Chapter 5

Multigrid Methods

5.1 Core Ideas

Linear multigrid methods are an important class of algorithms to iteratively approximate linear systems of equations of the form

$$Au = f \tag{5.1}$$

where $A \in \mathbb{R}^{m \times m}$ is usually a sparse symmetric positive definite matrix, that arises from the discretisation of a differential equation on a discretised domain Ω_h. They were first developed in the early 1960s by the soviet mathematicians Fedorenko and Bachlwalow. They then became more well-known and further established in the late 1970s especially through the works of Hackbusch [26] and Brandt [6].

Remark: Let u now denote the solution of a discretised problem, which has in previous chapters been called u^h. As from now on we will only be dealing with finite dimensional approximations, we will for brevity drop the superscript h. From now on we will be dealing more with iterative solutions and denote them by a superscript k.

There are different ways to motivate the construction of multigrid algorithms. One of them is to start from an iteration built on approximating solutions of the residual equation and is *based on* [27]. That is for any approximation u^k of (5.1), we denote the error e^k, the residual or defect r^k and the overall iteration called defect equation by

$$e^k := u - u^k \tag{5.2}$$

$$r^k := f - Au^k \tag{5.3}$$

$$Ae^k = r^k \tag{5.4}$$

For now we have only rewritten equation (5.1), subtracting the term Au^k on either side, as $Ae^k = A(u - u^k) = f - Au^k$. However if A is approximated by a "simpler" invertible operator $\hat{A}$, then the solution of

$$\hat{A}c^k = r^k \tag{5.5}$$

gives us a new approximation

$$u^{k+1} := u^k + c^k. \tag{5.6}$$

That is given the current iterate u^k, we compute the residual r^k, solve the simpler system (5.5) for the correction and can hence update the new iterate u^{k+1}. If we start from some initial iterate u^0, the consecutive application leads to an iterative procedure, with the iteration operator M and the error iteration

$$M = I_h - (\hat{A})^{-1}A \tag{5.7}$$

$$e^{k+1} = Me^k \tag{5.8}$$

where I_h stands for the identity operator. Hence the error reduction can be bounded from above by $\rho := \|M\|$ for one iteration, where it is worth noting that we can only be sure to converge if $\rho < 1$. The question that arises now is how to construct an operator $\hat{A}$, that leads to a value of ρ that is as small as possible and still simple to invert.

One possibility to solve the defect equation approximately is to construct an approximation $A_H : \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_H)$ *need to describe* $\mathcal{G}(\Omega_H)$*?* of A on a coarser grid Ω_H, where we assume that A_H is invertible and $\dim(\mathcal{G}(\Omega_H)) < \dim(\mathcal{G}(\Omega_h))$ and where one now attempts to solve

$$A_H e_H^k = r_H^k \qquad \text{with} \tag{5.9}$$

$$r_H^k := I_h^H r^k \qquad \text{and} \tag{5.10}$$

$$e_h^k := I_H^h e_H^k. \tag{5.11}$$

The restriction operator $I_h^H : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_H)$ is used to restrict the fine grid residual to the coarse grid while the interpolation operator $I_H^h : \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_h)$ is used to interpolate the correction $\hat{e}^k$ back to the fine grid from the coarse grid. One coarse grid correction step i.e. calculating u^{k+1} from u^k, then entails the computation of the fine grid residual r^k using u^k, restricting it to the coarse grid, obtaining r_c^k, solving the coarse grid problem (5.9), interpolating the solution back to the fine grid and computing the new approximation u^{k+1} using (5.6). The corresponding iteration operator that arises is $(I_h - I_h^H A_c^{-1} I_H^h A)$, unfortunately $\hat{A} = I_h^H A_c^{-1} I_H^h$ is not invertible as I_h^H maps into a lower dimensional space and therefore we have that $(\hat{A})Ae = 0$ for certain $e \neq 0$. Which means that this error function e will not be changed by a coarse grid correction step and therefore cannot give rise to a converging iteration. In order to succomb this problem one can combine the process of coarse grid correction with a classical iteration method also referred to as smoothers or relaxation schemes in this context. They are usually simple (block-) Jacobi or Gauss-Seidel iterations that are computationally inexpensive and have the property of indeed showing a *notable* smoothing effect, for which a concrete numerical showcase can be found in section (6.2.1). Thus let us denote the following relaxation scheme on the error e^k

$$e^{k+1} = e^k + P(f - Ae^k) \quad \text{with the iteration matrix} \tag{5.12}$$

$$S = (I_h - PA) \tag{5.13}$$

where P describes the preconditioner and would for example in the basic Jacobi case be a diagonal matrix with $(\frac{1}{a_{11}}, \frac{1}{a_{22}}, ..., \frac{1}{a_{mm}})$ on the diagonal. If we now combine the two processes in an iteration step or cycle consisting of presmoothing, coarse grid correction and postsmoothing, we obtain the following iteration operator for a two-grid method

