AMG to solve Laplace's equation

We use Laplace's equation in two dimensions with rectangular boundaries as a simple model to test the Algebraic Multigrid Method (AMG) with *standard coarsening* and *direct interpolation* (see Refs. [1, 2]). Later we also compare with Smoothed Aggregation-Based AMG (see [3]). The algorithms were implemented with python and can be used for any matrix problem as long as the matrix operator and the boundary conditions are known. Here we only refer to Laplace's equation for convenience. Implementations are on this GitHub repository.

1 Model problem

Let us consider Laplace's equation

$$\nabla^2 u(x,y) = 0 \tag{1}$$

with boundary conditions

$$\begin{aligned} u(x,H) = 0, \quad u(L,y) = 0, \\ u(0,y) = 0, \quad u(x,0) = V(x), \end{aligned} \tag{2}$$

where $V(x)$ is a function of x. The general solution to this problem is

$$u(x,y) = \sum_{n=1}^{\infty} a_n \sin\frac{n\pi x}{L} \sinh\frac{n\pi(H-y)}{L}, \tag{3}$$

where

$$a_n = \frac{2}{L\sinh\frac{n\pi H}{L}} \int_0^L V(x)\sin\frac{n\pi x}{L}dx. \tag{4}$$

For simplicity, we choose $V(x) = \sin\frac{2\pi x}{L}$, which yields

$$u(x,y) = \frac{\sin\frac{2\pi x}{L}\sinh\frac{2\pi(H-y)}{L}}{\sinh\frac{2\pi H}{L}}. \tag{5}$$

This is useful to verify the results of the algorithms.

We proceed to discretize $\nabla^2 u(x,y) = 0$. Let us consider a grid of dimensions $(N+1)\times(N+1)$ with grid space given by h. The relevant equation for the interior grid points (the boundaries are already known) is

$$u(x+h,y) + u(x-h,y) - 4u(x,y) + u(x,y+h) + u(x,y-h) = 0, \tag{6}$$

or written with a different notation

$$u_{i+1j} + u_{i-1j} - 4u_{ij} + u_{ij+1} + u_{ij-1} = 0, \quad i,j = 0,1\ldots,N. \tag{7}$$

These equations can be arranged as a matrix product

$$A\vec{u} \equiv \begin{pmatrix} T & I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ I & T & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & I & T & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & I & T & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots I & T & I \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots 0 & I & T \end{pmatrix} \begin{pmatrix} u_{11} \\ u_{12} \\ u_{13} \\ \vdots \\ u_{N1} \\ \vdots \\ u_{NN} \end{pmatrix} = \vec{f}, \tag{8}$$

where T is the following tridiagonal matrix of dimension $N \times N$

$$\begin{pmatrix} -4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -4 & 1 \end{pmatrix} \tag{9}$$

and I the identity matrix of the same dimension. A has, however, dimension $N^2 \times N^2$. The vector $\vec{f}$ contains the information of the boundary and has the following structure (we write it for $N = 4$ for simplicity)

$$\vec{f} = \begin{pmatrix} -u_{01} - u_{10} \\ -u_{02} \\ -u_{03} \\ -u_{04} - u_{15} \\ -u_{20} \\ 0 \\ 0 \\ -u_{25} \\ -u_{30} \\ 0 \\ 0 \\ -u_{35} \\ -u_{40} - u_{51} \\ -u_{52} \\ -u_{53} \\ -u_{45} - u_{54} \end{pmatrix}. \tag{10}$$

The easiest way of solving the problem is just by computing the inverse $\vec{u} = A^{-1}\vec{f}$. For very large matrices this is, of course, very inefficient. Still, for comparing the solution with AMG is worth computing the inverse of A. To relate the solution on the grid with the analytical solution we consider $x_i = ihL$ and $y_j = jhH$ where $h = 1/(N+1)$, then $u(x_i, y_j) = u_{ij}$.

2 AMG with standard coarsening and direct interpolation

We explain the idea of the two-grid method, for the multilevel case we just have to apply the two-grid method recursively. Let us introduce the notation. The index h defines the fine level, while H defines the coarse level. We want to solve the problem

$$A_h u^h = f^h, \quad \sum_{j \in \Omega^h} a_{ij}^h f_j^h \quad i \in \Omega^h, \tag{11}$$

where $\Omega^h = \{1, 2, \ldots, N^2\}$ represents the set of all variables. We split Ω^h into a set of fine-variables and coarse-variables $\Omega^h = F \cup C$. The coarse level is given by $\Omega^H = C$. Once the procedure to create these sets is defined, the prolongation, $I_H^h : \Omega^H \to \Omega^h$, and restriction, $I_h^H : \Omega^h \to \Omega^H$, operators can be constructed. We will consider $I_h^H = (I_H^h)^T$. The matrix on the coarse level is taken as the Galerkin operator

$$A_H = I_h^H A_h I_H^h. \tag{12}$$

The coarse-grid correction is obtained by computing

$$u_{\text{new}}^h = u_{\text{old}}^h + A_H e^H, \quad A_H e^H = I_h^H (f^h - A_h u_{\text{old}}^h). \tag{13}$$

