Numerical Analysis – Lecture 17[1]

Observation 4.12 *(Attenuation for different frequencies)* The speed of convergence of some iterative methods, e.g. Gauss–Seidel, can be increased drastically within the context of solving linear equations that originate in the discretization of PDEs. Herewith we analyse (with a great deal of hand-waving) the 5-point formula in an $m \times m$ square grid, being solved by the Gauss–Seidel iteration. We commence from a computational observation: once the Gauss–Seidel method is applied, the error *in the first few iterations* decays very substantially, roughly by a factor of $\frac{1}{2}$ in each iteration. Subsequently, everything slows down and the method settles to its excruciatingly-slow asymptotic rate of convergence.

We let $\varepsilon_{l,j}^{(k)} = u_{l,j}^{(k)} - u_{l,j}$, $l, j = 1, \ldots, m$ be the error in the solution of the linear equations.[2] Subtracting,

$$\begin{array}{lrcl} & -4u_{l,j}^{(k+1)} + u_{l-1,j}^{(k+1)} + u_{l+1,j}^{(k)} + u_{l,j-1}^{(k+1)} + u_{l,j+1}^{(k)} & = & (\Delta x)^2 f_{l,j} \\ - & -4u_{l,j} \quad + \ u_{l-1,j} + u_{l+1,j} + u_{l,j-1} \ + \ u_{l,j+1} & = & (\Delta x)^2 f_{l,j} \\ \hline = & -4\varepsilon_{l,j}^{(k+1)} + \varepsilon_{l-1,j}^{(k+1)} + \varepsilon_{l+1,j}^{(k)} + \varepsilon_{l,j-1}^{(k+1)} + \varepsilon_{l,j+1}^{(k)} & = & 0. \end{array}$$

Let $r_{\theta,\psi}^{(k)} = \sum_{l,j} \varepsilon_{l,j}^{(k)} \mathrm{e}^{\mathrm{i}(l\theta + j\psi)}$. It is possible to prove that, in suitable norms, $\|\boldsymbol{r}^{(k)}\| = |||\boldsymbol{\varepsilon}^{(k)}|||$. (More about Fourier transforms and norms later!) Were the boundary conditions *periodic*, it would have been easy to prove that

$$(4 - \mathrm{e}^{\mathrm{i}\theta} - \mathrm{e}^{\mathrm{i}\psi}) r_{\theta,\psi}^{(k+1)} = (\mathrm{e}^{-\mathrm{i}\theta} + \mathrm{e}^{-\mathrm{i}\psi}) r_{\theta,\psi}^{(k)}$$

– for Dirichlet boundary conditions this isn't strictly true but the difference is of a lower order of magnitude and we'll disregard it. Thus, the *local attenuation* of the error is roughly

$$\rho_{\theta,\psi}^{(k+1)} = \left| \frac{r_{\theta,\psi}^{(k+1)}}{r_{\theta,\psi}^{(k)}} \right| = \left| \frac{\mathrm{e}^{\mathrm{i}\theta} + \mathrm{e}^{\mathrm{i}\psi}}{4 - \mathrm{e}^{\mathrm{i}\theta} - \mathrm{e}^{\mathrm{i}\psi}} \right|.$$

Note that the frequencies 'supported' by the grid live in $[-\pi, \pi]$: wiggles between grid points (i.e., frequencies which lie outside $[-\pi, \pi]$ relative to the grid) pass unnoticed!
Bad news: For $|\theta|, |\psi| = \mathcal{O}\big(m^{-1}\big)$ we obtain $\rho_{\theta,\psi}^{(k+1)} \approx 1 - cm^{-2}$ for some $c > 0$, the disappointingly small attenuation already familiar from the analysis of Gauss–Seidel.
Good news: Consider just the *large frequencies.* Then

$$\max_{\frac{\pi}{2} \le \max\{|\theta|, |\psi|\} \le \pi} \rho_{\theta,\psi}^{(k+1)} = \rho_{\frac{\pi}{2}, \tan^{-1}\frac{3}{4}}^{(k+1)} = \tfrac{1}{2}.$$

Thus, Gauss–Seidel converges fast for *large frequencies.*

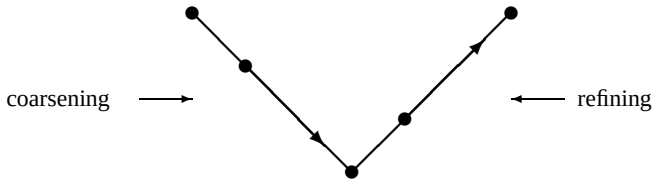
Algorithm 4.13 *(The multigrid method)* Suppose that we cover the square domain by a range of nested grids, of increasing coarseness, $D_m \subset D_{m/2} \subset \cdots$, say. The corresponding 'fast' frequencies are

The idea is to cover the whole relevant range of frequencies by 'fast bands'. In other words, the purpose of Gauss–Seidel iterations is solely to remove the contribution of fast frequencies relative to each nested grid.

