Multigrid methods applied to the Helmholtz equation

Clément Richefort

May 5, 2022

Table of contents

1 - Introduction

Principle of multigrid methods :

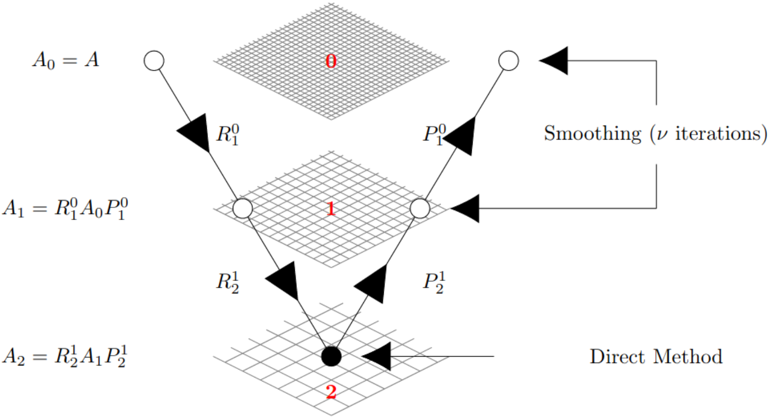
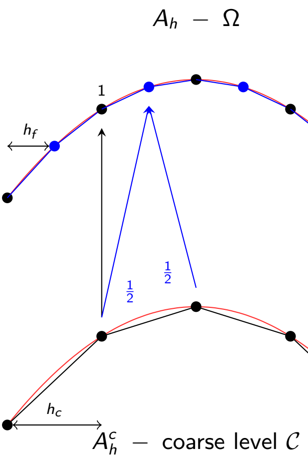


Figure: Illustration of 3 levels V-cycle multigrid method

1-Introduction

Most trivial interpolator :



Interpolation from coarse space to original space is made by

$$P_c^{\Omega} = \frac{1}{2}\begin{bmatrix} 1 & \dots & 0 \\ 2 & & \vdots \\ 1 & 1 & \\ 0 & 2 & \ddots \\ \vdots & 1 & \ddots \\ 0 & \dots & \ddots \end{bmatrix}$$

1 - Introduction

$$\text{(Indefinite Helmholtz Problem)} \quad \Leftrightarrow \quad \begin{cases} -\Delta u \; - k^2 u = f \text{ on } \Omega \\ \partial_{n_j} u = iku \text{ on } \partial\Omega \end{cases}$$

The discretization of Ω leads to an indefinite system $Au = f$, involving two major issues for multigrid methods :

- Eigenvalues are both signed $\Rightarrow$ Problematic for smoothing steps
- Oscillatory near kernel space $\Rightarrow$ Hard to make appropriate interpolators

Target : Find smoothers and interpolators making multigrid methods converging in a constant number of iterations and independently of k.

1. Smoother : Normal equations methods or Krylov iterations
2. Interpolator : **Still an open question ...**

1 - Introduction

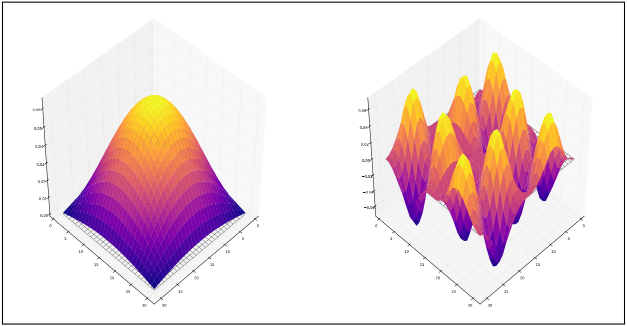


Figure: Laplace ($k = 0$ - *smooth*) / Helmholtz ($k \neq 0$ - *wave-like*) near-kernels

Reminder : Near-kernel space is defined by the set of eigenvectors associated to smallest absolute eigenvalues

$\rightarrow$ These eigenvectors are the most important ! $A^{-1}b = \sum_i \frac{\alpha_i}{\lambda_i} v_i$

2 - Approximating the Ideal Interpolator

Let A a $n \times n$ matrix where range$(A) = \mathbb{R}^n$, x and b respectively solution and right hand side of the system $Ax = b$.

