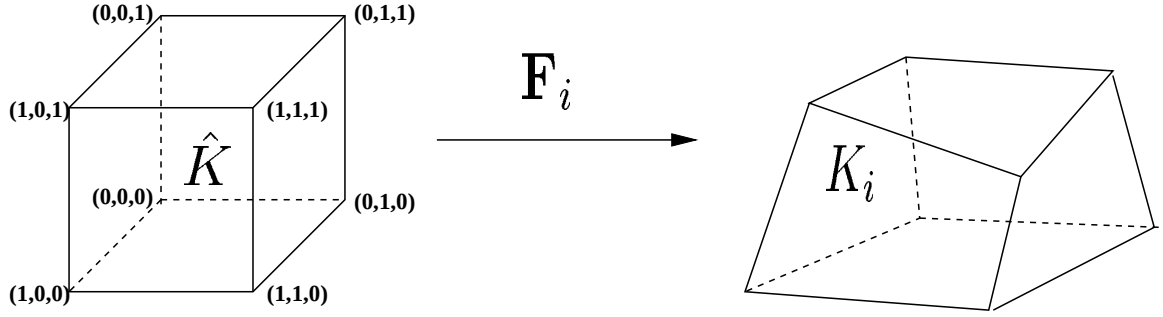
$\widehat{K} = [0,1]^3, \qquad K_j$: un hexaèdre j



Approximation

Maillage hexaédrique : $\mathcal{M}_h = \bigcup_{j=1}^{N_e} K_j$

$\widehat{K} = [0,1]^3$, K_j : un hexaèdre j

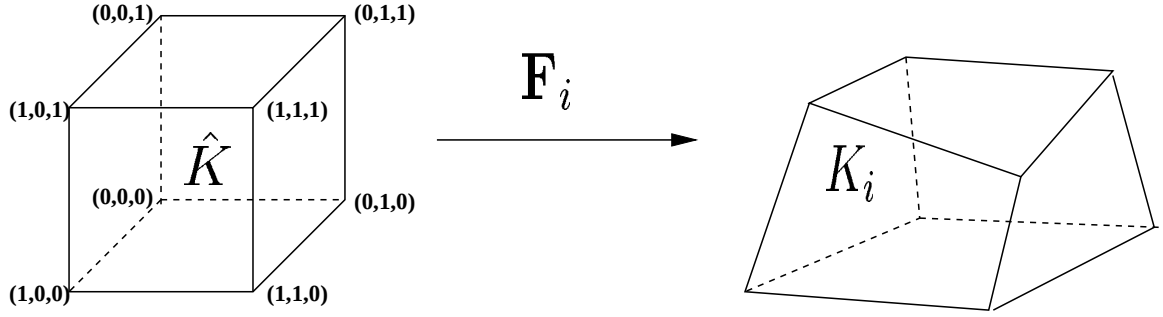


DF_i est la matrice jacobienne de F_i, $J_i = \det DF_i$.

Approximation

Maillage hexaédrique : $\mathcal{M}_h = \bigcup_{j=1}^{N_e} K_j$

$\widehat{K} = [0,1]^3, \qquad K_j$: un hexaèdre j



DF_i est la matrice jacobienne de F_i, $J_i = \det DF_i$.

Q_r est l'espace des polynômes en $\hat{x} \in \hat{K}$ de degré inférieur ou égal à r en chaque variable.

Équation discrète

$$(-\omega^2 D_h + K_h)U_h = F_h$$

Équation discrète

$$(-\omega^2 D_h + K_h)U_h = F_h$$

$$(D_h)_{i,j} = \int_\Omega \varphi_i(x)\,\varphi_j(x)\,dx$$

Équation discrète

$$(-\omega^2 D_h + K_h)U_h = F_h$$

$$(D_h)_{i,j} = \int_\Omega \varphi_i(x)\,\varphi_j(x)\,dx$$

$$(K_h)_{i,j} = \int_\Omega \nabla\varphi_i(x)\cdot\nabla\varphi_j(x)\,dx$$

Équation discrète

$$(-\omega^2 D_h + K_h)U_h = F_h$$

$$(D_h)_{i,j} = \int_\Omega \varphi_i(x)\,\varphi_j(x)\,dx$$

$$(K_h)_{i,j} = \int_\Omega \nabla\varphi_i(x)\cdot\nabla\varphi_j(x)\,dx$$

$$(F_h)_i = \int_\Omega \varphi_i(x)\,f(x)\,dx$$

Équation discrète

$$(-\omega^2 D_h + K_h)U_h = F_h$$

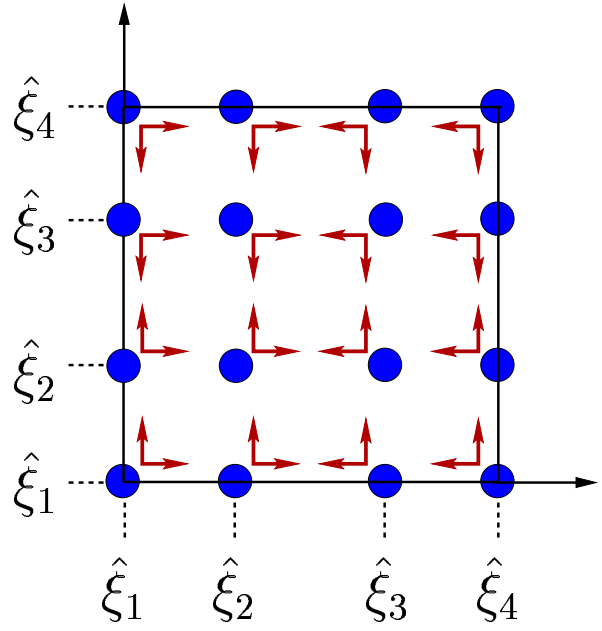
$$(D_h)_{i,j} = \int_\Omega \varphi_i(x)\,\varphi_j(x)\,dx$$

