

CANUM 2022 - 16 Juin 2022



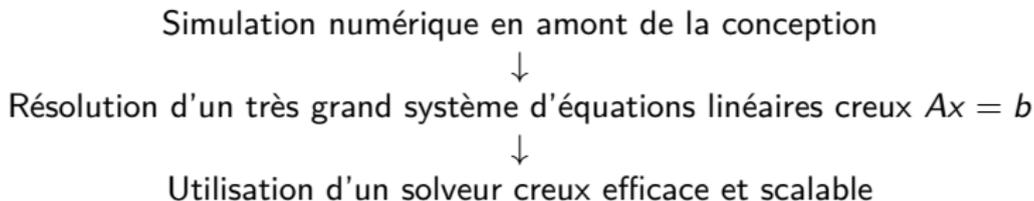
Rob FALGOUT (LLNL)
Matthieu LECOUEZ (CEA/CESTA)
Pierre RAMET (INRIA)
Clément RICHEFORT (CEA/CESTA)

DE LA RECHERCHE À L'INDUSTRIE

Développement d'un solveur creux de type
multigrille pour des problèmes à noyau oscillant

Dans le cadre du programme simulation, le CEA étudie le comportement électromagnétique d'objets 3D complexes grâce aux équations de **Maxwell**.

De manière générale :



Une méthode de **décomposition de domaines** est utilisée dans ce contexte, mais avec l'augmentation des capacités de calculs, certaines limitations apparaissent :

- Augmentation du nombre de sous-domaines \Rightarrow Dégradation de la convergence
- Augmentation de la taille des sous-domaines \Rightarrow Augmentation du temps de calcul

Recherche d'une alternative à la décomposition de domaines : **la méthode multigrille**.

① Les méthodes multigrilles

- 1.1 - Principe général de la méthode
- 1.2 - Application au cas Laplace
- 1.3 - Application au cas Helmholtz

② Opérateur d'interpolation adapté au cas Helmholtz

- 2.1 - Découpage \mathcal{C}/\mathcal{F}
- 2.2 - Définition de l'interpolateur idéal
- 2.3 - Approximation de l'interpolateur idéal
- 2.4 - Définition de règles d'interpolation initiales à l'aide d'information spectrale locale
- 2.5 - Élimination des fréquences résiduelles par une matrice de lissage