Before applying the correction, u^h has to be approximated with a smoother. We use the Gauss-Seidel method, which is described by

$$u_i^{(k+1)} = \frac{1}{a_{ii}} \left(f_i - \sum_{j=i+1}^{n-1} a_{ij} u_j^{(k)} - \sum_{j=0}^{i-1} a_{ij} u_j^{(k+1)} \right) \quad i = 0, 1, \ldots, n-1, \tag{14}$$

where (k) denotes the iteration number.

2.1 Standard coarsening

We explain the process in a mere practical way, without dwelling on theoretical details. We need to create strong couplings between the variables of Ω^h. Three sets need to be defined

$$\begin{aligned} N_i &= \{j \in \Omega^h : j \neq i \text{ and } a_{ij} \neq 0\}, \\ S_i &= \{j \in N_i : -a_{ij} \geq \epsilon \max(|a_{ik}|, a_{ik} < 0)\}, \\ S_i^T &= \{j \in N_i : i \in S_j\}. \end{aligned} \tag{15}$$

The value of ϵ can be arbitrarily chosen in $(0, 1)$, it defines the strength of the coupling between the variables. The set S_i contains all the strong couplings of i, while S_i^T contains all variables j which are strongly coupled to i. These two sets are not necessarily equal, but for the model problem here presented they are indeed the same.

The coarsening works as follows:

1. Define some first variable i to become a C-variable. All variables j that are strongly connected with i, *i.e.* $\forall j \in S_i^T$, become F-variables.
2. From the remaining undecided variables, another one is defined to become a C-variable and all variables, which are strongly connected to it and are still undecided, become F-variables.

To obtain an approximately uniform distribution of C- and F- variables, we use a "measure of importance" in step 2 for any undecided variable

$$\lambda_i = |S_i^T \cap U| + 2|S_i^T \cap F|, \quad i \in U, \tag{16}$$

where U is the set of undecided variables and $|.|$ denotes the cardinality. The U-variable with the largest λ_i will be the new C-variable. If more than U-variable has the largest λ_i, we choose randomly. Each time new C- and F-variables are defined, we have to update the value of λ_i for the undecided variables.

3. The process stops if all variables are either C-variables or F-variables.

For the particular case of Laplace's equation it can be shown that this process leads to a checkerboard pattern formed by the C- and F-variables.

2.2 Direct interpolation

Once the C- and F-variables are defined, we can interpolate a vector from the coarse grid to the fine one by computing

$$e_i^h = (I_H^h e^H)_i = \begin{cases} 1 & \text{if } i \in C, \\ \sum_{k \in P_i} \omega_{ik} e_k^H & \text{if } i \in F, \end{cases} \tag{17}$$

where ω_{ik} are the interpolation weights and $P_i \subseteq C \cap N_i$. We will consider

$$P_i = C \cap S_i \subseteq C \cap N_i. \tag{18}$$

For direct interpolation, the weights are given by (see Refs. [1, 2])

$$\omega_{ij} = -\left(\frac{\sum_{k \in N_i} a_{ik}}{\sum_{k \in P_i} a_{ik}} \right) \frac{a_{ij}}{a_{ii}}, \quad i \in F, \quad j \in P_i. \tag{19}$$

Once the weights are known, we can use the action of the prolongation operator, eq. (17), to create I_H^h. We just have to map the basis vectors on the coarse grid and rearrange the result in columns to form the matrix representation of I_H^h. After that, the restriction operator and subsequently the Galerkin operator are obtained.

The two-grid method can be implemented by doing the following steps:

- Construct N_i, S_i and S_i^T.
- Apply standard coarsening.
- Construct the prolongation operator and with it the restriction and Galerkin operator.
- Perform ν_1 Gauss-Seidel steps to obtain an approximate solution u (pre-smoothing).
- Apply a coarse-grid correction to improve the smooth part of the error, which Gauss-Seidel cannot efficiently deal with

$$e^H = A_H^{-1} I_h^H (f^h - A_h u^h), \quad u \to u + I_H^h e^H. \tag{20}$$

- Perform ν_2 Gauss-Seidel steps (post-smoothing).