$$(K_h)_{i,j} = \int_\Omega \nabla\varphi_i(x)\cdot\nabla\varphi_j(x)\,dx$$

$$(F_h)_i = \int_\Omega \varphi_i(x)\,f(x)\,dx$$

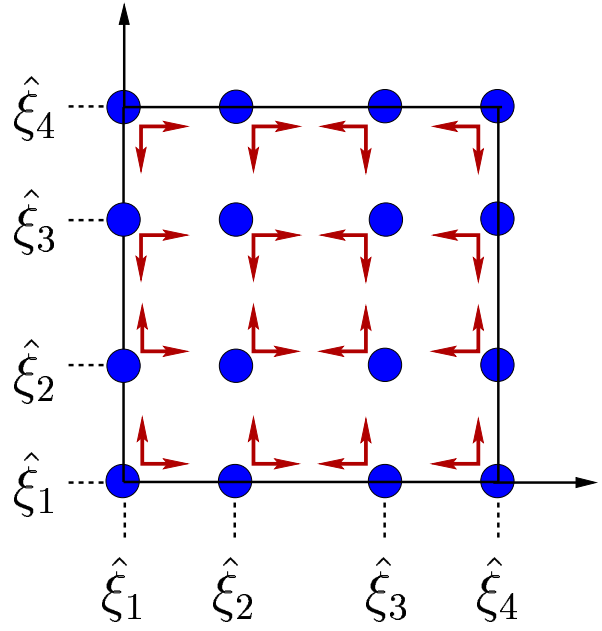
$$u, \varphi \in U_h = \{v \in H_0^1(\Omega) \quad \text{tel que } v \circ F_i \in Q_r\}$$

Degrés de liberté



Degrés de liberté pour u (ronds) et $\vec{v}$ (flèches)

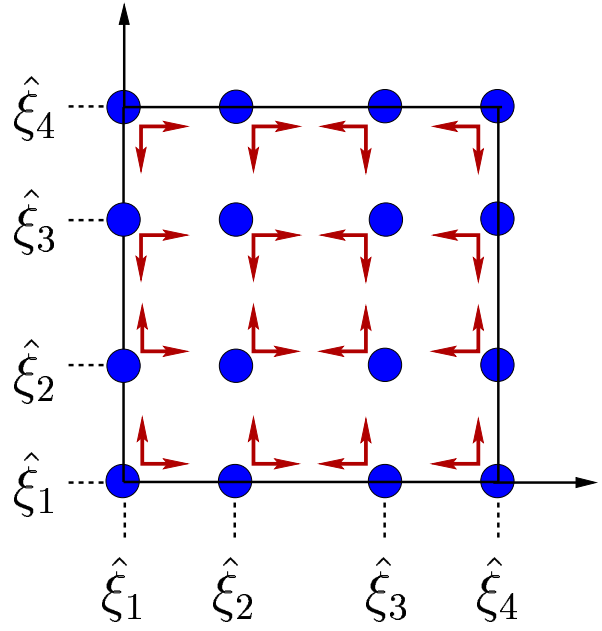
Degrés de liberté



Degrés de liberté pour u (ronds) et $\vec{v}$ (flèches)

Degrés de liberté et points de quadrature coincident sur les points de Gauss-Lobatto

Degrés de liberté



Degrés de liberté pour u (ronds) et $\vec{v}$ (flèches)

Degrés de liberté et points de quadrature coincident sur les points de Gauss-Lobatto

$\Rightarrow$ Condensation de masse (cf P. Grob), D_h matrice diagonale.

Factorisation de la matrice de rigidité

$$K_h = R_h \, B_h^{-1} \, R_h^t$$

Factorisation de la matrice de rigidité

$$K_h = R_h \, B_h^{-1} \, R_h^t$$

$$(R_h)_{i,j} = \int_{\hat{K}} \hat{\nabla}\hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

Factorisation de la matrice de rigidité

$$K_h = R_h \, B_h^{-1} \, R_h^t$$

$$(R_h)_{i,j} = \int_{\hat{K}} \hat{\nabla} \hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

$$(B_h)_{i,j} = \int_{\hat{K}} \frac{1}{J_e} \, DF_e^t \, DF_e \, \vec{\hat{\psi}}_i \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

Factorisation de la matrice de rigidité

$$K_h \;=\; R_h\, B_h^{-1}\, R_h^t$$

$$(R_h)_{i,j} \;=\; \int_{\hat{K}} \hat{\nabla}\hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

$$(B_h)_{i,j} \;=\; \int_{\hat{K}} \frac{1}{J_e}\, DF_e^t\, DF_e\, \vec{\hat{\psi}}_i \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

- S'obtient à l'aide d'une formulation mixte (cf P. Grob)

Factorisation de la matrice de rigidité

$$K_h \;=\; R_h\, B_h^{-1}\, R_h^t$$

$$(R_h)_{i,j} \;=\; \int_{\hat{K}} \hat{\nabla}\hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

$$(B_h)_{i,j} \;=\; \int_{\hat{K}} \frac{1}{J_e}\, DF_e^t\, DF_e\, \vec{\hat{\psi}}_i \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

- S'obtient à l'aide d'une formulation mixte (cf P. Grob)
- B_h diagonale par blocs 3x3.