③ Conclusion

- 3.1 - Benchmark
- 3.2 - Perspectives

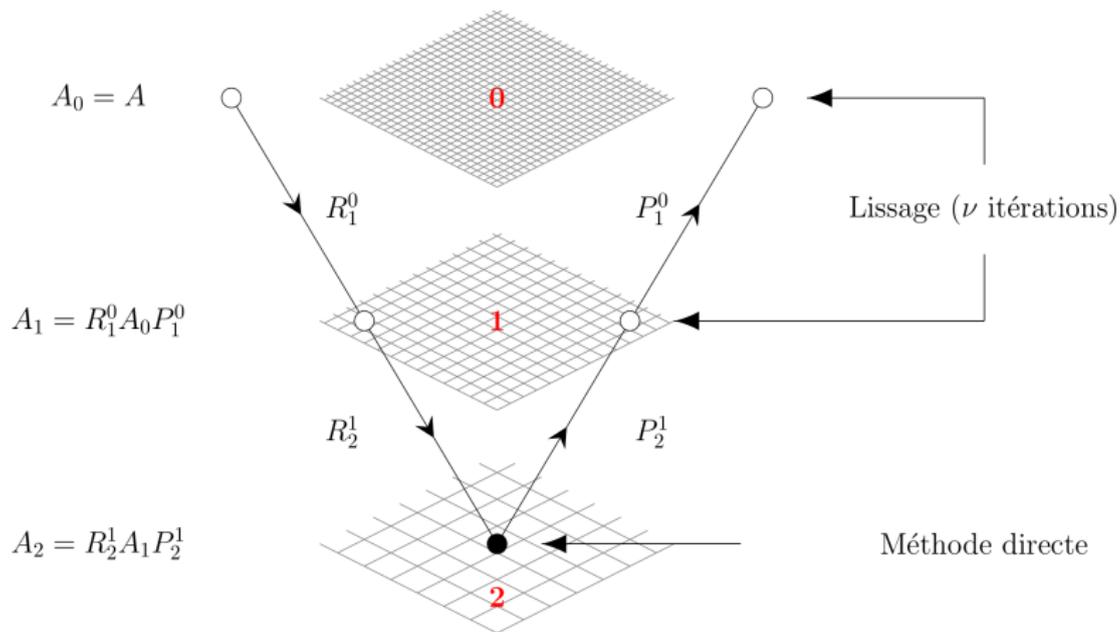


Figure 1: Illustration d'un V-cycle multigrille à 3 niveaux

Réputée scalable et quasi-optimale dans la résolution de problèmes elliptiques

- Processus itératif d'une méthode multigrille à deux niveaux :

$$e^{(i)} = M^\nu (I - PA_C^{-1}P^T A)M^\nu e^{(i-1)}$$

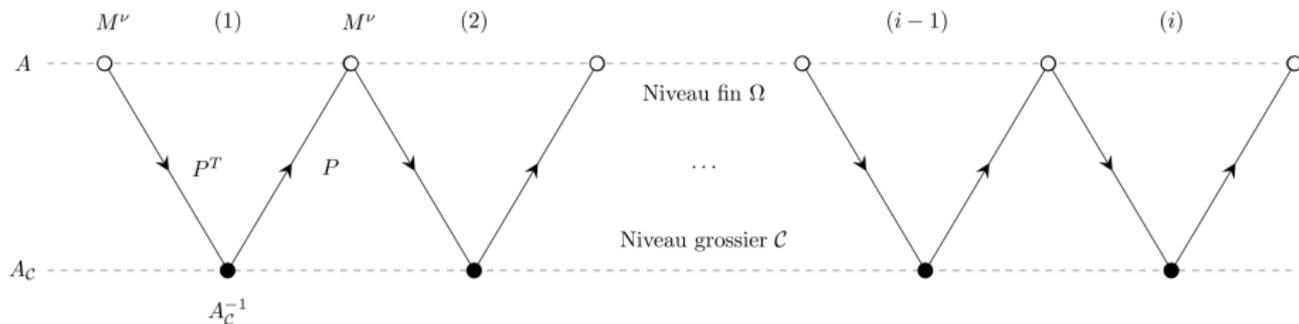


Figure 2: Illustration du processus itératif multigrille

- $e^{(i)}$: Erreur résiduelle après i cycles ($e^{(0)} = x$)
- M^ν : ν itérations d'une méthode de lissage M (exemple : $M_{\text{Jacobi}} = (I - wD^{-1}A)$)
- P : Opérateur d'interpolation du niveau grossier vers le niveau fin
- R : Opérateur de restriction du niveau fin vers le niveau grossier ($R = P^T$)
- A_C^{-1} : Résolution du problème grossier par méthode directe

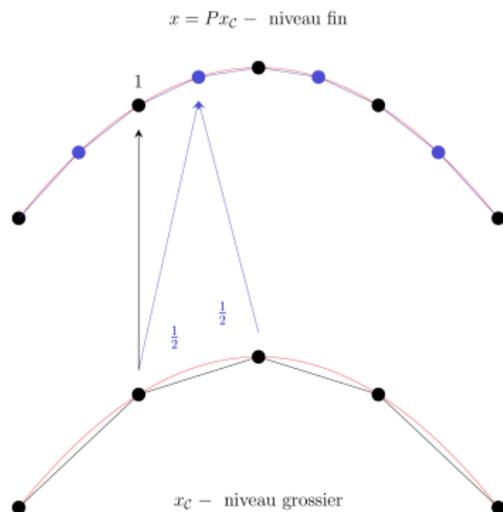


Figure 3: Interpolation triviale

Le transfert du niveau **grossier** vers le **fin** s'effectue par un opérateur d'*interpolation* :

$$P = \frac{1}{2} \begin{bmatrix} 1 & \dots & 0 \\ 2 & & \vdots \\ 1 & 1 & \\ 0 & 2 & \ddots \\ \vdots & 1 & \ddots \\ 0 & \dots & \ddots \end{bmatrix} \quad (1)$$

$$(\text{Problème de Laplace}) \Leftrightarrow \begin{cases} -\Delta u = f \text{ sur } \Omega \\ u|_{\partial\Omega} = 0 \end{cases} \quad (2)$$

