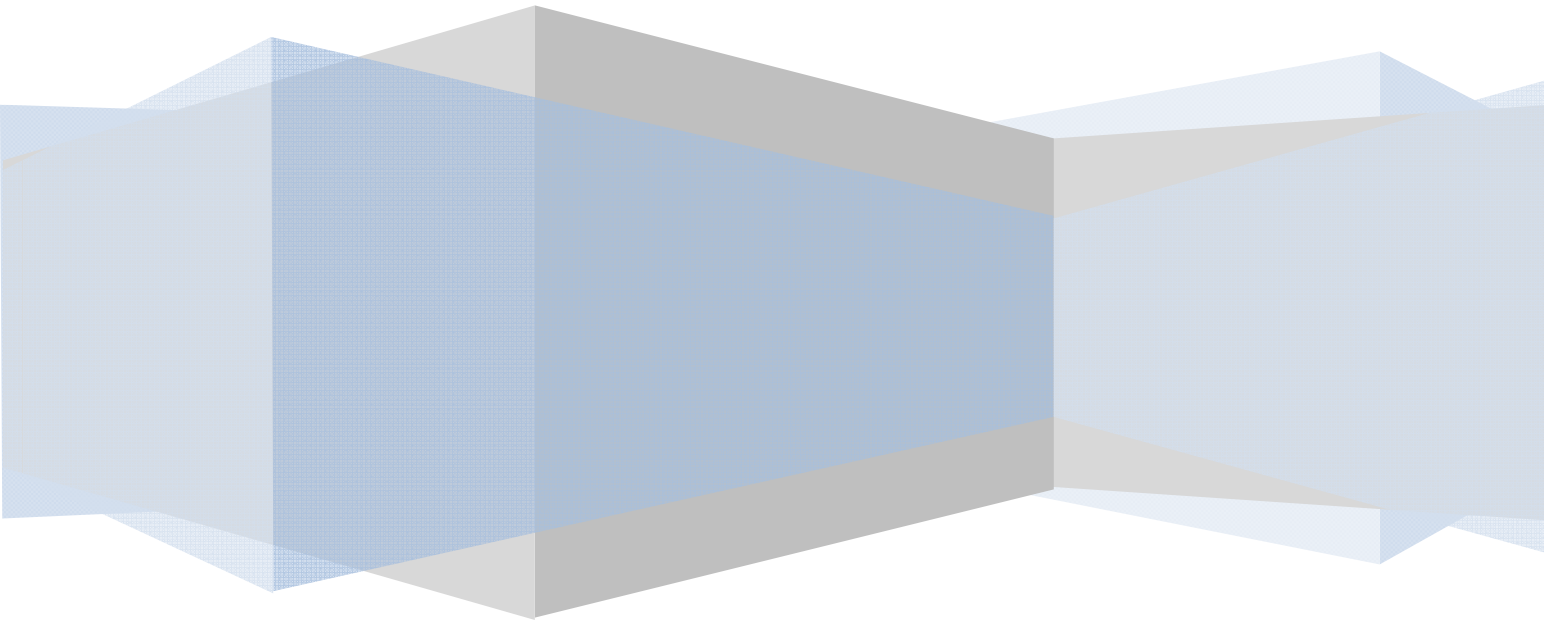
TECHNISCHE UNIVERSITÄT MÜNCHEN

JASS 2009

Multigrid

A Basic Introduction

Babak Gholami



Contents

Introduction

Multigrid methods, or more generally, Multilevel methods were originally developed to solve boundary value problems on spatial domains. However, this should be considered only as a start to these methods while today multigrid solvers are widely used for many different problems. Two very large and broadly used families of partial differential equations, Parabolic and Hyperbolic PDE's, are now treated using multigrid idea. These methods are applicable to problems where there is no actual grid (Algebraic Multigrid). Moreover, in optimization problems and solving integral equations one can see methods based on this idea.

Today problems of higher dimensions, in large scales and with very high accuracy demands have left us with no choice other than parallelization. One very pleasant feature of multigrid methods is that they can be efficiently parallelized. A Multilevel solver, with the right setup, is optimal; still being able to implement such a method in parallel is very important.