Factorisation de la matrice de rigidité

$$K_h = R_h \, B_h^{-1} \, R_h^t$$

$$(R_h)_{i,j} = \int_{\hat{K}} \hat{\nabla}\hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

$$(B_h)_{i,j} = \int_{\hat{K}} \frac{1}{J_e} \, DF_e^t \, DF_e \, \vec{\hat{\psi}}_i \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

- S'obtient à l'aide d'une formulation mixte (cf P. Grob)
- B_h diagonale par blocs 3x3.
- Matrice élémentaire de R_h indépendante de la géométrie

Factorisation de la matrice de rigidité

$$K_h = R_h \, B_h^{-1} \, R_h^t$$

$$(R_h)_{i,j} = \int_{\hat{K}} \hat{\nabla}\hat{\varphi}_i(\hat{x}) \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

$$(B_h)_{i,j} = \int_{\hat{K}} \frac{1}{J_e} \, DF_e^t \, DF_e \, \vec{\hat{\psi}}_i \cdot \vec{\hat{\psi}}_j \, d\hat{x}$$

- S'obtient à l'aide d'une formulation mixte (cf P. Grob)
- B_h diagonale par blocs 3x3.
- Matrice élémentaire de R_h indépendante de la géométrie
- Stockage optimal

Caractère creux de la matrice de rigidité élémentaire

Tensorisation d'ordre 3 des fonctions de base

Caractère creux de la matrice de rigidité élémentaire

Tensorisation d'ordre 3 des fonctions de base

$$\hat{\varphi}_i(\hat{x}) = \hat{\varphi}_{i_1}(\hat{x})\,\hat{\varphi}_{i_2}(\hat{y})\,\hat{\varphi}_{i_3}(\hat{z}) \qquad \vec{\hat{\psi}}_j(\hat{x}) = \hat{\varphi}_{j_1}(\hat{x})\,\hat{\varphi}_{j_2}(\hat{y})\,\hat{\varphi}_{j_3}(\hat{z})\,\vec{e_1}$$

Caractère creux de la matrice de rigidité élémentaire

Tensorisation d'ordre 3 des fonctions de base

$$\hat{\varphi}_i(\hat{x}) \;= \hat{\varphi}_{i_1}(\hat{x})\,\hat{\varphi}_{i_2}(\hat{y})\,\hat{\varphi}_{i_3}(\hat{z}) \qquad \vec{\hat{\psi}}_j(\hat{x}) \;= \hat{\varphi}_{j_1}(\hat{x})\,\hat{\varphi}_{j_2}(\hat{y})\,\hat{\varphi}_{j_3}(\hat{z})\;\vec{e_1}$$

$$(R_h)_{i,j} \;=\; \omega_j\;\hat{\varphi}'_{i_1}(\hat{\xi}_{j_1})\;\delta_{i_2,j_2}\;\delta_{i_3,j_3}$$

Caractère creux de la matrice de rigidité élémentaire

Tensorisation d'ordre 3 des fonctions de base

$$\hat{\varphi}_i(\hat{x}) \;= \hat{\varphi}_{i_1}(\hat{x})\,\hat{\varphi}_{i_2}(\hat{y})\,\hat{\varphi}_{i_3}(\hat{z}) \qquad \vec{\hat{\psi}}_j(\hat{x}) \;= \hat{\varphi}_{j_1}(\hat{x})\,\hat{\varphi}_{j_2}(\hat{y})\,\hat{\varphi}_{j_3}(\hat{z})\,\vec{e_1}$$

$$(R_h)_{i,j} \;=\; \omega_j\,\hat{\varphi}'_{i_1}(\hat{\xi}_{j_1})\,\delta_{i_2,j_2}\,\delta_{i_3,j_3}$$

- Présence de $\delta_{i_2,j_2}\delta_{i_3,j_3} \Rightarrow$ matrice élémentaire de R_h creuse.

Caractère creux de la matrice de rigidité élémentaire

Tensorisation d'ordre 3 des fonctions de base

$$\hat{\varphi}_i(\hat{x}) \;=\hat{\varphi}_{i_1}(\hat{x})\,\hat{\varphi}_{i_2}(\hat{y})\,\hat{\varphi}_{i_3}(\hat{z}) \qquad \vec{\hat{\psi}}_j(\hat{x}) \;=\hat{\varphi}_{j_1}(\hat{x})\,\hat{\varphi}_{j_2}(\hat{y})\,\hat{\varphi}_{j_3}(\hat{z})\;\vec{e_1}$$

$$(R_h)_{i,j} \;=\; \omega_j\;\hat{\varphi}'_{i_1}(\hat{\xi}_{j_1})\;\delta_{i_2,j_2}\;\delta_{i_3,j_3}$$

- Présence de $\delta_{i_2,j_2}\delta_{i_3,j_3} \Rightarrow$ matrice élémentaire de R_h creuse.
- Matrice R_h pleine dans le cas des tétraèdres (pas de tensorisation)

$\Rightarrow$ Gain en temps de calcul du produit matrice vecteur $Y \;=\; K_h\,X$

Calcul pratique

- Méthode de Newton-Fourier pour calculer les points de Gauss-Lobatto à n'importe quel ordre
  cf P. N. Swarztrauber, SIAM (2002)

Calcul pratique

- Méthode de Newton-Fourier pour calculer les points de Gauss-Lobatto à n'importe quel ordre
  cf P. N. Swarztrauber, SIAM (2002)
- Utilisation des expressions analytiques des fonctions de base, expressions génériques selon l'ordre

Calcul pratique

- Méthode de Newton-Fourier pour calculer les points de Gauss-Lobatto à n'importe quel ordre
  cf P. N. Swarztrauber, SIAM (2002)
- Utilisation des expressions analytiques des fonctions de base, expressions génériques selon l'ordre
- Utilisation du stockage morse pour la matrice élémentaire R_h

Numérotation locale et globale

Données du maillage :

- Sommets (coordonnées)
- Hexaèdres (numéros des 8 sommets pour chaque hexaèdre)

Numérotation locale et globale

Données du maillage :

- Sommets (coordonnées)
- Hexaèdres (numéros des 8 sommets pour chaque hexaèdre)

Utilisation d'algorithmes optimaux de recherche d'arêtes et de faces (cf F. Hecht (*))

Numérotation locale et globale

Données du maillage :

- Sommets (coordonnées)
- Hexaèdres (numéros des 8 sommets pour chaque hexaèdre)

Utilisation d'algorithmes optimaux de recherche d'arêtes et de faces (cf F. Hecht (*))

Numérotation des ddl, dans l'ordre :