En appliquant un schéma de différence finie de 2nd ordre sur une discrétisation uniforme de Ω en 1D:

$$Au = f \quad \text{with} \quad A = \frac{1}{h^2} \text{Tridiag}(-1, 2, -1) \quad (3)$$

En appliquant une analyse locale de Fourier A :

$$\forall j = 1, \dots, n, \quad \lambda_j(A) = \frac{2 - 2 \cos(j\pi h)}{h^2}, \quad v_j(A) = [\sin(lj\pi h)]_{l=1}^n$$

Le principe de complémentarité est respecté :

- Les $\lambda_j(A)$ sont positives \Rightarrow Méthodes de relaxation usuelles fonctionnent (Jacobi,...)
- Le NKS¹ est lisse \Rightarrow Un interpolateur conservant cette propriété est simple à trouver

¹ Near-Kernel Space : Ensemble des vecteurs propres associés aux plus petites valeurs propres

$$(\text{Problème de Helmholtz}) \Leftrightarrow \begin{cases} -\Delta u - k^2 u = f \text{ sur } \Omega = [0, 1] \\ u|_{\partial\Omega} = 0 \end{cases} \quad (4)$$

En appliquant un schéma de différence finie de 2nd ordre sur une discrétisation uniforme de Ω en 1D:

$$Au = f \quad \text{with} \quad A = \frac{1}{h^2} \text{Tridiag}(-1, 2, -1) - k^2 I \quad (5)$$

En appliquant une analyse locale de Fourier A :

$$\forall j = 1, \dots, n; \quad \lambda_j(A) = \frac{2 - 2 \cos(j\pi h)}{h^2} - k^2, \quad v_j(A) = [\sin(lj\pi h)]_{l=1}^n$$

Les valeurs propres sont décalées !

- Les $\lambda_j(A)$ sont positives \Rightarrow Méthodes de Krylov, Lissage sur équations normales
- Le NKS¹ est lisse \Rightarrow Quel interpolateur choisir pour propager efficacement le noyau?

¹ Near-Kernel Space : Ensemble des vecteurs propres associés aux plus petites valeurs propres

