Student Paper Competition, Copper Mountain Conference 2023 - Toward a multilevel method for the Helmholtz equation

Clément Richefort, Matthieu Lecouvez, Robert Falgout, Pierre Ramet

TOWARD A MULTILEVEL METHOD FOR THE HELMHOLTZ EQUATION *

CLÉMENT RICHEFORT †

In collaboration with: Matthieu Lecouvez, Rob Falgout, Pierre Ramet

Abstract. It is well known that multigrid methods are very competitive in solving a wide range of SPD problems. However achieving such performance for non-SPD matrices remains an open problem. In particular, two main issues may arise when solving a Helmholtz problem. Some eigenvalues become negative or even complex, requiring the choice of an adapted smoothing method for capturing them. Moreover, since the near-kernel space is oscillatory, the geometric smoothness assumption cannot be used to build efficient interpolation rules. We present some investigations about designing a method that converges in a constant number of iterations with respect to the wavenumber. The method builds on an ideal reduction-based framework and related theory for SPD matrices to correct an initial least squares minimization coarse selection operator formed from a set of smoothed random vectors. We also present numerical results at the end of the paper.

Key words. Multigrid, Helmholtz, Linear Algebra, Polynomial Smoother

1. Introduction.

The numerical simulation of various physical phenomena gives rise to potentially very large linear systems of equations written $Au = b$ in matrix form. These systems can be solved directly by a convenient factorization of A, or iteratively by computing and refining an approximation of the solution u starting from an initial guess u_0. Multigrid methods [15, 8] work iteratively and are known to be scalable and quasi-optimal for solving sparse linear systems of equations for many classes of problems. To simplify the discussion in what follows, we use the term "small/large eigenvector" to mean an eigenvector with small/large eigenvalue. We similarly say "positive/negative eigenvector" when referring to the eigenvalue sign.

1.1. General aspects on multigrid methods.

The core idea in multigrid methods is to accelerate the computation of u by way of a hierarchy of coarse problems $A_l u_l = b_l$, l being the level in the grid hierarchy. A restriction operator R_l transfers the information from a level l to a coarser one $l+1$, while an interpolation operator P_l transfers the information from level $l+1$ to l. In most symmetric applications, $R_l = P_l^T$ and coarse matrices are constructed following the Galerkin formula $A_{l+1} = P_l^T A_l P_l$. Two-grid methods actually need both types of solvers: a direct method for the coarse correction, and an iterative method called a smoother on the fine level. The error propagation matrix for the coarse correction is

$$E = I - P(P^T A P)^{-1} P^T A. \tag{1.1}$$

The error propagation matrix for the smoother is

$$E_M = I - M^{-1} A \tag{1.2}$$

where M^{-1} is an approximation of A^{-1}, for instance the diagonal inverse or a polynomial of A. The smoother is applied before each restriction and after each interpolation. Finding a smoother and a coarse correction that are complementary is a

*This work was funded by CEA.

†Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA), (richefort.clement@protonmail.com).

major concern in the design of the method. The interpolator must propagate the coarsest information back to the finest, and transferred errors should be eliminated by the smoother. The smoother targets the large eigenvectors and the coarse correction targets the small eigenvectors. The near-kernel space of smallest eigenvectors is especially important in the design of interpolation. A multi-level method can be created by recursively applying the two-level method to solve the coarse system, where a direct solver is used on the coarsest level. The context in which a multigrid method is applied determines what kind of operators should be used in the method. In elliptic problems, where the convergence of multigrid methods is well known, the matrix A is symmetric positive definite, so smoothers like weighted-Jacobi or Gauss-Seidel are known to be good smoothers since they damp the high frequencies without modifying the low frequency eigenvectors. Likewise, interpolators are designed to target slowly varying components as in classic algebraic multigrid methods (AMG) [16].

1.2. Why Helmholtz problems are difficult for multigrid. The Helmholtz problem involves indefinite matrices with potentially wide and oscillatory near-kernel space [7]. This complication breaks the near-kernel space geometric smoothness assumption, a keystone of many multigrid methods. To satisfy the complementarity principle in this context, interpolation rules must reproduce the near-kernel oscillation, and smoothers have to deal with both positive and negative eigenvalues. More importantly, finding a recurring process to build a scalable multilevel method is still an open question. The Helmholtz equation (1.3) is our target in this paper.

$$\text{(Continuous Helmholtz problem)} \Leftrightarrow \begin{cases} -\Delta u - k^2 u = f & \text{on } \Omega \\ \quad + \text{b. c.} & \text{on } \partial\Omega \end{cases} \tag{1.3}$$