- ddl sur les sommets

Numérotation locale et globale

Données du maillage :

- Sommets (coordonnées)
- Hexaèdres (numéros des 8 sommets pour chaque hexaèdre)

Utilisation d'algorithmes optimaux de recherche d'arêtes et de faces (cf F. Hecht (*))

Numérotation des ddl, dans l'ordre :

- ddl sur les sommets
- ddl sur les arêtes, puis sur les faces (*)

Numérotation locale et globale

Données du maillage :

- Sommets (coordonnées)
- Hexaèdres (numéros des 8 sommets pour chaque hexaèdre)

Utilisation d'algorithmes optimaux de recherche d'arêtes et de faces (cf F. Hecht (*))

Numérotation des ddl, dans l'ordre :

- ddl sur les sommets
- ddl sur les arêtes, puis sur les faces (*)
- ddl internes

lg(e,i) associe au ième ddl de l'élément e, le numéro global du ddl.

Algorithme du produit matrice-vecteur

Y = 0

Pour e = 1, nombre d'hexaèdres du maillage

Algorithme du produit matrice-vecteur

Y = 0

Pour e = 1, nombre d'hexaèdres du maillage

    Pour i = 1, $(r+1)^3$

        Ulocal(i) = Uglobal(lg(e,i))

    Fin Pour

Algorithme du produit matrice-vecteur

Y = 0

Pour e = 1, nombre d'hexaèdres du maillage

Pour i = 1, $(r+1)^3$

Ulocal(i) = Uglobal(lg(e,i))

Fin Pour

// produit matrice-vecteur creux standard

Vlocal = R_h * Ulocal

Algorithme du produit matrice-vecteur

Y = 0

Pour e = 1, nombre d'hexaèdres du maillage

Pour i = 1, $(r+1)^3$

Ulocal(i) = Uglobal(lg(e,i))

Fin Pour

// produit matrice-vecteur creux standard

Vlocal = R_h * Ulocal

// produit par la matrice diagonale par blocs B_h^{-1}

Pour i = 1, $(r+1)^3$

Pour j = 1,3

V(j) = Vlocal(3*(i-1) + j)

Algorithme du produit matrice-vecteur

// produit matrice-vecteur creux standard

Vlocal = R_h * Ulocal

// produit par la matrice diagonale par blocs B_h^{-1}

Pour i = 1, $(r+1)^3$

Pour j = 1,3

V(j) = Vlocal(3*(i-1) + j)

B_h^{-1}_V = $B_h^{-1}(e,i)$ * V

Pour j = 1,3

Vlocal(3*(i-1) + j) = B_h^{-1}_V(j)

Fin Pour

Algorithme du produit matrice-vecteur

// produit par la matrice diagonale par blocs B_h^{-1}

Pour i = 1, $(r+1)^3$

Pour j = 1,3

V(j) = Vlocal(3*(i-1) + j)

B_h^{-1}_V = $B_h^{-1}(e,i)$ * V

Pour j = 1,3

Vlocal(3*(i-1) + j) = B_h^{-1}_V(j)

Fin Pour

// produit matrice-vecteur creux standard

Ulocal = R_h^t * Vlocal

Algorithme du produit matrice-vecteur

B_h^{-1}_V = $B_h^{-1}(e,i)$ * V

Pour j = 1,3

Vlocal(3*(i-1) + j) = B_h^{-1}_V(j)

Fin Pour

// produit matrice-vecteur creux standard

Ulocal = R_h^t * Vlocal

Pour i = 1, $(r+1)^3$

Y(lg(e,i)) += Ulocal(i)

Fin Pour

Fin boucle éléments

Application de l'algorithme

En régime transitoire, itérations en temps :

$$D_h\left(\frac{U^{n+1} - 2\,U^n + U^{n-1}}{\Delta t^2}\right) + K_h U^n = F^n$$

Application de l'algorithme

En régime transitoire, itérations en temps :

$$D_h\left(\frac{U^{n+1} - 2\,U^n + U^{n-1}}{\Delta t^2}\right) + K_h U^n = F^n$$

- Utilisation du produit matrice-vecteur à chaque itération

Application de l'algorithme

En régime transitoire, itérations en temps :

$$D_h\left(\frac{U^{n+1} - 2\,U^n + U^{n-1}}{\Delta t^2}\right) + K_h U^n = F^n$$

- Utilisation du produit matrice-vecteur à chaque itération

En régime harmonique, résolution d'un système linéaire :

$$-\omega^2\, D_h\, U + K_h\, U = F$$

Application de l'algorithme

En régime transitoire, itérations en temps :

$$D_h \left(\frac{U^{n+1} - 2\,U^n + U^{n-1}}{\Delta t^2} \right) + K_h U^n = F^n$$

- Utilisation du produit matrice-vecteur à chaque itération

En régime harmonique, résolution d'un système linéaire :

$$-\omega^2\, D_h\, U + K_h\, U = F$$

- Solveur itératif : utilisation du produit matrice-vecteur

Application de l'algorithme

En régime transitoire, itérations en temps :

$$D_h\left(\frac{U^{n+1} - 2\,U^n + U^{n-1}}{\Delta t^2}\right) + K_h U^n = F^n$$

