Dr H. Fawzi

**Mathematical Tripos Part II: Michaelmas Term 2021**

# Numerical Analysis – Lecture 17

**Method 4.17 (Multigrid methods)** The speed of convergence of some iterative methods (Jacobi with relaxation, Gauss–Seidel, etc.) can be increased drastically when the linear system originates in the discretization of PDEs, using *multigrid methods*. Here we look at the system $A\boldsymbol{u} = \boldsymbol{b}$ originating from the 3-point formula for the Poisson equation on an $m$-grid $\Omega_h = \{ih : i \in (0, m)\}$, $h = 1/(m+1)$, being solved by the weighted Jacobi iteration.

Recall that the matrix $A$ in this case is given by

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

The diagonal part of $A$ is $D = 2I$. Thus the weighted Jacobi iterations takes the form:

$$\boldsymbol{u}^{(\nu+1)} = H_\omega \boldsymbol{u}^{(\nu)} + (\omega/2)\boldsymbol{b}$$

where $H = I - D^{-1}A = I - \frac{1}{2}A$, and $H_\omega = \omega H + (1-\omega)I = I - \frac{\omega}{2}A$. The error decay is expressed in terms of the iteration matrix $H$:

$$\boldsymbol{e}^{(\nu)} = H_\omega^\nu \boldsymbol{e}^{(0)}.$$

We know from the results of Lecture 2 that the eigenvectors and the eigenvalues of $H_\omega$ are

$$\boldsymbol{w}^k = \left[\sin i\tfrac{k\pi}{m+1}\right]_{i=1,\ldots,m}, \qquad \lambda_k(\omega) = 1 - 2\omega\sin^2\tfrac{k\pi}{2(m+1)} \qquad (k = 1,\ldots,m).$$

Consider the choice $\omega = 1/2$; then the eigenvalues of $H_\omega$ are $\lambda_k = 1 - \sin^2\frac{k\pi}{2(m+1)} = \cos^2\frac{k\pi}{2(m+1)}$. With this choice, the eigenvalues are all positive and decreasing with $k$. In particular $\rho(H) = \lambda_1 = \cos^2\frac{\pi}{2(m+1)} \approx 1 - \frac{\pi^2}{4m^2} < 1$, guaranteeing convergence, although a very slow one! However, expanding the error with respect to the (orthogonal) eigenvectors we obtain

$$\boldsymbol{e}^{(\nu)} = \sum_{k=1}^{m} a_k^{(\nu)}\boldsymbol{w}^k, \qquad \boldsymbol{e}^{(\nu)} = H_\omega^\nu\boldsymbol{e}^{(0)} \quad \Rightarrow \quad |a_k^{(\nu)}| = |\lambda_k|^\nu |a_k^{(0)}|,$$

i.e. the components of $\boldsymbol{e}^{(\nu)}$ (with respect to the basis of eigenvectors) decay at a different rate for different frequencies $k = 1,\ldots,m$. More precisely, the high frequencies, where $k$ is close to $m$, will decay faster than the low frequencies, where $k$ is closer to 1. Let us say that $k \in (0, m+1) = (0, \frac{1}{h})$ is high frequency (HF) wrt grid $\Omega_h$ if $kh \geq 1/2$ (i.e., $\frac{m+1}{2} \leq k \leq m$). Then the decay rate for the high frequency components of the error $\boldsymbol{e}$ is at least:

$$\mu_* = |\lambda_{(m+1)/2}| = 1 - \sin^2(\pi/4) = 1/2.$$

Therefore, for the coefficients at the HF components of $\boldsymbol{e}^{(\nu)}$ we obtain

$$|a_k^{(\nu)}| \leq |\mu_*|^\nu |a_k^{(0)}| = \left(\tfrac{1}{2}\right)^\nu |a_k^{(0)}| \ll |a_k^{(0)}|,$$

i.e. the Jacobi method converges fast for high frequencies.

The main observation of the multigrid is to note that the low frequencies $k \in (\frac{1}{4h}, \frac{1}{2h})$ with respect to the grid $\Omega_h$ become high frequencies for the *coarser grid* $\Omega_{2h}$ with step $2h$; indeed for such $k$ we have $k(2h) \geq 1/2$.