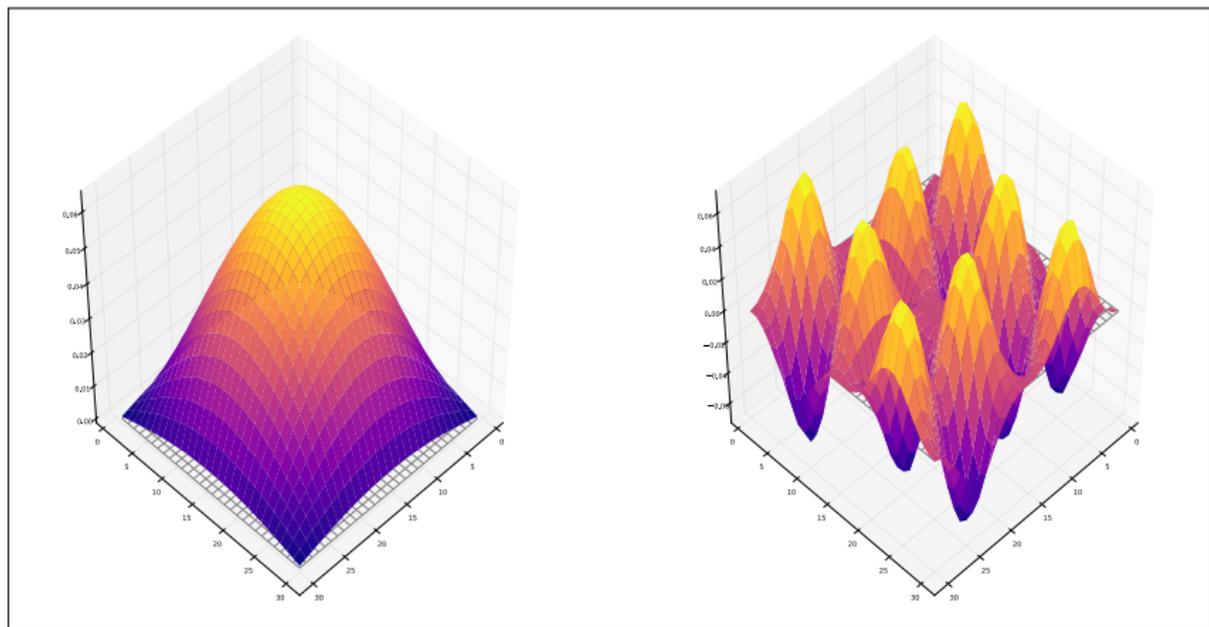


Figure 4: Illustration des noyaux - Laplace ($k = 0$ - lisse) vs. Helmholtz ($k \neq 0$ - oscillant)

- ① Les méthodes multigrilles
 - 1.1 - Principe général de la méthode
 - 1.2 - Application au cas Laplace
 - 1.3 - Application au cas Helmholtz
- ② Opérateur d'interpolation adapté au cas Helmholtz
 - 2.1 - Découpage \mathcal{C}/\mathcal{F}
 - 2.2 - Définition de l'interpolateur idéal
 - 2.3 - Approximation de l'interpolateur idéal
 - 2.4 - Définition de règles d'interpolation initiales à l'aide d'information spectrale locale
 - 2.5 - Élimination des fréquences résiduelles par une matrice de lissage
- ③ Conclusion
 - 3.1 - Benchmark
 - 3.2 - Perspectives

La sélection des variables grossières est préalable à la construction de l'interpolateur.

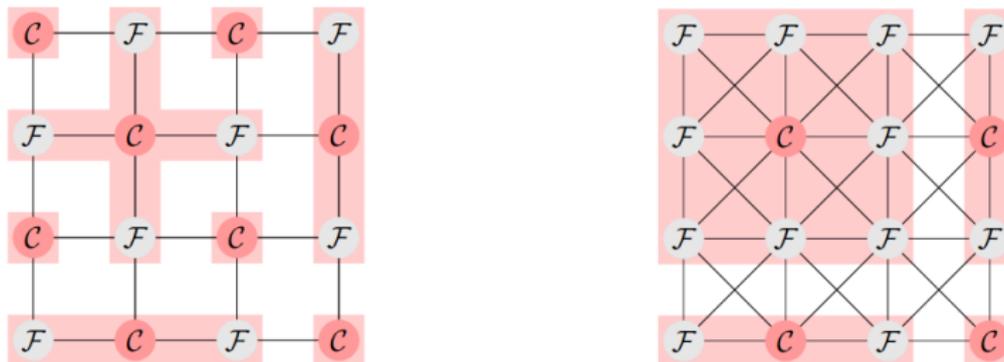


Figure 5: Exemples de découpage \mathcal{C}/\mathcal{F} pour des stencils à 5 et 9 points

Quelques techniques :

- Sélection basée sur la géométrie du problème
- Sélection algébrique en utilisant exclusivement la matrice
- Sélection par méthode d'optimisation (recuit simulé, etc...)