- Utilisation du produit matrice-vecteur à chaque itération

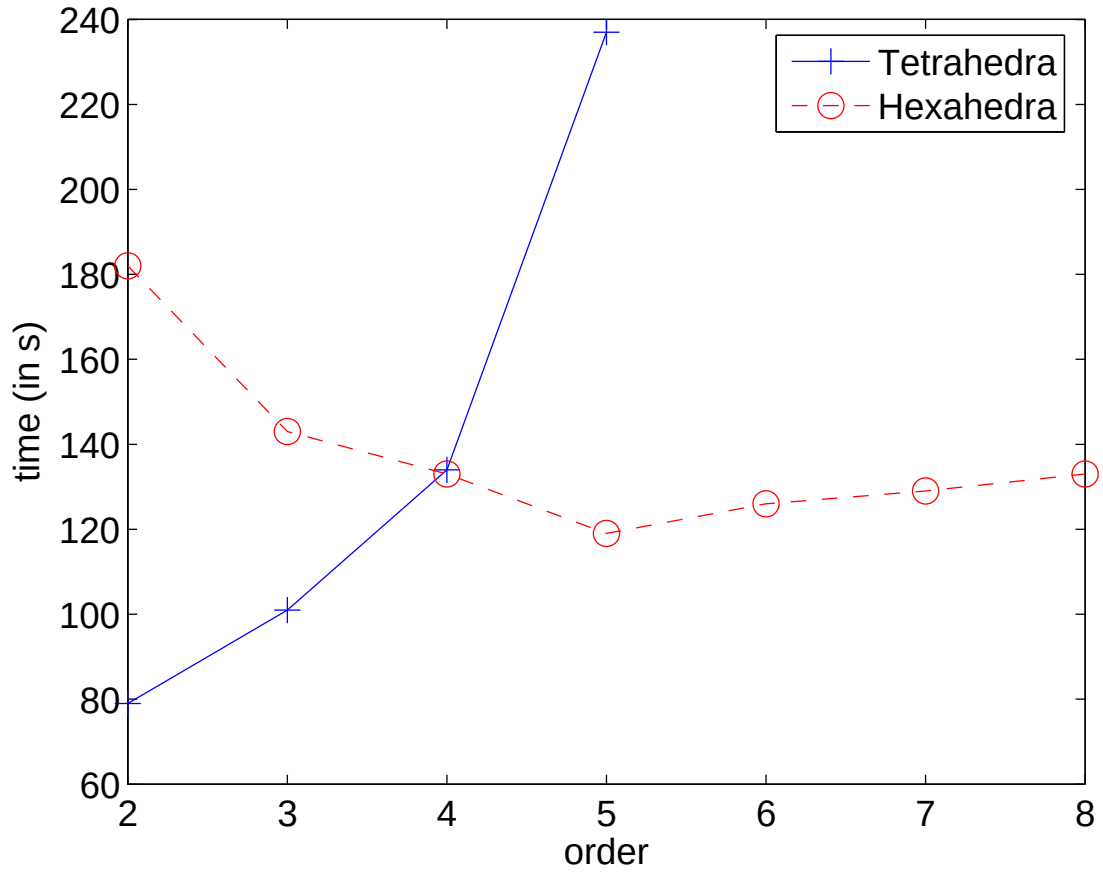
En régime harmonique, résolution d'un système linéaire :

$$-\omega^2\, D_h\, U + K_h\, U = F$$

- Solveur itératif : utilisation du produit matrice-vecteur

Remarque : pour un solveur direct, assemblage classique

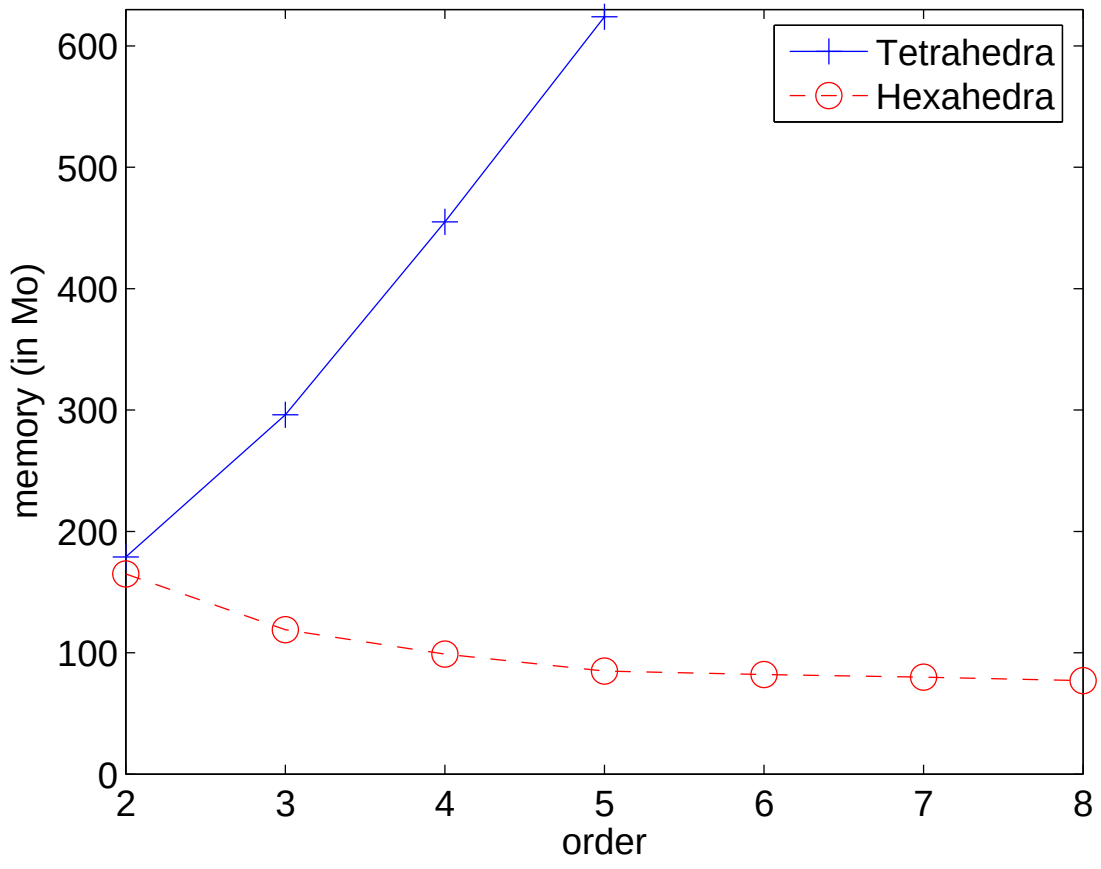
Temps de calcul



Temps de calcul pour 100 produits matrice vecteur

Nombre de ddl constant (environ 1 million)