Going further one can name nonlinear problems as another type of the problems treatable with multigrid methods. Nonlinear problems, because of their iterative nature, should be treatable with multigrid. We will start developing multigrid idea for linear problems at first. Since there is a significant difference between linear and nonlinear system of equations, one, for sure, cannot use the same setup for nonlinear problems as linear ones. There will be more details on this. As the last point it is worth mentioning that multigrid is also suitable for treating the local demands. This is called Multilevel Adaptive Methods.

In studying multigrid it is of crucial importance to consider it as an entire approach to computational problem solving. This means multigrid idea is not only not limited to some specific family of problems, but also is applicable to domains with arbitrary geometries and boundary conditions. In this way, one can say that multigrid is a collection of ideas.

Model Problems

As mentioned above, multigrid methods were originally applied to simple boundary value problems. For the reason of simplicity, we do the same here. Equation (1) shows the model problem:

$$-\nabla^2 u + \sigma u = f(x, y, z, \ldots), \qquad 0 < x, y, z, \ldots < 1, \quad \sigma \geq 0 \tag{1}$$

Boundary conditions are set to zero. Here is a discretization of the model problem in two dimensions:

$$\begin{aligned} &\frac{-v_{i-1,j} + 2v_{ij} - v_{i+1,j}}{h_x^{\,2}} + \frac{-v_{i,j-1} + 2v_{ij} - v_{i,j+!}}{h_y^{\,2}} + \sigma v_{ij} = f_{ij}, \\ &v_{i0} = v_{in} = v_{0j} = v_{mj} = 0, \qquad 1 \leq i \leq m-1, \quad 1 \leq j \leq n-1. \end{aligned} \tag{2}$$

Equation (2) can be shown more compactly like a system of equations:

$$A\mathrm{v} = \mathrm{f} \tag{3}$$

Relaxation methods and direct methods are the two large categories for solving system of equations in (3).

Direct methods determine a solution up to the machine's precision in a finite number of arithmetic steps. Gaussian Elimination is a prototype of them. But here we are interested in relaxation methods that are discussed in the next chapter.

Basic Iterative Methods

Denoting the exact solution by "**u**", error by "**e**" and residual by "**r**", the residual equation is as follows:

$$\mathrm{r} = \mathrm{f} - A\mathrm{v} \tag{4}$$

This directly results in:

$$A\mathrm{e} = \mathrm{r} \tag{5}$$

Equation (5) says the error will satisfy the same set of equations as the unknown when the right hand side is replaced by the residual. This will play an important role in setting up a multigrid scheme for linear problems.

A very simple and familiar method to solve (3) is Jacobi. It solves the j^{th} equation of system of equations for the j^{th} unknown using current approximation for all other variables. A modification to it introduces an entire family of iterations called weighted Jacobi. Here, the matrix form of the weighted (damped) Jacobi, shows how the new approximation is computed using the old one (ω is the relaxation factor):

$$\mathrm{v}^{(1)} = \mathrm{v}^{(0)} + \omega D^{-1}\mathrm{r}^{(0)}. \tag{6}$$

One other famous relaxation method is called Gauss-Seidel. In Gauss-Seidel the components of the new approximation are used as soon as they are computed. This way the pattern by which the domain is updated makes difference. In simple GS the components are updated in ascending order. An effective alternative to this is called Red-Black Gauss-Seidel in which the components are updated in two groups; for example first the even-numbered grid points and then the odd-numbered ones. It is interesting to know that this scheme is very well suited for parallel computation.

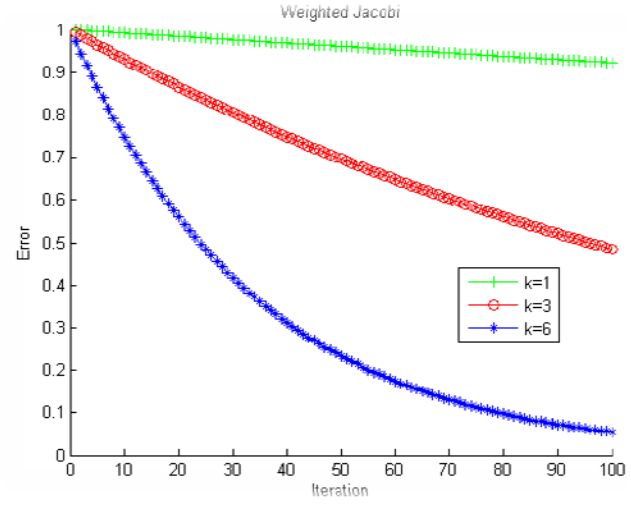
In order to investigate the effectiveness of these three introduced methods we study the performance of the methods in solving homogeneous system of linear equations of (7) with the initial guess of form (8).