Soit S et R^T deux interpolateurs d'injection tels que :

$$\Omega \xrightarrow{S^T} \mathcal{F}, \quad \Omega \xrightarrow{R} \mathcal{C}, \quad \text{avec } \Omega = \mathcal{C} \cup \mathcal{F} \quad (6)$$

La littérature [3] définit un opérateur *idéal* P^* afin d'étudier la convergence optimale d'une méthode à 2 niveaux étant donné un découpage \mathcal{C}/\mathcal{F} , tel que

$$P^* = (I - S(S^T A S)^{-1} S^T A) R^T$$

Après renumérotation par bloc selon le découpage (6)

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}, \quad \text{and } S = \begin{bmatrix} I_{ff} \\ 0 \end{bmatrix}, \quad R^T = \begin{bmatrix} 0 \\ I_{cc} \end{bmatrix}$$

$$\text{Alors } P^* = \begin{bmatrix} -A_{ff}^{-1} A_{fc} \\ I_{cc} \end{bmatrix} \text{ et } A_c = P^{*T} A P^* = \underbrace{A_{cc} - A_{cf} A_{ff}^{-1} A_{fc}}_{\text{Complément de Schur}} \neq R A R^T = A_{cc}$$

P^* est appelé idéal car il retire l'information fine de la représentation grossière

$$\Rightarrow P = (I - S(S^T A S)^{-1} S^T A) \begin{bmatrix} W_{fc} \\ I_{cc} \end{bmatrix} = P^* \quad (7)$$

Problème : P^* nécessite une inversion exacte (trop couteuse et trop dense)

Il est cependant possible d'en faire une approximation :

- Inversion de blocs diagonaux disjoints
- Approximation des connexions courtes entre blocs par compléments de Schur

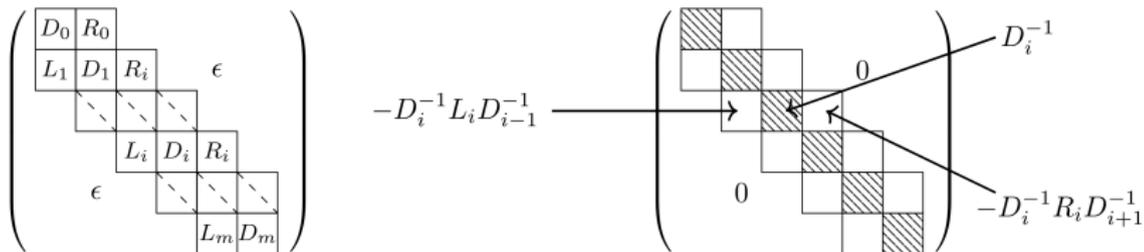


Figure 6: Approximation inverse de $S^T A S$ par compléments de Schur

$$P = \underbrace{(I - S(S^T A S)^{-1} S^T A)}_{\mathcal{F}\text{-Relaxation}} R^T \quad (8)$$

Puisque l'inversion n'est plus exacte,

$$P = (I - S(S^T A S)^{-1} S^T A) \begin{bmatrix} W_{fc} \\ I_{cc} \end{bmatrix} \neq P^* \quad (9)$$

Comment déterminer de bonnes règles d'interpolation initiales permettant de conserver une bonne représentation du noyau à travers les niveaux ?

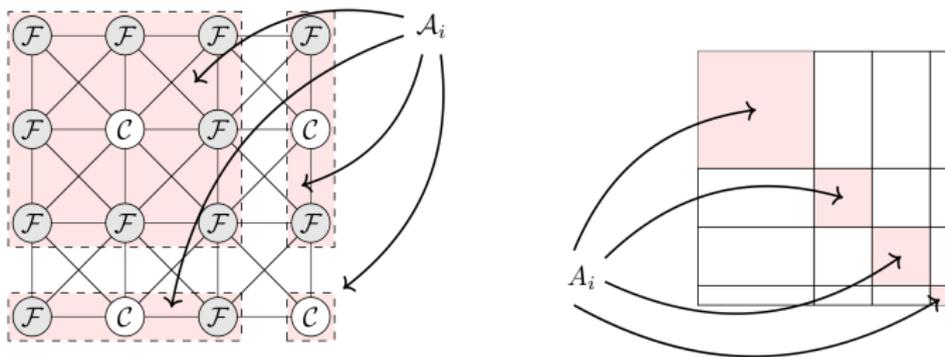


Figure 7: Illustration du processus de création d'agglomérats

→ Possibilité de calculer les plus petits vecteurs propres $v_0(A_i)$ à partir de sous-blocs A_i .