The idea of the multigrid method then is that, although the global error may decrease slowly by iteration, its components with high frequencies relative to $\Omega_h$ are suppressed (or smoothed) very quickly, and that dealing with the remaining components (with low frequencies relative to $\Omega_h$) we can move to the coarse grid $\Omega_{2h}$, where these components (in part) would be of high frequencies, and thus they can be smoothed in a similar way. Therefore, we cover the domain $[0,1]$ by a range of nested grids, of increasing coarseness, say,

$$\Omega_h \subset \Omega_{2h} \subset \Omega_{4h} \subset \cdots \subset \Omega_{2^j h}\,.$$

At every $\Omega_{h_i}$, the iterations (Jacobi, or Gauss-Seidel) remove the high frequencies relative to this grid, and we move to $\Omega_{2h_i}$. On the coarsest grid, where the number of variables is small, we can afford to solve the equations with a direct method, by Cholesky, say.

A typical multigrid method can be summarized as follows. We call $A_h$ the discretized Poisson matrix for grid size $h$.

1. Presmoothing: Perform a small number $\nu$ (typically 2,3) of Jacobi iterations on $A_h \boldsymbol{u}^h = \boldsymbol{b}^h$.
2. Let $\boldsymbol{r}^h = \boldsymbol{b}^h - A_h \boldsymbol{u}^h$ be the residual.
3. Let $R_h^{2h} : \mathbb{R}^m \to \mathbb{R}^{m/2}$ be a *restriction operator* that restricts vectors on the fine grid $\Omega_h$ to vectors on the coarse grid $\Omega_{2h}$; and let $I_{2h}^h$ be an *interpolation operator* that interpolates vectors on the coarse grid $\Omega_{2h}$ to vectors on the fine grid $\Omega_h$.
4. Recurse: Use the multigrid method to solve $A_{2h}\boldsymbol{\delta}^{2h} = \boldsymbol{r}^{2h}$, where $\boldsymbol{r}^{2h} = R_h^{2h}\boldsymbol{r}^h$ (if the size of this linear system is small, use a direct solver instead of recursing)
5. Let $\boldsymbol{u}^h = \boldsymbol{u}^h + I_{2h}^h \boldsymbol{\delta}^{2h}$
6. Postsmoothing: apply a few Jacobi iterations (typically 2,3) starting from $\boldsymbol{u}^h$ on $A_h \boldsymbol{u}^h = \boldsymbol{b}^h$

To make the algorithm above complete, one needs to define the restriction and the interpolation operators. A common choice for the interpolation operator is linear interpolation, i.e.,

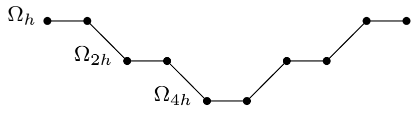
$$(I_{2h}^h \boldsymbol{v})_{2i} = \boldsymbol{v}_i \quad \text{and} \quad (I_{2h}^h \boldsymbol{v})_{2i+1} = (\boldsymbol{v}_i + \boldsymbol{v}_{i+1})/2$$

for $i \in [0, \frac{1}{2h})$. A natural choice for the restriction operator is the canonical injection $(R_h^{2h}\boldsymbol{v})_i = \boldsymbol{v}_{2i}$; another more common choice is to take $R_h^{2h}$ to be an averaging operator

$$(R_h^{2h}\boldsymbol{v})_i = \frac{1}{4}(\boldsymbol{v}_{2i-1} + 2\boldsymbol{v}_{2i} + \boldsymbol{v}_{2i+1})$$

for $i \in (0, \frac{1}{2h})$. (The latter corresponds, up to scaling, to the transpose of the linear interpolation operator above.)

If we follow the recursive procedure outlined above, then we see that the algorithm starts at the finest grid, travels to the coarsest (where we apply a direct solver), and back to the finest:



For this reason, the algorithm above describes what is known as a V-cycle.

**Matlab demo:** Download the Matlab GUI for *Multigrid Methods* from `https://www.damtp.cam.ac.uk/user/hf323/M21-II-NA/demos/multigrid/multigrid.html` and see the tremendous effect of multigrid (in comparison with Jacobi and Gauss-Seidel) for solving the Poisson equation with a forcing term $f$ that possesses multiple frequencies.