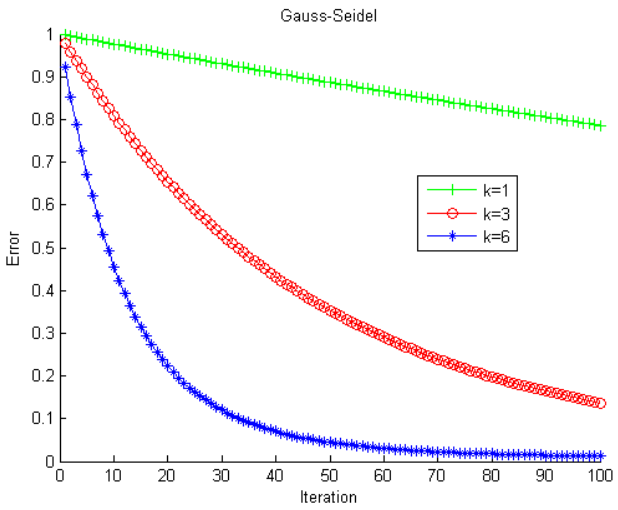
$$A\mathrm{v} = 0 \tag{7}$$

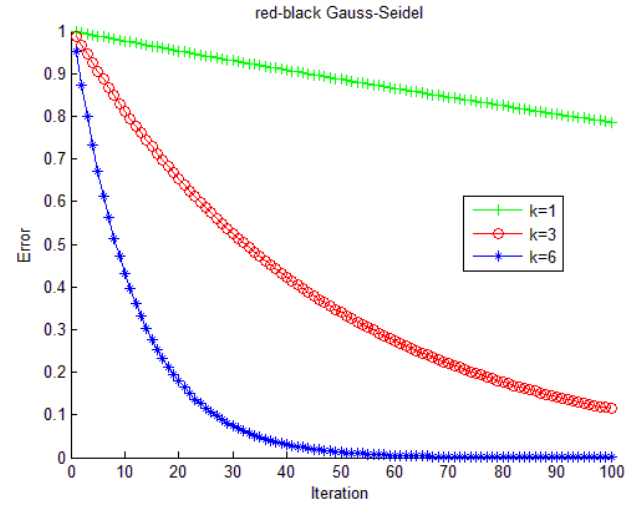
$$v_j = \sin\left(\frac{jk\pi}{n}\right), \qquad 0 \le j \le n, \quad 1 \le k \le n-1. \tag{8}$$

"*k*" is called the wave number and is the number of half sine waves in the domain. In simple words, the lower the wave number is the smoother is the wave that constitutes the domain.

The results for solving the one-dimensional model problem in a grid with 64 points and three different initial guesses (*k*=1, 3 and 6) are shown below in figure (1):

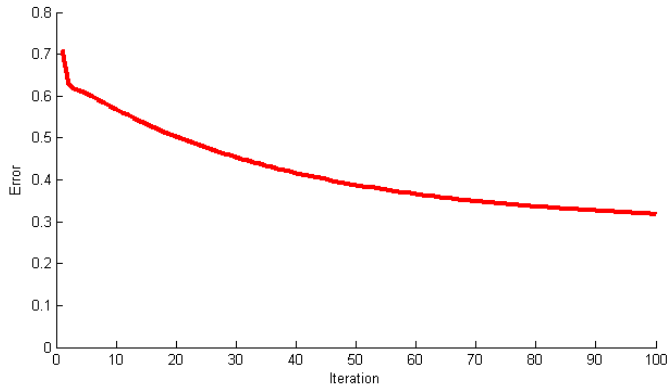






Figure(1). Results of performing 100 iterations on the one-dimensional model problem with weighted Jacobi, Gauss-Seidel and Red-black Gauss-Seidel. The norm of the error is plotted against the number of iterations.