Since the Helmholtz equation can be seen as a shifted Poisson equation, the geometrically smooth components (ie. low Fourier modes) become negative. Because of the shift, the smallest eigenvectors are higher in frequency. Multigrid interpolators must now focus on this more oscillatory spectrum interval. Multiple correction [12] and wave-ray [4, 11] approaches have already been investigated to address this issue. In this paper, we present an approach built on ideal reduction-based ideas, and demonstrate its potential for solving the Helmholtz problem in constant iteration count independent of the wavenumber k. In Section 2, we present a normal equation polynomial smoother specifically designed to damp the desired proportion of highest amplitude eigenvalues, while interpolation rules for propagating oscillatory near-kernel information are established in Section 3. Finally, Section 4 contains benchmarks of this new multigrid cycle for different Helmholtz problems, with a varying wavenumber k.

2. Polynomial Smoothers for Indefinite Problem. Working with a smoothing method whose behavior on each portion of the spectrum is *a priori* known is interesting to guarantee the effectiveness of the cycle. Here, the smoother must drop large positive and negative eigenvalues, which is problematic for most standard methods. Generally, a polynomial method with degree greater than one can work. Krylov iterations are good polynomial smoother in the indefinite case but they minimize the global residual norm regardless of the eigenvalues and are non-linear because of their right-hand side dependence. A linear polynomial is more convenient for generating the set of smoothed candidates vectors needed to construct the interpolation operator described in Section 3.

2.1. General considerations on polynomial smoothers. One way to ensure that both positive and negative eigenvectors are damped is to consider a normal

equation polynomial smoother. In general, the degree d of the polynomial must be greater than 1 to damp positive and negative eigenvectors, as the polynomial illustrated in Figure 2.1 does. This condition is guaranteed here since the normal equations lead to an even polynomial degree. In the future, it might be interesting to investigate more general polynomials to avoid normal equations and consider odd degrees. In this first approach, we use the convenient symmetric property enabled by normal equations in the Chebyshev framework. For any positive x and given some positive spectrum interval $\mathcal{I} := [x_{\min}, x_{\max}]$, we look for a polynomial $p(x)$ of degree d maximizing the damping of all components lying in $\mathcal{I}$. In that direction, and according to Equation (1.2), let

$$q(A^H A) := I - p(A^H A)A^H A \tag{2.1}$$

be the associated error propagation operator. Then, for any right singular vector v of A and σ the associated singular value,

$$q(A^H A)v = q(\sigma^2)v = (1 - p(\sigma^2)\sigma^2)v \tag{2.2}$$

According to Equation (2.2), the lower the amplitude of a component, the less the polynomial smoother will damp it. This intact portion of low components enables the construction of a set of smoothed vectors approximating the near-kernel space, as detailed in Section 3. Let L be a set of points lying in this interval such that

$$\forall\, x_i \in L = (x_1, \ldots, x_d)\ ,\ x_i \neq 0\ ,\ q(x_i) = 0 \ \Leftrightarrow\ p(x_i) = \frac{1}{x_i}. \tag{2.3}$$

Then the polynomial smoother p is constructed following the Lagrangian formula

$$x_i \neq x_j\ ,\ p(x) = \sum_{j=0}^{d} \frac{1}{x_j} \prod_{i=0, i\neq j}^{d} \frac{x - x_i}{x_j - x_i}. \tag{2.4}$$

However, such interpolation points should not be selected randomly within the interval, but in order to minimize the polynomial amplitude and avoid Runge's phenomenon. Those best interpolating points are defined by the scaled first kind Chebyshev polynomial roots

$$x_i := \frac{x_{\max} + x_{\min}}{2} + \frac{x_{\max} - x_{\min}}{2} \cos\left(\frac{(2i-1)\pi}{2d}\right). \tag{2.5}$$

Large intervals $\mathcal{I}$ require a higher polynomial degree to flatten its oscillations.

2.2. Constructing an appropriate target interval $\mathcal{I}$. One way to determine a good interval without preliminary information [2, 1] is to compute a few power iterations to determine $x_{\max}$, overestimate the result by 10%, and choose the lower bound $x_{\min}$ according to $x_{\max}$, for example $x_{\min} = 0.5x_{\max}$. To respect the complementarity principle, the percentage of damped eigenvalues by the smoother must approximate the proportion of non-coarse variables. For instance, if a coarse level space is defined by a selection of a quarter of the finer level's degrees of freedom, then three-quarters of the largest amplitude components should be damped by smoothing steps, while the coarse correction deals with the other part. Consequently, since eigenvalues are not necessarily uniformly separated, $x_{\min}$ should be determined so that such a proportion of eigenvalues belongs to the interval $\mathcal{I}$. In this paper and following [10], we