The first three steps conform the setup phase of the two-grid method. The multigrid method is obtained when we apply recursively the two-grid method to solve the equation

$$A_H e^H = f^H, \quad f^H \equiv I_h^H(f^h - A_h u^h). \tag{21}$$

We write a recipe to perform a V-cycle with AMG using a recursive algorithm. The function V_cycle receives ν_1, ν_2, l, f_l and u_l as input parameters. l refers to the grid-level, f_l to the residual restriction at that level and u_l to the solution to eq. (21).

- We first define a number of grid-levels, $\max_l$, and construct the prolongation P_l and Galerkin operators A_l for each level in a setup phase.
- If $l = \max_l$ then return $A_l^{-1} f_l$ (at this point the matrix A_l should have a dimension considerably smaller than the original matrix of the problem).
- If $l < \max_l$ perform ν_1 pre-smoothing steps to approximate u_l and compute the restriction $f_{l+1} = P_l^T(f_l - A_l u_l)$.
- Give an initial solution for u_{l+1} and call V_cycle$(\nu_1, \nu_2, l+1, f_{l+1}, u_{l+1})$.
- Apply a coarse-grid correction $u_l = u_l + P_l u_{l+1}$.
- Perform ν_2 post-smoothing steps.

In Table 1 we show results of some components of the vector $\vec{u}$ (the one of eq. (8)) when computed by inverting the Laplace matrix operator, applying Gauss-Seidel, the two-grid method and AMG with 4 levels. For the latter we use a V-cycle. The Gauss-Seidel solution was obtained with 10 iterations, while for the two- and four-level AMG method we perform 5 pre-smoothing and 5-post smoothing steps at each grid level. We fix $\epsilon = 0.25$. The computing times are displayed in Table 2, the setup phase is not included there. For the two-grid method the setup took 10.68(3) s and for AMG with 4 levels the time was 12.87(4) s. We observe that AMG with 4 levels using a V-cycle is the fastest method (after the setup), although its precision is not as good as with the two-grid method. Still, the results coincide up to the second digit, except for the Gauss-Seidel method which is quite inefficient by itself. Precision can be improved by increasing the number of smoothing steps in any case.

u_{ij}	Analytical	Exact Inversion	Gauss-Seidel	Two-Grid	AMG 4 levels
u_{11}	0.10864	0.10866	0.08808	0.10863	0.10854
u_{19}	0.04055	0.0406	0.00102	0.04051	0.03994
u_{38}	0.13482	0.13499	0.00714	0.13471	0.13319

Table 1: Comparison of the exact analytical solution, the result of inverting the full matrix operator, Gauss-Seidel's solution, the two-grid method and AMG with 4 levels. AMG is based on standard coarsening and direct interpolation. A number of $N = 50^2$ variables was considered, so the Laplace matrix operator has 50^4 entries.

3 AMG with smoothed aggregation

The idea of AMG with aggregation is different from the C/F splitting. In aggregation schemes we divide the total number of variables into disjoint subsets called aggregates and assign the same

Exact Inversion	Two-Grid	AMG 4 levels
0.199(2) s	0.146(2) s	0.133(1) s

Table 2: Time taken to compute the solution of the $N = 50^2$ variables using different methods. The setup phase is not considered to measure the time. The results were obtained in the workstation with an i7-12700 × 20 processor. We see that an exact inversion is the slowest method for the matrix size.

value for interpolation to all variables in the aggregate. This is essentially piecewise constant interpolation. If we call the aggregates C_k, then the Galerkin operator is just computed as

$$a_{kl}^H = I_h^H A_h I_H^h = \sum_{i \in C_k} \sum_{j \in C_l} a_{ij}^h, \quad (k,\, l \text{ indexing the aggregates}), \tag{22}$$

where

$$(I_H^h)_{ij} = \begin{cases} 1 & \text{if } i \in C_j, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

In some way, this is similar to interpreting the aggregates as C-variables.

Piecewise constant interpolation only makes sense if the aggregates are constructed with variables that strongly depend on each other. A way of improving the interpolation matrix consists of applying a damped Jacobi smoother, as proposed by [3]. This is known as smoothed aggregation. We will describe an algorithm for constructing the aggregates and the operator I_H^h.