$$M = S^{\nu_2} K S^{\nu_1} \qquad \text{with } K := I_h - I_h^H A_H^{-1} I_H^h A \tag{5.14}$$

where ν_1 denotes the number of presmoothing steps and ν_2 the number of postsmoothing steps. A way to *describe/imagine* the interplay of the coarse grid correction and the smoothing that works *especially* well for *elliptic* problems is to consider a decomposition of the error e into eigenmodes which correspond to discrete Fourier modes of different frequencies. Due to the structure of the matrix of discretised differential equations we only have a local transport of information, therefore simple iterative solvers such as a Jacobi or Gauss-Seidel method damp high frequency components of the error fastest, which leads to a much smoother, lower frequency error after only a few iterations. Now one takes advantage of the fact that the low frequency error on the fine grid becomes a higher frequency error on the coarse grid, especially if one considers recursively applying this approach, that is using more than 2 levels, and can therefore be damped there more efficiently also applying a smoother there. That is the usage of coarser grids accelarates the transport of low frequency error across the discretised domain. Additionally the coarser grids have significantly less degrees of freedom which makes them computationally much cheaper to handle, until we can solve the problem directly on the coarsest grid. The key is to then restrict and interpolate effectively between the series of nested grids to approximate the error of the current and the actual solution. There is a wide field of reasearch on how to choose the appropriate operators for which classes of problems, where for a further discussion we can refer to [source]. The choices made in our implementation will be presented in further detail in the following chapter.

With suitable combinations of coarse grid correction and smoothing one can achieve fast, mesh-size independent convergence rates for a variety of problems [source]. *convergence .... what to say here?*

There are different ways of how to construct the coarse level spaces. In geometric multigrid we have a predefined sequence of nested meshes of specific coarsening factors that can vary in different directions and on different levels. Nevertheless the coarse level spaces still represent the original problem at a lower resolution. In algebraic multigrid on the other hand one does not have predefined coarse level spaces but instead they are chosen according to some rules, that is one constructs the operators directly from the system matrix. In classical AMG, the levels of the hierarchy are simply subsets of unknowns without any geometric interpretation. This can be favorable in instance where we have a irregular meshes Ω_h or a very particular structure ... *what else can I say here? ...* but also more expensive to compute.

5.2 Basic Algorithm

In this section we present a classical multigrid V–cycle, which is based on [22]. We need the following main ingredients to construct a multigrid method with $l+1$ levels, where $l+1$ represents the finest grid; a smoother S, a set of interpolation and restriction matrices $I^l_{l-1}, I^{l-1}_{l-2}, ...I^0_1$ and $I^{l-1}_l, I^{l-2}_{l-1}, ...I^1_0$, represenations of the matrix A on all levels, that is $A^l, A^{l-1}, A^{l-2}, ..., A^0$ with $A^l = A$, an initial guess s_{init}. These we can then put together in a multigrid V-cycle who looks as follows.

Multigrid V-cycle

Let w^J be the initial guess (on the finest grid level). Then repeat the following until convergence criterium is met or number of iterations exceeds a certain threshold:

```
• do $\nu_{J_a}$ smoothing steps on $A^J u^J = f^J$ with initial guess $w^J$
• compute $f^{J-1} = I^{J-1}_J r^J$ with $r^J = f^J - A^J u^J$
    – do $\nu_1$ smoothing steps on $A^{J-1} u^{J-1} = f^{J-1}$ with initial guess $w^{J-1} = 0$
    – compute $f^{J-2} = I^{J-2}_{J-1} r^{J-1}$ with $r^{J-1} = f^{J-1} - A^{J-1} u^{J-1}$
        ∗ do $\nu_1$ smoothing steps on $A^{J-2} u^{J-2} = r^{J-2}$ with initial guess $w^{J-2} = 0$
        ∗ compute $f^{J-3} = I^{J-3}_{J-2} r^{J-2}$
          ...
          ...
            · solve $A^0 u^0 = f^0$
          ...
          ...
        * correct $w^{J-2} = w^{J-2} + I_{J-3}^{J-2} w^{J-3}$
        * do $\nu_2$ smoothing steps on $A^{J-2} u^{J-2} = f^{J-2}$ with initial guess $w^{J-2}$
    – correct $w^{J-1} = w^{J-1} + I_{J-2}^{J-1} w^{J-2}$
    – do $\nu_2$ smoothing steps on $A^{J-1} u^{J-1} = f^{J-1}$ with initial guess $w^{J-1}$
• correct $w^J = w^J + I_{J-1}^{J} w^{J-1}$
• do $\nu_2$ smoothing steps on $A^J u^J = f^J$ with initial guess $w^J$
```