first compute a rough approximation of the matrix *spectral density* defined by the distribution function $\phi(t)$, which represents the probability of finding an eigenvalue at each point t of a given interval. We set the lower bound $x_{\min}$ of the Chebyshev nodes interval in a second step so that the probability within the interval is equal to the target proportion, for instance half of the total area in a scenario of exact balance between $\mathcal{C}$ and $\mathcal{F}$ points. The squared singular values should be scaled by setting $B := \frac{2}{x_{\max}} A^H A - I$, so the distribution function ϕ can be approximated by a linear combination of orthogonal Chebyshev polynomial functions

$$\phi(t) = \sum_{k=0}^{\infty} \mu_k T_k(t) \approx \sum_{k=0}^{M} \mu_k T_k(t) \ , \ \mu_k = \frac{2 - \delta_{k0}}{n\pi} \times \mathrm{Trace}(T_k(B)) \tag{2.6}$$

where $T_k(t) = \cos(k \arccos(t))$ and the coefficients μ_k are determined by a moments matching procedure. Here, n corresponds to the matrix size and δ_{k0} the Kronecker symbol. The trace can be estimated using a set U of n_{vec} random and orthogonal vectors u, where each element of these vectors is chosen following a normal distribution with zero mean and a unit standard deviation. Any vector u can be written as a linear combination of B eigenvectors v, giving the expression $u = \sum_{j=1}^{n} \beta_j v_j$. As a consequence, $\mathbb{E}[u] = 0$ and $\mathbb{E}[\beta_i \beta_j] = \delta_{ij}$, giving the following convenient property

$$\mathbb{E}[u^T T_k(B) u] = \mathbb{E}[\sum_{j=1}^{n} \beta_j^2 T_k(\lambda_j(B))] = \sum_{j=1}^{n} T_k(\lambda_j(B)) = \mathrm{Trace}(T_k(B)). \tag{2.7}$$

According to (2.7), each trace can be estimated by a sample mean of n_{vec} products $u^T T_k(B) u$, and the M vectors $T_k(B)u$ can be computed from the three-term recurrence relation $T_{k+1}(B)u = 2BT_k(B)u - T_{k-1}(B)u$, with initial terms $T_0(B)u = u$, $T_1(B)u = Bu$. Once the distribution function ϕ is approximated following the expression given in (2.6), a rough area approximation by trapezoid rule yields a correct lower bound x satisfying a proportion around $\mathrm{card}(\mathcal{F})/n$. Computing $x_{\min}$ by remapping x to the initial squared singular values scaling leads to a satisfying and purely algebraic interval in which our polynomial smoother will be the most efficient. The bounds $x_{\min}$ and $x_{\max}$ are represented in Figure 2.1, where $x_{50\%}$ represents the theoretical lower bound targeted.

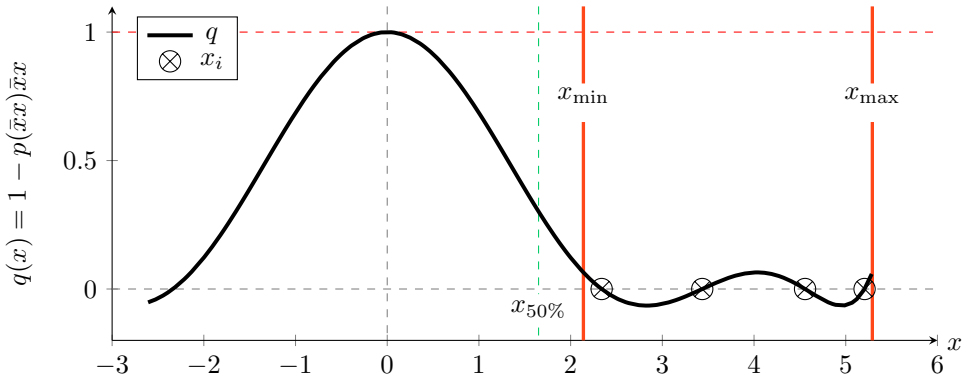

FIG. 2.1. *Spectrum of the polynomial smoother error propagation matrix*

3. Constructing good interpolation rules. Interpolators are used both to construct the coarse level matrices and to transfer information across levels. SPD and geometric smoothness assumptions cannot be used to determine appropriate interpolators in our case. Some methods such as smoothed aggregation [6, 13] and bootstrap-AMG [3] use candidate vectors that are close to the near-kernel space to design the interpolation rules. These test vectors are either deduced from geometric information [4] or algebraically as in adaptive multigrid methods [5]. Here, we prefer to stick to a fully algebraic and recurring process to create our interpolators. Candidate vectors will be generated from random vectors smoothed by the polynomial presented in Section 2, and used by the least squares minimization framework to determine good fine variable interpolation rules. This initial least squares interpolator is used as a coarse selection operator in the ideal reduction-based framework [9].