Let $\epsilon \in (0, 1)$ and define the strongly-coupled neighborhood of a variable i as

$$N_i = \{j : |a_{ij}| \geq \epsilon\sqrt{a_{ii}a_{jj}}\}. \tag{24}$$

Let A be the matrix operator of dimension $n_l \times n_l$, with $n_l = N^2$ for the model problem. We have to generate a disjoint cover of the set $\{1, 2, \ldots, n_l\}$. Let us call such a cover $\{C_i\}_{i=1}^{n_{l+1}}$, where n_{l+1} represents the number of aggregates. The algorithm works as follows:

1. Define $R = \{1, 2, \ldots, n_l\}$ and $j = 0$. Select disjoint neighborhoods to form an initial approximation of the covering, *i.e.*

   $\forall i \in R$ if $N_i^l \subset R \Rightarrow j \to j + 1, C_j = N_i, R \to R \backslash C_j$. This does not necessarily cover the whole set $\{1, 2, \ldots, n_l\}$.

2. If there are some $i \in R$ left, we add them to the sets C_k, $k = 1, \cdots, j$, by doing the following: $\forall i \in R$ if $\exists k$ such that $N_i \cap C_k \neq \emptyset \Rightarrow C_k \to C_k \cup \{i\}, R \to R\backslash\{i\}$. If more than one set C_k exists, then we choose the one to which node i has the strongest coupling (the k for which $|a_{ik}|/\sqrt{a_{ii}a_{kk}}$ is largest). If i has more than one strongest coupling we pick randomly.

3. In case there is still any remnant in R, we pick $i \in R$ and update $j \to j+1, C_j = R \cap N_i, R \to R\backslash C_j$.

The final value of j will be n_{l+1} and R should be empty. After this process we define

$$(\widehat{I_H^h})_{ij} = \begin{cases} 1 & \text{if } i \in C_j, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

A Jacobi step is used to improve the interpolator

$$I_H^h = (I - \omega D^{-1} A^F)\widehat{I_H^h}, \tag{26}$$

where I is the identity matrix, ω is a free parameter usually fixed to be $\omega = 2/3$, $D = \text{diag}(A^F)$ and

$$a_{ij}^F = \begin{cases} a_{ij} & \text{if } j \in N_i, \\ 0 & \text{otherwise,} \end{cases} \quad i \neq j, \tag{27}$$

$$a_{ii}^F = a_{ii} - \sum_{j=1, j\neq i}^{n_l} (a_{ij} - a_{ij}^F). \tag{28}$$

A_F is called the filtered matrix. Once the interpolator is constructed, the multigrid algorithm is implemented in the same way as with the C/F splitting, only the setup phase is different.

In Table 3 we present the results of some components of the vector $\vec{u}$ (similar to Table 1). The same parameters ν_1, ν_2 are considered and we fix $\omega = 2/3$. For ϵ we choose

$$\epsilon = 0.08 \left(\frac{1}{2}\right)^l, \tag{29}$$

as suggested in Ref. [3]. One has to carefully fix this number, because if it is too large it can lead to a number of aggregates equal to the number of total variables, so in the end the coarse grid would be exactly the same as the fine grid.

We observe that, for a fixed number of pre- and post-smoothing steps, AMG based on smoothed aggregation with a V-cycle is slightly less precise than AMG with standard coarsening and direct interpolation, as can be seen from the last two columns of Tables 1 and 3. However, computing times are shorter, even in the setup phase. For the two-grid case the setup took 5.07(3) s and for the four level case 5.23(4) s. The solution is also faster than with standard coarsening, mainly because the number of aggregates is usually less than the number of C-variables at each coarse level, so it is faster to perform operations with the interpolation and Galerkin operators since their dimension is not too big. Once again, precision can be improved by increasing the smoothing steps.

u_{ij}	Analytical	Exact Inversion	Gauss-Seidel	Two-Grid	AMG 4 levels
u_{11}	0.10864	0.10866	0.08808	0.10839	0.10821
u_{19}	0.04055	0.0406	0.00102	0.03909	0.03777
u_{38}	0.13482	0.13499	0.00714	0.12991	0.12645

Table 3: Same as Table 1 but results obtained through AMG based on smoothed aggregation.

Exact Inversion	Two-Grid	AMG 4 levels
0.199(2) s	0.098(2) s	0.104(1) s

Table 4: Similar to Table 2. Computing times for the AMG are shorter than with standard coarsening, because the number of aggregates is less than the number of C-variables per level.

References

[1] K. Stüben, "Algebraic Multigrid (AMG): An Introduction with Applications", GMD, (1999).

[2] J. Volker, "Multigrid Methods", Weierstrass Institute, (2013).

[3] P. Vaněk, J. Mandel and M. Brezina, "Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems", *Computing* **56**, 179-196 (1996).