Deux méthodes peuvent être utilisées pour calculer un bon interpolateur initial \mathcal{T} :

- Minimisation de la différence [1] entre l'interpolation des variables \mathcal{F} et leurs valeurs effectives dans les composants locaux $v_0(A_i)$ extraits des agglomérats

$$\forall i \in \mathcal{F}, t_{i,:} = \arg \min_{t_i} \sum_{k=1}^{\kappa} w^{(k)} (v_i^{(k)} - t_i v_C^{(k)})^2 = v_i W V_C^H (V_C W V_C^H)^{-1} \quad (10)$$

- Agglomérats centrés sur les variables \mathcal{F} et recouvrements possibles
- κ doit être au moins égal au nombre de variables \mathcal{C} connecté à la variable \mathcal{F} ciblée
- Calcul des colonnes disjointes par réflecteurs de Householder [2]

$$\forall i \in \mathcal{C}, t_{:,i} = Q_{:,i} \text{ tq } Q^T v_0(A_i) = \|v_0(A_i)\|_2 \cdot \vec{u}_i \Leftrightarrow v_0(A_i) = Q \|v_0(A_i)\|_2 \cdot \vec{u}_i \quad (11)$$

- Agglomérats centrés sur les variables \mathcal{C} et disjointes
- Un seul vecteur propre peut suffire

A ce stade :

$$P = (I - S(S^T A S)^{-1} S^T A) \mathcal{T} \quad (12)$$

Dans la littérature [2], l'opérateur \mathcal{T} est couplé à une matrice de lissage pour :

- Étendre le pattern d'éléments non-nuls
- Éliminer les hautes fréquences puisqu'elles sont indésirables pour cibler le noyau

Ce qui donne finalement :

$$P = (I - wD^{-1}A)(I - S(S^TAS)^{-1}S^T A)\mathcal{T} \quad (13)$$

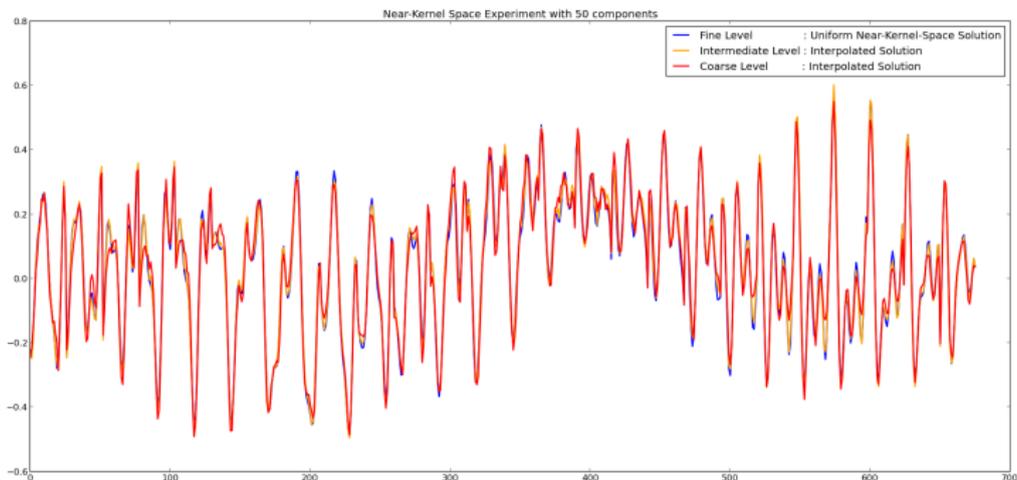


Figure 8: $x =$ Somme des composants proches du noyau vs. interpolations des x_C

1 Les méthodes multigrilles

- 1.1 - Principe général de la méthode
- 1.2 - Application au cas Laplace
- 1.3 - Application au cas Helmholtz