3.1. Ideal framework. Even though the ideal framework in [9] requires an SPD assumption and has not been generalized to indefinite problems, removing orthogonal information from the interpolation range still improves its accuracy. Furthermore, assuming the smoother captures this orthogonal information, we can still expect good convergence as exposed by Equation (3.3) below. Following [9], let $\mathcal{C}$ and $\mathcal{F}$ be complementary coarse and fine variables subsets of Ω. Let $R^T : \mathbb{R}^{n_\mathcal{C}} \to \mathbb{R}^n$ and $S : \mathbb{R}^{n_\mathcal{F}} \to \mathbb{R}^n$ be coarse and fine selection operators respectively, such that $RS = 0$, for instance, the orthogonal matrices

$$R^T = [\ 0\ \ I_\mathcal{C}\ ]^T\ ,\ S = [\ I_\mathcal{F}\ \ 0\ ]^T. \tag{3.1}$$

The space defined by the coarse selection operator R^T must be handled by the coarse correction, whereas the fine variables selection operator S defines a space where smoothing must operate in order to respect the complementarity principle. The *Ideal Interpolator* is a theoretical operator that is the best interpolator satisfying $RP = I_\mathcal{C}$ in the sense that it minimizes the difference between variables and interpolated coarse variables, within a space that is the most complementary to the range of the smoother. For this reason, P_* is called *ideal* and is defined by

$$P_* = (I - S(S^TAS)^{-1}S^TA)R^T. \tag{3.2}$$

The left operator in (3.2) removes all complementary $\mathcal{F}$-related information from R^T. Such information is irrelevant at a coarse level and should be handled by the smoother. Under the assumption that the smoother captures $\mathcal{F}$-related information, the best coarse matrix, according to the complementarity principle, is a matrix where fine variable information is removed. Furthermore, a simple development of a two-grid cycle combining the ideal coarse error propagation operator E_*, as initially defined in (1.1), with an $\mathcal{F}$-relaxation matrix $E_\mathcal{F}$ as a pre-smoother shows that

$$E_*E_\mathcal{F} = (I - P_*(P_*^TAP_*)^{-1}P_*^TA)(I - S(S^TAS)^{-1}S^TA) = 0. \tag{3.3}$$

In (3.3), the final error propagation matrix is null, meaning that in this idealistic scenario, one iteration of $\mathcal{F}$-relaxation and one coarse correction is equivalent to a direct method. As shown in [9], we can extend this principle by separating coarse and fine spaces respectively in directions of low and high frequencies. Let $V_0 = (v_1, \ldots, v_{\text{Card}(\mathcal{C})})$ and $V_+ = (v_{\text{Card}(\mathcal{C})+1}, \ldots, v_n)$ respectively be low frequency and high frequency eigenvectors sets, assuming here that $\lambda_i \leq \lambda_{i+1}$. Then we define $R^T = V_0$ and its counterpart $S = V_+$. Naturally, since eigenvectors are orthonormal, the necessary condition $RS = 0$ is still satisfied. It finally gives

$$P_* = R^T \text{ and } A_\mathcal{C} = \text{Diag}(\lambda_0, \ldots, \lambda_{\text{Card}(\mathcal{C})}). \tag{3.4}$$

This example proposes another idealistic dichotomy enabled by P_*, maximizing the complementarity principle. The near-kernel space is solved directly at the coarsest level while the high frequencies remain in the smoothing space. Even if $(S^T AS)^{-1}$ is most of the time impossible to use in practice, it gives insight on an idealistic convergence scenario.

3.2. Least Squares Minimization Interpolator. As mentioned at the beginning of Section 3.1, demonstrating that interpolator (3.2) is ideal in the theoretical framework of [9] requires A to be symmetric positive-definite. However, the reduction viewpoint of Equation (3.3) is still valid, hence removing the orthogonal fine information from the coarse selection operator is a viable approach. Numerical experiments show that the coarse selection operator in (3.1) is not a good option for Helmholtz. Using the lowest components V_0 from Section 3.1 to guarantee the representation of the near-kernel space within the interpolation range is not practical. Instead, we construct a set of vectors approximating an oscillatory and potentially large near-kernel space by using the normal equations polynomial smoother developed in Section 2.