Ideal interpolator P^* is mostly used for theoretical purpose, and permits to give information on the convergence scenario following a given $\mathcal{C}(coarse)/\mathcal{F}(fine)$-splitting.

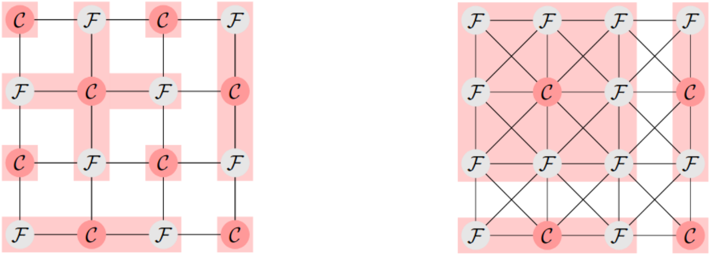


Figure: 5 and 9 points Stencil $\mathcal{C}/\mathcal{F}-$Splittings

2 - Approximating the Ideal Interpolator

Let S and R^T both injection interpolators such that

$$\mathcal{C} \cup \mathcal{F} \overset{S^T}{\longmapsto} \mathcal{F} \, , \, \mathcal{C} \cup \mathcal{F} \overset{R}{\mapsto} \mathcal{C}$$

In the literature, the Ideal Interpolator is defined the following way

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

Let reorganize A such that

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

Thus $P^* = \begin{bmatrix} -A_{ff}^{-1} A_{fc} \\ I_{cc} \end{bmatrix}$ and $A_c = {P^*}^T A P^* = \underbrace{A_{cc} - A_{cf} A_{ff}^{-1} A_{fc}}_{\text{Schur Complements formula}} \neq RAR^T = A_{cc}$

P^* removes the whole fine related information from the coarse space representation !

2 - Approximating the Ideal Interpolator

Problem : P^* contains an exact inversion, which is too expensive. Plus, if A_c is dense, it will limit our capacity to coarsen deeper

But it is still possible to approximate $(S^TAS)^{-1}$ by $(S^TAS)^{\sim 1}$!

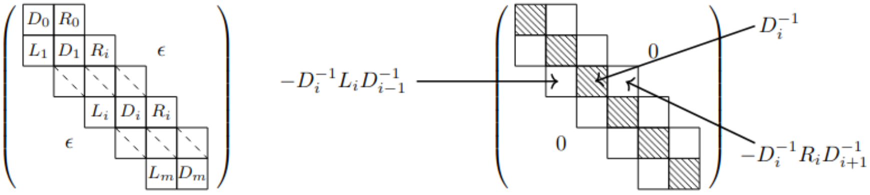


Figure: Approximation of $(S^TAS)^{-1}$ using Schur complements

Then we define approximation of Ideal Interpolator as

$$P = \underbrace{(I - S(S^TAS)^{\sim 1}S^TA)}_{\mathcal{F}-\text{Relaxation}} R^T$$

2 - Approximating the Ideal Interpolator

Let $\mathcal{E} = (S^T AS)^{-1} - (S^T AS)^{\sim 1} = A_{ff}^{-1} - A_{ff}^{\sim 1}$. Thus it follows that

$$P = \begin{bmatrix} -A_{ff}^{-1} A_{fc} + \mathcal{E} A_{fc} \\ I_{cc} \end{bmatrix} = \underbrace{P^*}_{\text{Ideal Interpolator}} + \underbrace{\begin{bmatrix} \mathcal{E} A_{fc} \\ 0 \end{bmatrix}}_{Noise}$$

<u>Goal :</u> Find a good trade-off between sparsity and noise reduction !

How to remove the noise which degrades the coarse representation of the near-kernel space?

<u>Idea :</u> Adding a correction matrix to the ideal approximation formula !

$$P = (I - X^{-1}A)(I - S(S^T AS)^{\sim 1} S^T A)R^T$$

<u>Remark :</u> We need to keep in mind that P should be the sparsest possible, and also that M should not damage the near-kernel space while removing the noise. $(X^{-1} = wD^{-1})$

3 - Tentative interpolator built from local NKC

The following idea is inspired by the Smoothed Aggregation method.