Stockage



Mémoire utilisée pour stocker la matrice

Nombre de ddl constant (environ 1 million)

Validation sur le cas de la sphère

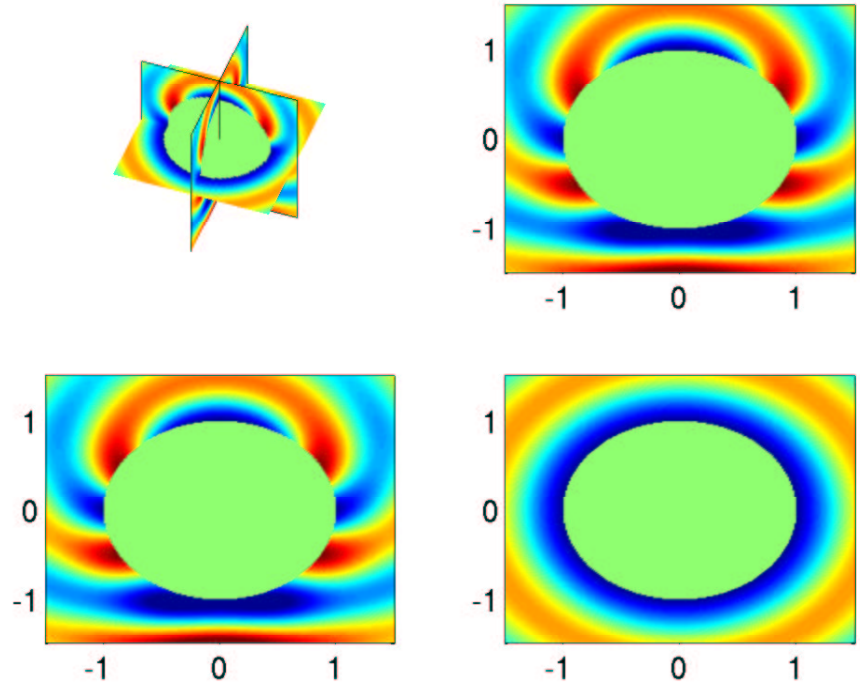
Frontière intérieure Γ = sphère de rayon a