2 Opérateur d'interpolation adapté au cas Helmholtz

- 2.1 - Découpage \mathcal{C}/\mathcal{F}
- 2.2 - Définition de l'interpolateur idéal
- 2.3 - Approximation de l'interpolateur idéal
- 2.4 - Définition de règles d'interpolation initiales à l'aide d'information spectrale locale
- 2.5 - Élimination des fréquences résiduelles par une matrice de lissage

3 Conclusion

- 3.1 - Benchmark
- 3.2 - Perspectives

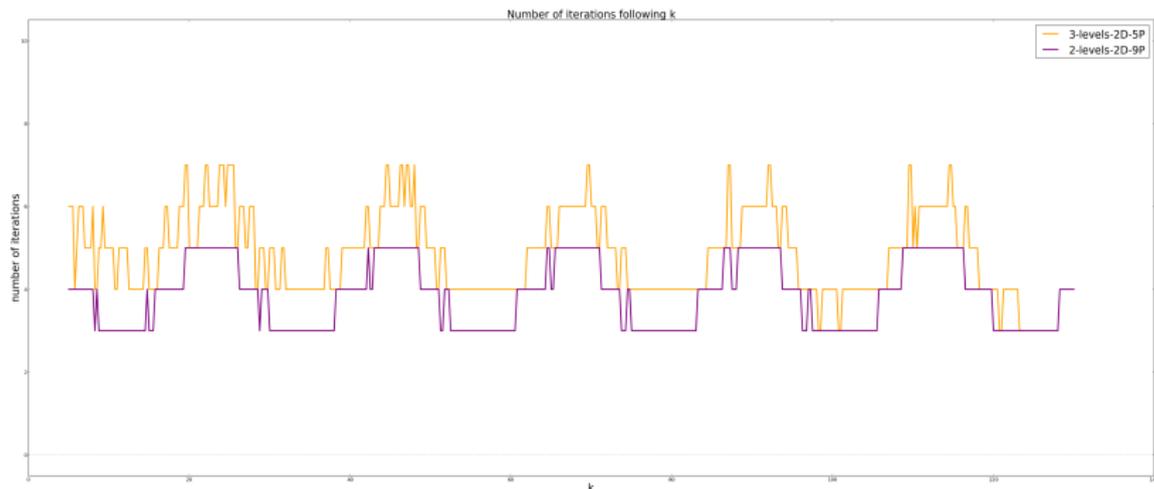


Figure 9: Tests en 2D sur des stencils à 5 & 9 points - $kh = 0.625$

Pour $k = 123$, $n = 38416$:

- 2D - 5 Points : Réduction de taille par 8 $\rightarrow n_C = 4850$.
- 2D - 9 Points : Réduction de taille par 4 $\rightarrow n_C = 9604$.

- Remplacer la méthode de lissage utilisée dans P pour rendre le cycle plus profond
 - Méthodes de Krylov - Plus adaptées aux matrices indéfinies
 - Ajout de contraintes - Permet de conserver un pattern d'éléments non-nuls
- Premiers bons résultats en 3D (réduction de taille par 15) \Rightarrow paralléliser le code pour augmenter k
 - Pour $k = 29$, $n = 103823 \rightarrow 6$ itérations, $n_C = 6895$.
 - Pour $k = 36$, $n = 195112 \rightarrow 7$ itérations, $n_C = 12652$.
- Comparer avec d'autres méthodes existantes
 - *Complex-Shifted Laplacian*
 - *Multiple coarsening*

- [1] A. Brandt et al. "Bootstrap AMG". English (US). In: [SIAM Journal of Scientific Computing](#) 33.2 (2011), pp. 612–632. ISSN: 1064-8275. DOI: 10.1137/090752973.
- [2] Petr Ek, Marian Brezina, and Jan Mandel. "Convergence Of Algebraic Multigrid Based On Smoothed Aggregation". In: [Computing](#) 56 (May 1998). DOI: 10.1007/s002110000226.
- [3] Robert D. Falgout and Panayot S. Vassilevski. "On Generalizing the AMG Framework". In: [SIAM J. NUMER. ANAL](#) 42 (2003), pp. 1669–1693.

Merci pour votre attention !