The qualitative behavior indicates that for all methods the convergence is highly dependent on the wave number; waves with higher frequencies are damped more efficiently. In contrast, smooth modes are being damped so slowly. This is called “smoothing property”. Here a more realistic situation is shown where the initial condition contains more than one frequency:



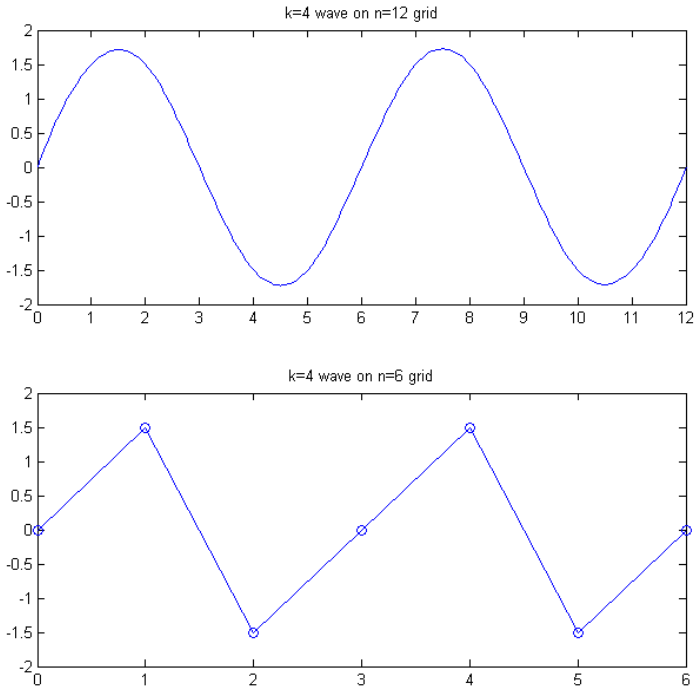
Figure(2). Performing 100 iterations of weighted Jacobi on a wave with three modes (k=1, 6 and 32).

Figure (2) clearly shows the weakness of iterative methods; they damp the components of higher frequency very efficiently but experience a dramatic reduction in convergence speed afterwards. In fact most of the basic iterative methods possess the smoothing property. Here is where the multigrid enters!

Elements of Multigrid

We concluded in the last chapter that we more ideas to add to the iterative schemes in order to get a powerful and efficient solver. One idea is to take advantage of a good initial guess. A good initial guess can be obtained by performing some iteration on a coarse grid. On the one hand, this is cheaper due to the fewer number of unknowns. On the other hand, the convergence behavior is more ideal on a coarse grid since it improves with the increase of the grid size. The idea can be used in a better way; iterating on the coarsest grid to obtain an initial guess for on finer grid and now iterating on this grid to get a guess for another finer grid and so on. This is the basis of a strategy called "nested iteration".

A second idea is also worth mentioning. Below, in the figure (3), a wave in two grids is shown. On the top, the grid has 12 grid points. At the bottom the same wave is represented on a coarser grid.



Figure(3). A wave with k=4 represented on a fine and a coarse grid.

We can say that the wave in the coarse grid seems to have the higher frequency. This is stated more precisely in below:

$$w_{k,2j}^{h} = \sin\left(\frac{2jk\pi}{n}\right) = \sin\left(\frac{jk\pi}{n/2}\right) = w_{k,j}^{2h}, \qquad 1 \le k < \frac{n}{2} \tag{9}$$

Equation (9) shows that a wave with wave number k on the fine grid has the same wave number on the coarse grid (provided that k is within the mentioned range). Knowing that there are half as many modes on the coarse grid as there are on the fine grid, the statement is confirmed. This suggests that when relaxation begins to stall, move to a coarser grid. This is the basis of a strategy called "correction scheme".