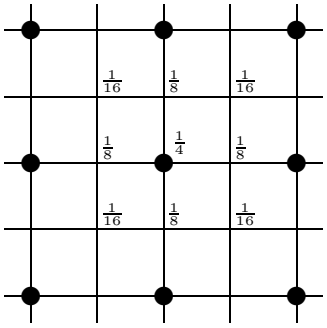
A *multigrid sweep* starts at the finest grid, travels to the coarsest (where the number of variables is small and we can afford to solve the equations with Cholesky, say) and back to the finest:

[1] Please email all corrections and suggestions to these notes to `A.Iserles@damtp.cam.ac.uk`. All handouts are available on the WWW at the URL `http://www.damtp.cam.ac.uk/user/na/PartII/Handouts.html`.

[2] $\varepsilon_{l,j}$ should not be confused with the error in the solution of the original Poisson equation!



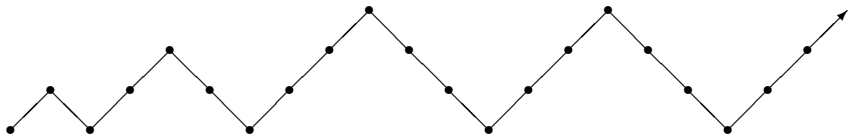
Each *coarsening* stage involves computing the residual $\boldsymbol{r}_h = \boldsymbol{b}_h - A_h\boldsymbol{x}_h$ (h is the size of the grid) and restricting it to the coarser grid. There again we are solving *for the residual*, i.e. we iterate for the equations $A_{2h}\boldsymbol{y}_{2h} = \boldsymbol{r}_{2h}$. A good restriction combines 9 'fine' values according to the rule



Refinement entails a prolongation $\boldsymbol{y}_h = P\boldsymbol{y}_{2h}$ by linear interpolation (the exact opposite of the above procedure) and correction $\boldsymbol{x}_h^{\text{new}} = \boldsymbol{x}_h^{\text{old}} + \boldsymbol{y}_h$.

It is usual to employ only a moderate number of iterations in each restriction (3–5, say) and prolongation (just 1–2 iterations, to take care of fast frequencies that have been reintroduced by prolongation) and to check for convergence only by the end of the sweep. Unless convergence occurs, we embark on another multigrid sweep and so on.

Algorithm 4.14 *(Full multigrid)* Start from the coarsest grid and advance to the finest in a 'zig-zag' fashion to obtain good starting value, subsequently continue with the V-cycles, as above:



Observation 4.15 *(Special structure of 5-point equations)* We wish to motivate and introduce our next family of efficient solution methods for the 5-point equations: the *fast Poisson solvers.* Thus, suppose that we are solving $\nabla^2 u = f$ in a square $m \times m$ grid with the 5-point formula (all this can be generalized a great deal, e.g. to the nine-point formula). Let the grid be enumerated in *natural ordering*, i.e. by columns. Thus, the linear system $\mathcal{A}\boldsymbol{u} = \boldsymbol{b}$ can be written explicitly in the form

$$\mathcal{A} = \begin{bmatrix} A & I & & 0 \\ I & A & I & \\ & \ddots & \ddots & \ddots \\ 0 & & I & A \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{m-1} \\ \boldsymbol{u}_m \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_{m-1} \\ \boldsymbol{b}_m \end{bmatrix}, \quad A = \begin{bmatrix} -4 & 1 & & 0 \\ 1 & -4 & 1 & \\ & \ddots & \ddots & \ddots \\ 0 & & 1 & -4 \end{bmatrix}.$$

The matrix A is $m \times m$, *tridiagonal*, *symmetric* and *Toeplitz* (i.e., constant along diagonals): we call such a matrix *TST.* You can verify easily that its eigenvalues are $-4 + 2\cos\frac{k\pi}{m+1}$, $k = 1, \ldots, m$, and the corresponding orthogonal eigenvectors are $q_{k,l} = \sqrt{\frac{2}{m+1}}\sin\frac{kl\pi}{m+1}$, $k, l = 1, \ldots, m$. *[Note that all $m \times m$ TST matrices share the same full set of eigenvectors, hence they all commute!]*