In this section, we present a coarse selection operator R^H constructed by a least squares minimization strategy [3]. Let K be a set of κ vectors that approximate the near-kernel space, and assume some $\mathcal{C}/\mathcal{F}$ splitting with $n_\mathcal{C}$ and $n_\mathcal{F}$ their respective size. Coarse variables are interpolated to the finer level with a simple injection rule, meaning the coarse interpolation block in $\hat{R}^H$ corresponds to a $n_\mathcal{C} \times n_\mathcal{C}$ identity matrix, while fine interpolation rules are determined by the least squares minimization method presented in this section. Let i be a fine variable and r_i the i^{th} row of $\hat{R}^H$. The idea consists of constructing each fine interpolation rule by minimizing the squared difference between fine values of the near-kernel candidate vectors and the interpolation from their connected coarse variables $\mathcal{C}_i$. Denote by $K_{:,l}$ the l^{th} test vector, $K_{i,:}$ a row vector containing the i^{th} values of each test vector, and $K_{\mathcal{C}_i,l}$ a vector containing the values in $K_{:,l}$ of the coarse variables that are connected to the i^{th} fine variable. Then

$$\forall i \in \mathcal{F}\ ,\ r_i = \arg\min_r \sum_{l=1}^{\kappa} w_l \left(K_{i,l} - r \cdot K_{\mathcal{C}_i,l}\right)^2 := \arg\min_r \mathcal{L}_i(r) \tag{3.5}$$

where w_l are scaling weights (for instance $w_l = 1/\lambda_l$ if K contains near-kernel eigenvectors). Finding the minimum of the convex loss function $\mathcal{L}_i$ is equivalent to solving

$$\nabla \mathcal{L}_i(r_i) = 0. \tag{3.6}$$

Equation (3.6) can be rewritten element-wise

$$\frac{\partial \mathcal{L}_i(r_i)}{\partial r_{ij}} = \sum_{l=1}^{\kappa} 2w_l(K_{i,l} - r_i \cdot K_{\mathcal{C}_i,l})K_{\mathcal{C}_{ij},l} = 0\ ,\ \forall j \in [1, \text{card}(C_i)]. \tag{3.7}$$

Finally, (3.7) leads to a system of linear equations to solve for each fine variable i

$$r_i K_{\mathcal{C}_i} W K_{\mathcal{C}_i}^H = K_i W K_{\mathcal{C}_i}^H \tag{3.8}$$

The matrix is full rank and the solution of Equation (3.8) is unique if we have at least $\kappa = \max_i \{\text{Card}(C_i)\}$ locally linearly independent test vectors. Even if it is statistically always the case when starting from random candidate vectors, the matrix singularity can be detected during the factorization. In that special case, a pseudo-inverse can be computed to find an optimal solution in the least squares sense.

3.3. Ideal approximation from least squares coarse selection. In Section 3.2, we presented a better coarse selection operator for Helmholtz designed by a least squares minimization strategy. Using the framework presented in 3.1, let us define new coarse and fine selection rules

$$\hat{R}^H = [\ R_{\mathcal{F}}\ \ I_{\mathcal{C}}\ ]^T\ ,\ \hat{S} = [\ I_{\mathcal{F}}\ \ -R_{\mathcal{F}}^H\ ]^T, \tag{3.9}$$

where $\hat{R}^H$ is the least squares operator presented in Section 3.2 and $R_{\mathcal{F}}$ is its fine variable interpolation block. Note that $\hat{R}\hat{S} = 0$ as required. To simplify the discussion, let $A_{\mathcal{F}} := \hat{S}^H A \hat{S}$. Beyond the necessity to find appropriate coarse and fine selection operators, another important concern is related to the inverse of $A_{\mathcal{F}}$ required in (3.2). Reorganizing the definition of the ideal interpolator,

$$\hat{P} := (I - \hat{S}A_{\mathcal{F}}^{-1}\hat{S}^H A)\hat{R}^H \approx \hat{R}^H - \hat{S}X_{\mathcal{K}}^{-1}\hat{S}^H A\hat{R}^H, \tag{3.10}$$

where $X_{\mathcal{K}}^{-1}$ is the best polynomial approximating $A_{\mathcal{F}}^{-1}$ within the Krylov subspace $\mathcal{K}$. Since we need to ensure our interpolator keeps good sparsity, the subspace must be constrained according to a given pattern $\mathcal{P}$ as detailed in [14]. In our case, the matrix inversion approximation will be computed column-wise, giving more flexibility than by computing the global constrained matrix approximation at once with a single polynomial. Consequently, let $\mathcal{P}_i$ be some vector sparsity pattern, and define the associated operator $Z_i : \mathbb{C}^n \to \mathbb{C}^{\mathrm{Card}(\mathcal{P}_i)}$ filled with ones and zeros that restricts any full vector to the non-zero pattern $\mathcal{P}_i$. Also, define

$$b_i := \hat{S}^H A \hat{R}^H_{:,i}. \tag{3.11}$$

In practice, we choose $\mathcal{P}_i := \mathcal{P}(b_i)$ such that $Z^T Z_i b_i = b_i$. Then, we construct for each right hand side b_i the corresponding constrained Krylov subspace