Both strategies mentioned need to move between grids of different size. Moving from a fine grid to a coarser one is called restriction. One obvious operator is direct injection which only uses the grid points that are also in the coarse grid. There are, of course, many other methods to perform this. An example is full weighting; but what is of great importance is that one should come to a compromise in choosing among these operators since there is no guarantee to, necessarily, get a better result (say convergence rate) by using a higher order operator. This also applies to the other intergrid transfer family, namely interpolation, which is transferring information from a coarse grid to a fine one. Interpolation (linear interpolation is an example) is mostly effective when the error is smooth, so multigrid schemes should also deal with this difficulty. We will see that these schemes provide a complement which is effective even when the error is oscillatory. Now we have enough means to introduce one simple multigrid scheme which is a basis for many other ones: Two-Grid Correction Scheme. Here the algorithm is described (To associate a grid to a vector or matrix its superscript represents its grid size):

$$\mathbf{v}^h \leftarrow MG\left(\mathbf{v}^h, \mathbf{f}^h\right)$$

1. Relax v_1 times on $A^h\mathbf{u}^h=\mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$.

2. Compute the fine-grid residual $\mathbf{r}^h=\mathbf{f}^h-A^h\mathbf{v}^h$ and restrict it to the coarse grid.

3. Solve $A^{2h}\mathbf{e}^{2h}=\mathbf{r}^{2h}$ on Ω^{2h}.

4. Interpolate the coarse-grid error and correct the fine grid approximation by $\mathbf{v}^h \leftarrow \mathbf{v}^h+\mathbf{e}^h$.

5. Relax v_2 times on $A^h\mathbf{u}^h=\mathbf{f}^h$ on Ω^h with initial guess $\mathbf{v}^h$.

It is important that Two-Grid scheme be understood completely because other schemes introduced here are natural extensions of it. Based on it a more advanced scheme can be obtained. This scheme is called V-cycle; a short look at figure (4) explains it. V-cycle is the recursive use of the Two-Grid scheme. Even more complicated schemes can be derived from the V-cycle itself, so if one fails to understand the core idea behind the simple Two-Grid scheme it is probable that he or she faces lots of difficulties in developing more advanced schemes. Having this on mind, we move on to the algorithm description for the V-cycle.

$$\mathbf{v}^h \leftarrow V^h\left(\mathbf{v}^h, \mathbf{f}^h\right)$$

1. Relax v_1 times on $A^h\mathbf{u}^h=\mathbf{f}^h$ on Ω^h with a given initial guess .

2. If Ω^{2h} = coarsest grid then go to step 4.

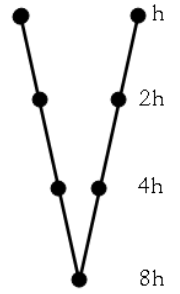
   else $\mathbf{f}^{2h} \leftarrow I_h^{2h}\left(\mathbf{f}^h - A^h\mathbf{v}^h\right)$ (I_h^{2h} is restriction)

   $\mathbf{v}^{2h} \leftarrow 0$

   $\mathbf{v}^{2h} \leftarrow V^h\left(\mathbf{v}^{2h}, \mathbf{f}^{2h}\right)$

3. Correct $\mathbf{v}^h \leftarrow \mathbf{v}^h + I_{2h}^h\mathbf{v}^{2h}$

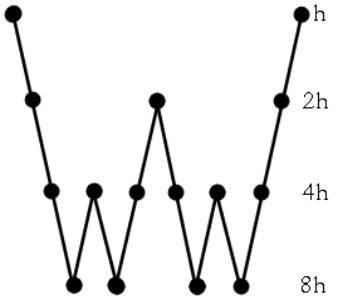
4. Relax v_2 times on $A^h\mathbf{u}^h=\mathbf{f}^h$ on Ω^h.



Figure(4).
V-cycle schedule

V-cycle is a member of a more general family of multigrid cycles. The only difference in the algorithm is that the recursive call will not happen only once (as in V-cycle) but μ times. W-cycle (shown in figure (5)) is the case when $\mu = 2$.

So far we only used the correction scheme idea. It is possible to still introduce more efficient schemes by employing the nested iteration idea and combining it with the existing schemes. The result is FMG (figure (6)).