Here is its principle :

1. Let a system $Ax = b$ with a known solution (for instance $Ax = 0$).
2. Approximate x by $\tilde{x}$ with few smoothing iterations, then compute $e = x - \tilde{x}$.
3. Construct a *Tentative interpolator* $\mathcal{T}$ such that $e = \mathcal{T} e_c \Leftrightarrow \mathcal{T}^T e = e_c$.
4. Then compute $P = M\mathcal{T}$ with M some error propagation matrix.

$\Rightarrow$ P targets remaining information e that smoother is not able to capture.

3 - Tentative interpolator built from local NKC

The previous operator : $P = (I - X^{-1}A)(I - S(S^TAS)^{\sim 1}S^TA)R^T$ with

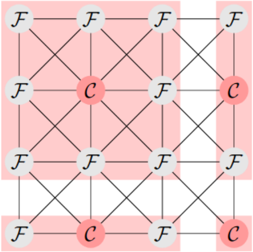
$$\underbrace{\begin{bmatrix}1\\0\\1\\0\\1\\0\end{bmatrix}}_{e} = R^T \underbrace{\begin{bmatrix}1\\1\\1\end{bmatrix}}_{e_c} \Leftrightarrow R \begin{bmatrix}1\\0\\1\\0\\1\\0\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

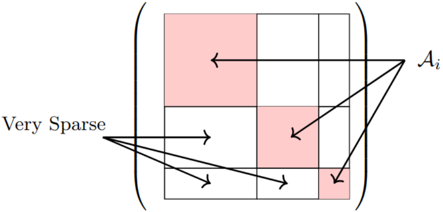
Would a $\mathcal{T}$ satisfying some $e = \mathcal{T}e_c \Leftrightarrow \mathcal{T}^Te = e_c$ better than R^T?
What if e contains near-kernel information?

3 - Tentative interpolator built from local NKC

How to construct $\mathcal{T}$?

- 1. From a given $\mathcal{C}/\mathcal{F}$ splitting, divide Ω in $\mathcal{A}_i$ agglomerates and compute its lowest component $v_0(\mathcal{A}_i)$.





3 - Tentative interpolator built from local NKC

How to construct $\mathcal{T}$?

- 2. For each agglomerate $\mathcal{A}_i$, compute the Householder Matrix Q_i such that $Q_i^T v_0(\mathcal{A}_i) = \|v_0(\mathcal{A}_i)_i\|_2 u_i^{(\mathcal{C})}$ with $u_i^{(\mathcal{C})}$ canonical vector of axis $(\mathcal{C})$. Since $Q_i^T v_0(\mathcal{A}_i)$ is null on each $(\mathcal{F})$ elements, keep only column $(\mathcal{C})$ of Q_i.

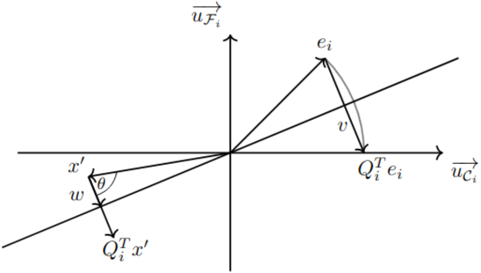


Figure: Householder reflection

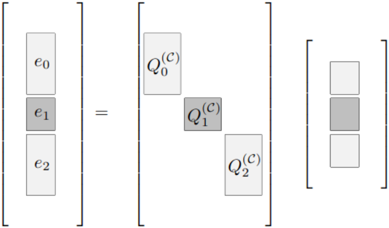
$$Q_i^T = \left(I - 2\frac{v_0(\mathcal{A}_i)v_0(\mathcal{A}_i)^T}{\|v_0(\mathcal{A}_i)\|_2^2}\right)$$

and

$$Q_i^T v_0(\mathcal{A}_i) = \|v_0(\mathcal{A}_i)\|_2 u_i^{(\mathcal{C})}$$
$$\Leftrightarrow$$
$$v_0(\mathcal{A}_i) = Q_i \|v_0(\mathcal{A}_i)\|_2 u_i^{(\mathcal{C})}$$