$$\mathcal{K}^m_{\mathcal{P}_i} = \left\{ Z_i b_i\ ,\ Z_i A_{\mathcal{F}} Z_i^T Z_i b_i\ ,\ \dots\ ,\ (Z_i A_{\mathcal{F}} Z_i^T)^{m-1} Z_i b_i \right\}, \tag{3.12}$$

within which we approximate the multiplication of $A_{\mathcal{F}}^{-1}$ with b_i. Since each subspace is constructed under a sparsity constraint, the solution is approximated locally with a window of $A_{\mathcal{F}}$. Consequently, it is not guaranteed that the accuracy will increase with respect to m, however a few iterations are enough to reach a good approximation in practice. It is still possible to converge toward the best solution in a least squares sense, but this requires reformulating the problem with normal equations which increases the cost of construction.

4. Numerical Experiments. In this section, we present some numerical results performed with this new multigrid cycle. One pre-smoothing and one post-smoothing iteration of the normal equations Chebyshev polynomial presented in Section 2 are computed before each restriction and after each interpolation respectively. The interpolator $\hat{P}$ is constructed following Section 3. In practice, the degree d of the normal equation polynomial smoother p of Equation 2.2 is equal to 3, the $x_{\max}$ is computed by several power iterations (5 to 10 iterations are enough in practice) and $x_{\min}$ by the spectral density approximation described in Section 2.2, with the parameters $M = 5$ and $n_{\mathrm{vec}} = 15$. The number κ_l of input random vectors per level l to construct a correct approximation of near-kernel space follows the arbitrary recursive equation

$$\kappa_l = 5 \times \max_{i \in \mathcal{F}} \{\mathrm{Card}(\mathcal{C}_i)\} + \kappa_{l-1} + 10l\ ,\ \kappa_{-1} = 0 \tag{4.1}$$

where $\mathcal{C}_i$ is the set of coarse variables strongly connected to the fine variable i. The strongly connected variables are selected according to a θ-rule [16] comparing matrix entries in absolute value. Before being smoothed by the normal equation polynomial, test vectors are created either by the restriction of finer level test vectors or randomly. In practice, Card($\mathcal{C}_i$) never exceeds 10. The size m of the Krylov subspaces needed to approximate $A_{\mathcal{F}}^{-1}$ in the ideal framework is set to 3.

4.1. Benchmarks. We apply this multigrid method on a 5-points stencil Cartesian discretization of the Helmholtz equation (1.3) with absorbing boundary conditions ($\partial_n u - iku = 0$ on $\partial\Omega$), where 10 points per wavelength are used ($h = \lambda/10 \Leftrightarrow kh = 2\pi/10 \approx 0.625$). Multigrid cycles are iterated until the residual norm falls below 10^{-6}. This method will be benchmarked on more difficult geometries in the future, however this simple discretization already allows us to tackle the oscillating near-kernel space problem.

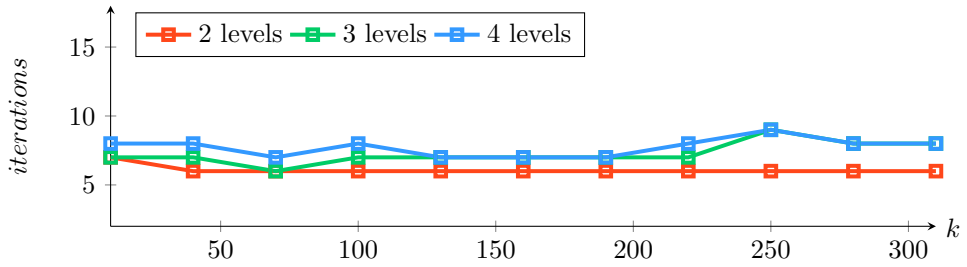

FIG. 4.1. *Number of iterations following the wavenumber k*

The three V-cycle schemes benchmarked in Figure 4.1 converge in a roughly constant number of iterations independent of the wavenumber k. We also present these results together with the overall complexity of our multigrid method. Let $\overline{nnz}(\cdot)$ be the average number of non-zeros per row of a given matrix. We measure the sparsity of each interpolator $\hat{P}_l$ and level matrix A_l in Table 4.1. As expected, matrices on deeper levels are denser. Even if $A_{\mathcal{F}}^{-1}$ is approximated under pattern constraints and already allows to find an interesting trade-off between sparsity and good interpolation properties, it will be necessary to improve the sparsity of $\hat{P}$ to reach a more competitive multigrid method. A thresholding strategy or other heuristics on its pattern will be one of the main concerns for the future developments of the method.