Frontière extérieure Σ = sphère de rayon b

Validation sur le cas de la sphère

Frontière intérieure Γ = sphère de rayon a

Frontière extérieure Σ = sphère de rayon b

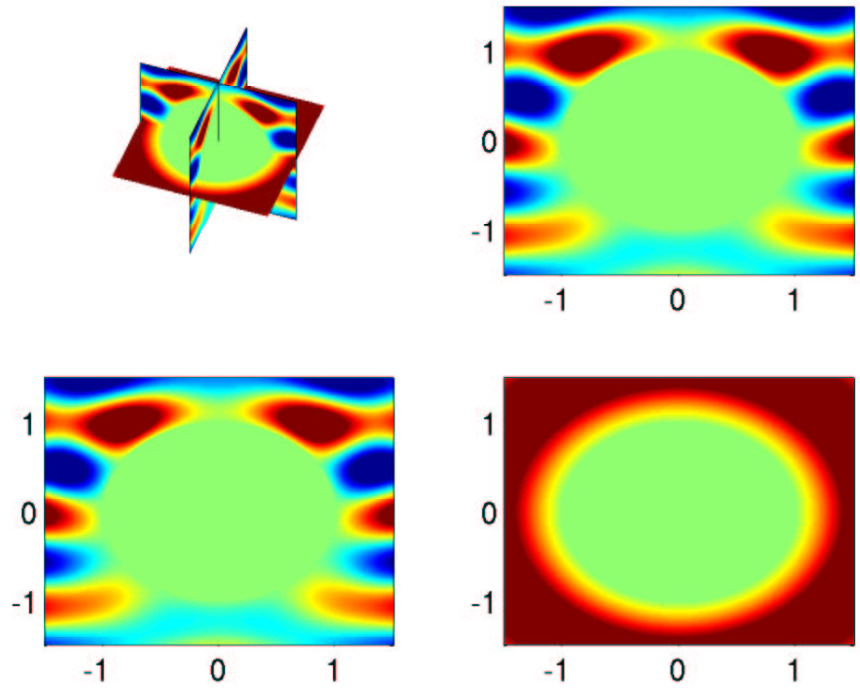


Partie réelle du champ diffracté $a = 1.0 \quad b = 1.5$

Validation sur le cas de la sphère

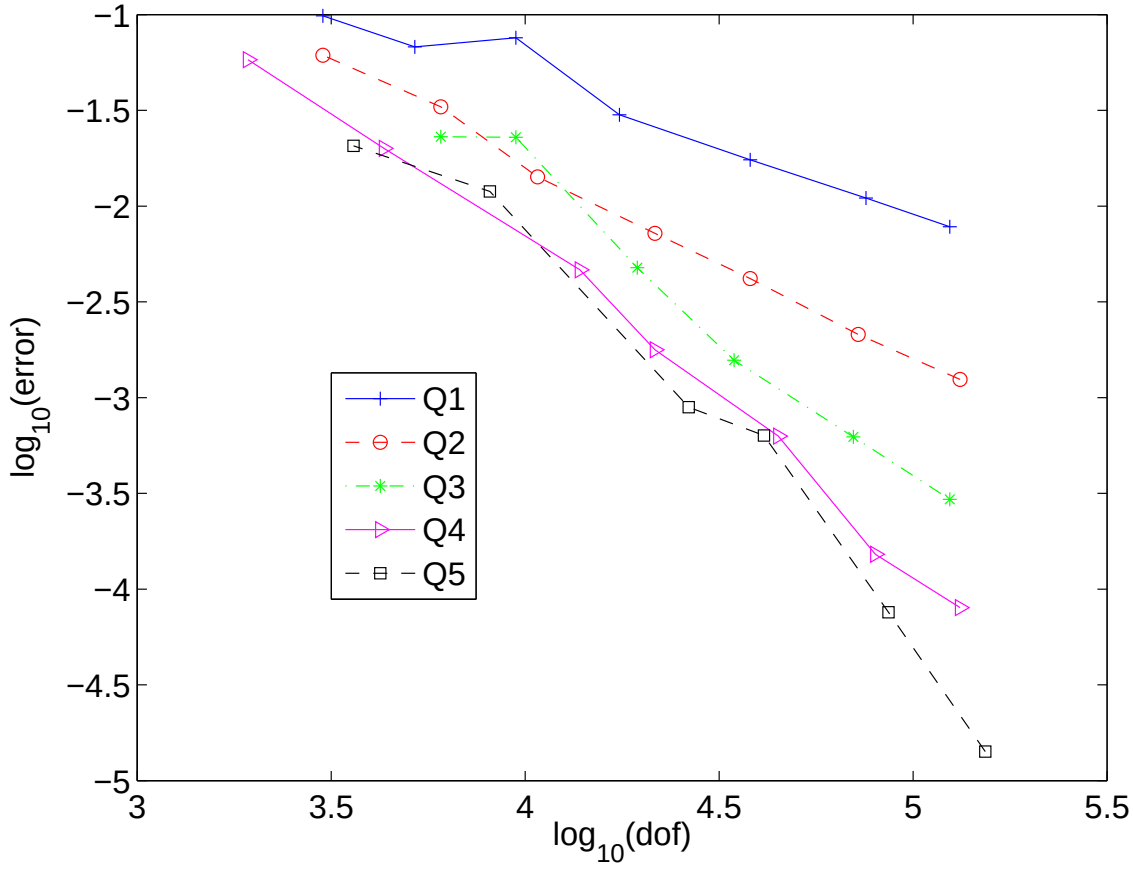
Frontière intérieure Γ = sphère de rayon a

Frontière extérieure Σ = sphère de rayon b



Partie réelle du champ total $a = 1.0 \quad b = 1.5$

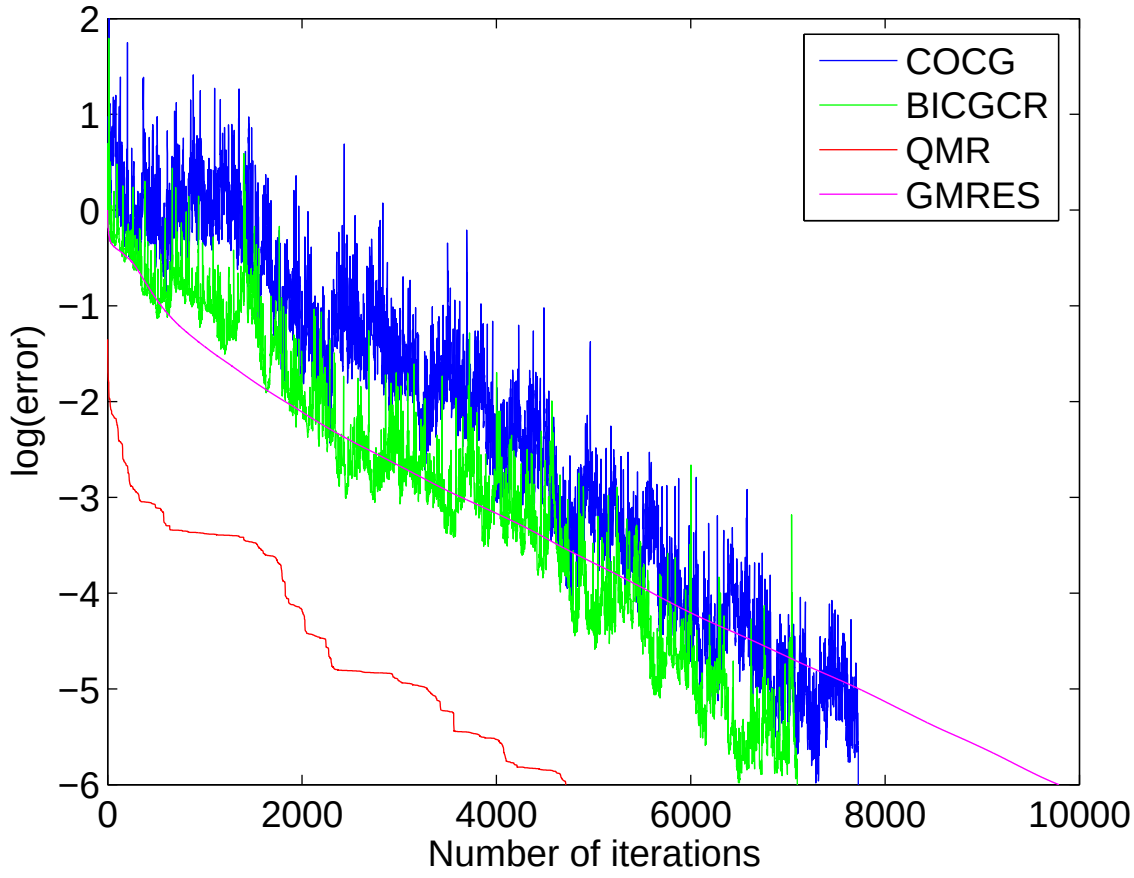
Convergence en fonction de l'ordre



Evolution de l'erreur L^2 en fonction du nombre de ddl, échelle log-log.

Utilisation d'éléments finis courbes.