3 - Tentative interpolator built from local NKC

- 3. Repeat the process for each $\mathcal{A}_i$, and build the block column matrix $\mathcal{T}$



3 - Tentative interpolator built from local NKC

To summarize the construction of $\mathcal{T}$:

Algorithm 1 Tentative Prolongator with local lowest components

$\mathcal{C}/\mathcal{F} \leftarrow \text{ComputeCFSplitting}(A)$
$\mathcal{A} \leftarrow \text{ComputeAgglomerates}(A, \mathcal{C}, \mathcal{F})$
for $i \leq \text{card}(\mathcal{C})$ **do** :
    $v_0(\mathcal{A}_i) \leftarrow \text{ComputeLowestEigenVector}(\mathcal{A}_i)$
    $Q_i \leftarrow \text{ComputeHouseholderReflector}(v_0(\mathcal{A}_i))$
    $\mathcal{T} \leftarrow \text{InsertAsNewColumn}(Q_i)$
end for
Return $\mathcal{T}$

$$\boxed{P = (I - wD^{-1}A)(I - S(S^T AS)^{\sim 1} S^T A)\mathcal{T}}$$

4 - Benchmarks

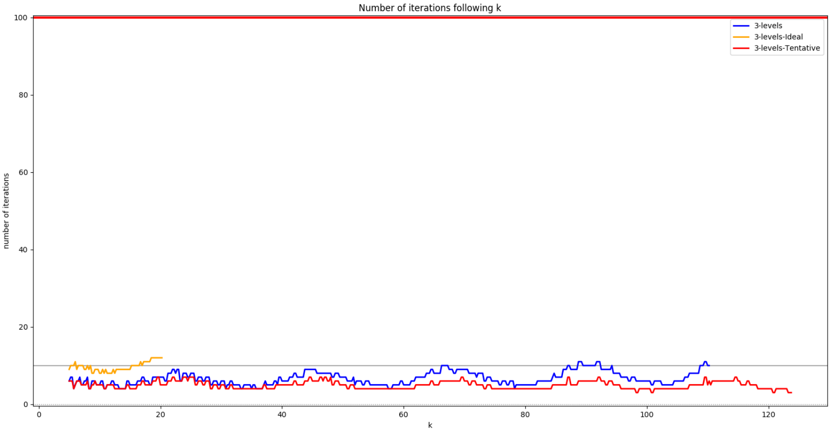


Figure: Ideal vs. Smoothed Ideal Block Approximation - 5P Stencil

4 - Benchmarks

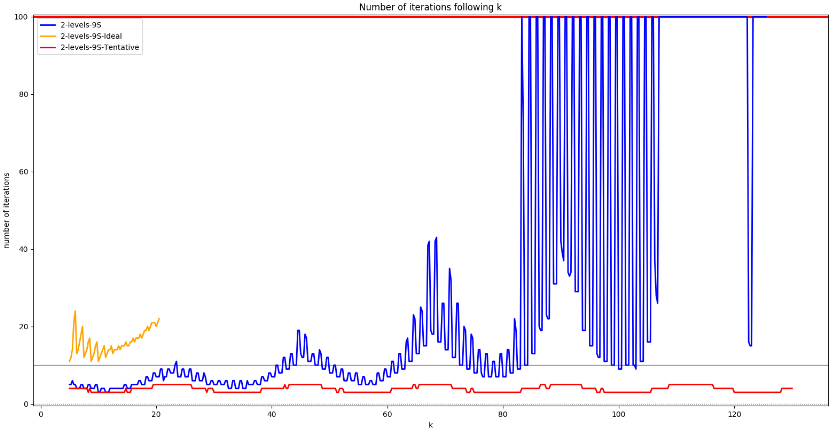


Figure: Ideal vs. Smoothed Ideal Block Approximation - 9P Stencil

5 - To do next

1. Use Conjugate Gradient on Normal Equations (CGNR) instead of w-Jacobi as smoothing matrix
2. Add constraint in CGNR sub-research space to keep interesting properties in coarse matrices in order to coarsen deeper. (structure, clean near-kernel space, etc.)