k	**70**	**100**	**130**	**160**	**190**	**220**	**250**	**280**	**310**
n ($\times 10^5$)	**1.28**	**2.59**	**4.37**	**6.60**	**9.30**	**12.46**	**16.08**	**20.16**	**24.7**
n_c ($\times 10^5$)	**0.16**	**0.32**	**0.55**	**0.83**	**1.16**	**1.56**	**2.01**	**2.52**	**3.09**
$\overline{\text{nnz}}(A_0)$	5	5	5	5	5	5	5	5	5
$\overline{\text{nnz}}(A_1)$	77	78	79	79	79	80	80	80	80
$\overline{\text{nnz}}(A_2)$	299	315	316	322	332	335	337	339	341
$\overline{\text{nnz}}(A_3)$	322	342	345	359	376	380	385	386	386
$\overline{\text{nnz}}(\hat{P}_1)$	20	20	20	20	20	20	20	20	20
$\overline{\text{nnz}}(\hat{P}_2)$	63	64	65	66	66	66	67	67	67
$\overline{\text{nnz}}(\hat{P}_3)$	168	178	178	183	188	190	191	192	193

TABLE 4.1
Sparsity measurement of level matrices and interpolators following k

4.2. Large shift experiment. Even if the approach presented in this paper can be improved in many ways, it provides a direction for constructing interpolation for problems with oscillatory near-kernel spaces like the Helmholtz equation. However, since the problem is indefinite, the matrix A does not provide a norm. As a consequence, there is no guidance on the convergence. Moreover, even under the convenient assumption where $\hat{P}\hat{P}^T v \approx v$ where v is an eigenvector associated with a very small eigenvalue, for some large shift problems, the method can be divergent. To illustrate this issue, let L_s be a scaled laplacian matrix shifted by a large coefficient α (yielding $\lambda(L_s) \in [-\alpha, 8-\alpha]$), such that its near-kernel space is very oscillatory.

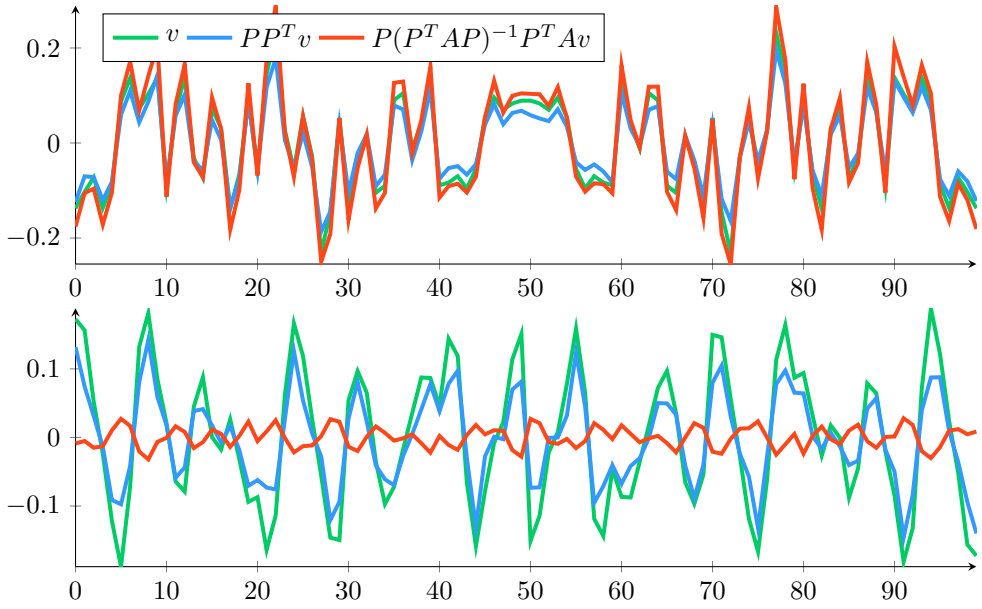

FIG. 4.2. *Layering of lowest eigenvector (green), interpolation of restricted lowest eigenvector (blue) and coarse correction applied to lowest eigenvector (red) for* $\alpha = 2.68$ *(top) vs.* $\alpha = 2.98$ *(bottom)*

Figure 4.2 shows the layering of the lowest eigenvector of L_s (v - green), the interpolation of the restricted lowest eigenvector ($\hat{P}\hat{P}^T v$ - blue), and the vector returned by the coarse correction ($\hat{P}(\hat{P}^T A\hat{P})^{-1}\hat{P}^T Av$ - red). As we can see, v and $\hat{P}\hat{P}^T$ are very close in both large shift experiments, the very oscillatory near-kernel vector v is well approximated by the interpolation range. However, in the second experiment, this is not true for the coarse correction where the red vector seems to be oriented oppositely to v. This experiment shows that capturing the oscillatory near-kernel space with an appropriate set of interpolators, such as $\hat{P}$, will not necessarily be enough to reach a perfectly recurring method for solving indefinite problems like Helmholtz. For this reason, adding more levels has been challenging. One of our priorities is about finding a framework guaranteeing the convergence of the method in an indefinite context such as this large shift experiment.