Figure(5). W-cycle schedule

$$\mathbf{v}^h \leftarrow FMG^h\left(\mathbf{f}^h\right)$$

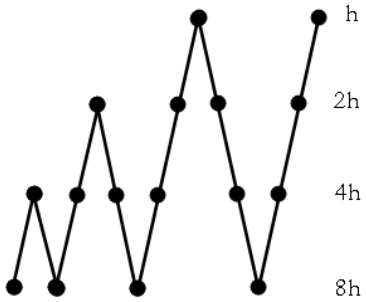
1. If Ω^{2h} = coarsest grid set $\mathbf{v}^h \leftarrow 0$ and go to step 3.

   else $\mathbf{f}^{2h} \leftarrow I_h^{2h}\left(\mathbf{f}^h\right)$

   $\mathbf{v}^{2h} \leftarrow FMG^{2h}\left(\mathbf{f}^{2h}\right)$

2. Correct $\mathbf{v}^h \leftarrow I_{2h}^h\mathbf{v}^{2h}$

3. $\mathbf{v}^h \leftarrow V^h\left(\mathbf{v}^h, \mathbf{f}^h\right)$ v_0 times.



Figure(6). FMG schedule

One concern can be the extra effort we do for obtaining a good initial guess. In fact Full Multgrid is not only inexpensive, but easily pays for itself.

Nonlinear Problem

Up to now we developed a few schemes suitable for solving linear problems; but what about nonlinear problems? Do we need to make any changes in what we have until now? In order to answer this let's take a look at the most significant difference between linear and nonlinear systems. Consider the nonlinear system of equations below:

$$A(\mathrm{u}) = \mathrm{f} \tag{10}$$

The definition of residual gives:

$$A(\mathrm{u}) - A(\mathrm{v}) = \mathrm{r}. \tag{11}$$

Up to this stage everything seems the same. Even though the error definition is also the same we cannot conclude (12):

$$A(\mathrm{u}) - A(\mathrm{v}) = A(\mathrm{e}). \tag{12}$$

Apparently, a new scheme has to be developed. The new residual equation can be written as in (13):

$$A(\mathrm{v} + \mathrm{e}) - A(\mathrm{v}) = \mathrm{r}. \tag{13}$$

By expanding the first term in Taylor series about v and truncating the series after two terms, we have a linear equation (J is the Jacobi matrix):

$$J(\mathrm{v})\mathrm{e} = \mathrm{r} \tag{14}$$

This linear system is an approximation to the nonlinear system of (10) and, of course, is highly recommended to be solved by one of the schemes developed before. This scheme is called Newton-Multigrid. Despite its being effective we need to develop another method since Newton-Multigrid does not use the idea of multigrid directly to treat the nonlinearity.

In a two-grid setting, the residual on the coarser grid appears as:

$$A^{2h}\left(\mathrm{v}^{2h} + \mathrm{e}^{2h}\right) - A^{2h}\left(\mathrm{v}^{2h}\right) = \mathrm{r}^{2h}. \tag{15}$$

The coarse-grid residual can be the restriction of the fine-grid residual:

$$\mathrm{r}^{2h} = I_h^{2h}\mathrm{r}^h = I_h^{2h}\left(\mathrm{f}^h - A^h\left(\mathrm{v}^h\right)\right). \tag{16}$$

The current approximation on the coarse grid will also be obtained by restricting the fine grid approximation with the same operator:

$$\mathrm{v}^{2h} = I_h^{2h}\mathrm{v}^h. \tag{17}$$

By solving the resulting nonlinear equation (by means of a nonlinear solver, Non-linear Gauss-Seidel is a good example) on the coarse grid the error on this grid is calculated and can be interpolated to the fine grid to correct the fine grid approximation. This scheme is called Fully Approximation Scheme (FAS).

This two-step FAS is also a basis for more complicated methods. Analogously, a more useful FAS should be implemented recursively, so variations like V-cycles and W-cycles are helpful. One last comment is on the use of a good initial guess. It is important to know that the convergence of nonlinear iterations depends critically on a good initial guess. As a result, using one FMG cycle can provide accuracy to the level of discretization.

References

[1] William L. Briggs, Van Emden Henson, and Steve F. McCormick, *A Multigrid Tutorial, Second Edition*, SIAM, 2000.

[2] Pieter Wesseling, *An Introduction to Multigrid Methods,* Edwards, 2004.

[3] Vahid Isfahanian, *Computational Fluid Dynamics I*, University of Tehran, 2001.