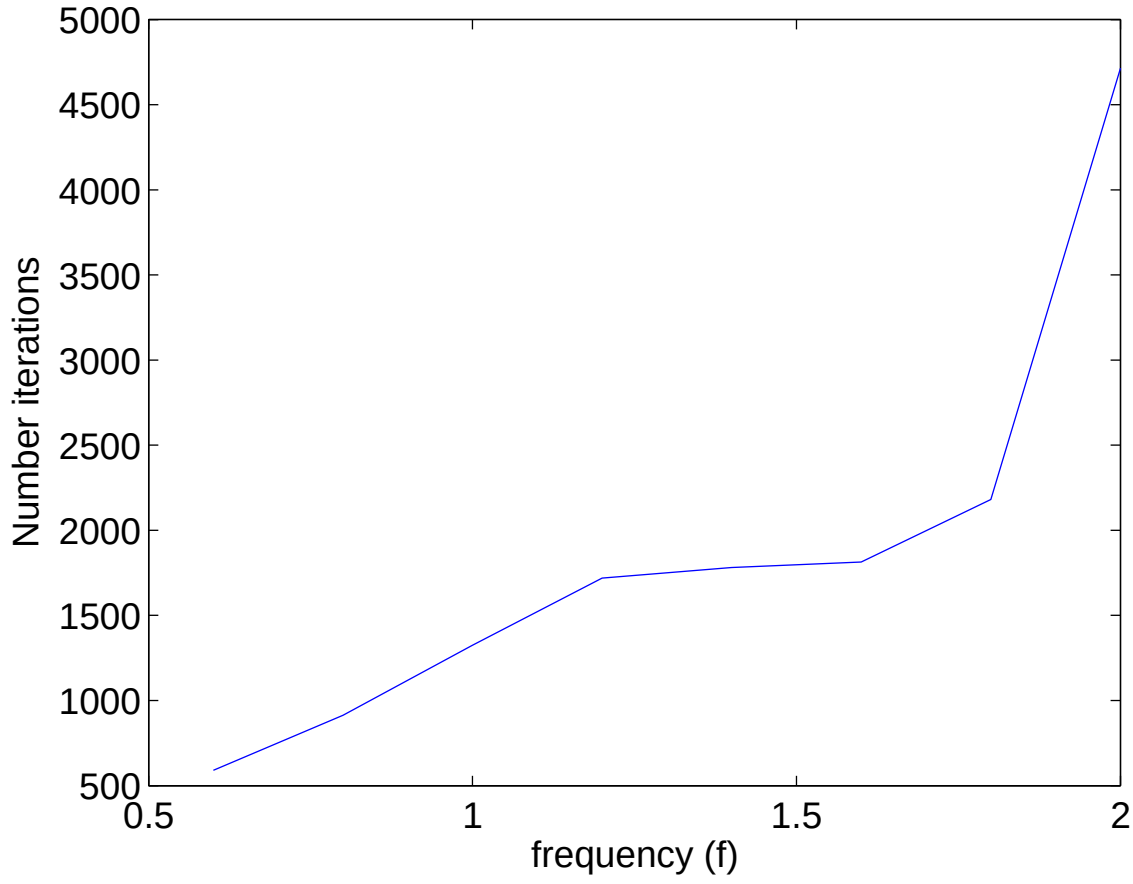
Choix du solveur itératif



Evolution du résidu relatif en fonction du solveur utilisé

Cas haute-fréquence $a = 10 \quad b = 12$

Influence de la fréquence



Nombre d'itérations en fonction de la fréquence (solveur QMR)

Nombre de ddl constant (environ 1,1 million)

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Décomposition en sous-domaines $\Omega = \bigcup_{i=1}^{N_s} \Omega_i$

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Décomposition en sous-domaines $\Omega = \bigcup_{i=1}^{N_s} \Omega_i$

$$M^{-1} = \sum_{i=1}^{N_s} P_i \, A_i^{-1} \, P_i^t$$

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Décomposition en sous-domaines $\Omega = \bigcup_{i=1}^{N_s} \Omega_i$

$$M^{-1} = \sum_{i=1}^{N_s} P_i \, A_i^{-1} \, P_i^t$$

P_i, opérateur de projection de Ω_i vers Ω

A_i matrice élément fini de Ω_i

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Décomposition en sous-domaines $\Omega = \bigcup_{i=1}^{N_s} \Omega_i$

$$M^{-1} = \sum_{i=1}^{N_s} P_i \, A_i^{-1} \, P_i^t$$

P_i, opérateur de projection de Ω_i vers Ω

A_i matrice élément fini de Ω_i

Condition absorbante d'ordre 1 sur les frontières internes des sous-domaines

Préconditionneur utilisé

Méthode de Schwarz additive sans recouvrement

Décomposition en sous-domaines $\Omega = \bigcup_{i=1}^{N_s} \Omega_i$

$$M^{-1} = \sum_{i=1}^{N_s} P_i \, A_i^{-1} \, P_i^t$$

P_i, opérateur de projection de Ω_i vers Ω

A_i matrice élément fini de Ω_i

Condition absorbante d'ordre 1 sur les frontières internes des sous-domaines

Factorisation des matrices A_i à l'aide d'un solveur direct (MUMPS)

Performances avec préconditionneur

Cas moyenne fréquence $a = 5 \quad b = 6$, 170 000 ddl environ

Performances avec préconditionneur

Cas moyenne fréquence $a = 5 \quad b = 6$, 170 000 ddl environ

QMR sans préconditionneur, tolérance de 10^{-6}

Ordre	$Q3$	$Q5$	$Q7$
Nombre itérations	585	826	1305
Temps	$128s$	$154s$	$244s$

Performances avec préconditionneur

Cas moyenne fréquence $a = 5 \quad b = 6$, 170 000 ddl environ

QMR sans préconditionneur, tolérance de 10^{-6}

Ordre	$Q3$	$Q5$	$Q7$
Nombre itérations	585	826	1305
Temps	$128s$	$154s$	$244s$

COCG avec préconditionneur($N_s = 26$), tolérance de 10^{-5}

Ordre	$Q3$	$Q5$	$Q7$
Nombre itérations	91	110	199
Temps	$75s$	$89s$	$230s$