5. Conclusions. Indefinite and oscillatory problems are difficult for multigrid methods. The negative eigenvalues require an adapted smoother, and the interpolator should be able to propagate the oscillatory near-kernel space. We presented a method that reaches those requirements up to a certain limit. The normal equation polynomial smoother is designed to target a desired proportion of components according to their amplitude, and the range of our interpolator offers a good approximation of the near-kernel space despite its oscillations. Finding more accurate interpolation rules, improving the sparsity of our operators, and constructing a polynomial without resorting to normal equations will be important points in our future investigations. However, the ultimate objective is to find a proper framework for indefinite problems guaranteeing the convergence of our multigrid method.

REFERENCES

[1] M. F. Adams, M. Brezina, J. J. Hu, and R. S. Tuminaro, *Parallel multigrid smoothing: polynomial versus gauss–seidel*, Journal of Computational Physics, 188 (2003), pp. 593–610.

[2] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, *Multigrid smoothers for ultra-parallel computing*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2864–2887, https://doi.org/10.1137/100798806, https://doi.org/10.1137/100798806, https://arxiv.org/abs/https://doi.org/10.1137/100798806.

[3] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, *Bootstrap amg*, SIAM Journal of Scientific Computing, 33 (2011), pp. 612–632, https://doi.org/10.1137/090752973.

[4] L. I. Brandt A., *Wave-ray multigrid method for standing wave equations.*, ETNA. Electronic Transactions on Numerical Analysis [electronic only], 6 (1997), pp. 162–181, http://eudml.org/doc/119506.

[5] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αsa)*, SIAM Journal on Scientific Computing, 25 (2004), pp. 1896–1920, https://doi.org/10.1137/S1064827502418598, https://doi.org/10.1137/S1064827502418598, https://arxiv.org/abs/https://doi.org/10.1137/S1064827502418598.

[6] P. Ek, M. Brezina, and J. Mandel, *Convergence of algebraic multigrid based on smoothed aggregation*, Computing, 56 (1998), https://doi.org/10.1007/s002110000226.

[7] O. G. Ernst and M. J. Gander, *Why it is difficult to solve helmholtz problems with classical iterative methods*, (2010).

[8] R. D. Falgout, *An introduction to algebraic multigrid*, Computing in Science and Engineering, vol. 8, no. 6, November 1, 2006, pp. 24-33, (2006), https://www.osti.gov/biblio/897960.

[9] R. D. Falgout and P. S. Vassilevski, *On generalizing the amg framework*, SIAM J. NUMER. ANAL, 42 (2003), pp. 1669–1693.

[10] L. Lin, Y. Saad, and C. Yang, *Approximating spectral densities of large matrices*, SIAM Review, 58 (2016), pp. 34–65, https://doi.org/10.1137/130934283, https://doi.org/10.1137/130934283, https://arxiv.org/abs/https://doi.org/10.1137/130934283.

[11] I. Livshits, *A scalable multigrid method for solving indefinite helmholtz equations with constant wave numbers*, Numerical Linear Algebra with Applications, 21 (2014), https://doi.org/10.1002/nla.1926.

[12] I. Livshits, *Multiple galerkin adaptive algebraic multigrid algorithm for the helmholtz equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. S195–S215, https://doi.org/10.1137/140975310, https://doi.org/10.1137/140975310, https://arxiv.org/abs/https://doi.org/10.1137/140975310.

[13] L. Olson and J. Schroder, *Smoothed aggregation for helmholtz problems*, Numerical Linear Algebra with Applications, 17 (2010), pp. 361 – 386, https://doi.org/10.1002/nla.686.

[14] L. N. Olson, J. B. Schroder, and R. S. Tuminaro, *A general interpolation strategy for algebraic multigrid using energy minimization*, SIAM Journal on Scientific Computing, 33 (2011), pp. 966–991, https://doi.org/10.1137/100803031, https://doi.org/10.1137/100803031, https://arxiv.org/abs/https://doi.org/10.1137/100803031.

[15] G. Strang, *Multigrid methods*, tech. report, MIT, 2006, https://math.mit.edu/classes/18.086/2006/am63.pdf.

[16] K. Stüben, *Algebraic multigrid (amg). an introduction with applications*, (1999).