Multigrid Methods

Lecturer notes for NMNV571, Faculty of Mathematics and Physics, Charles University

Jan Papez

text under construction, version of April 3, 2025

Contents

Introduction

These notes are based on my lectures for the course Multigrid Methods (NMNV571) at the Faculty of Mathematics and Physics, Charles University. As lecture notes, they are still a work in progress, with the goal of continuously improving the presentation of the material for students.

The course content is currently divided into three parts:

1. A thorough introduction to multigrid and its key principles (following [Briggs et al., 2000]),
2. An abstract formulation of multigrid, including convergence results and sufficient conditions for convergence (adapted from [Shaidurov, 1995]),
3. Stable splitting as an advanced topic (following [Rüde, 1993]).

The third part may be adjusted based on students' preferences. For instance, topics such as algebraic multigrid or multigrid for nonlinear problems could be considered.

As of February 2025, my aim is to cover at least the first two topics in these notes.

Chapter 1

Key principles of the multigrid

We begin with a detailed and gradual introduction to the key principles of multigrid. For clarity, we illustrate these concepts using a simple model problem—a Poisson equation on a basic domain—discretized with finite differences.

Our presentation follows [Briggs et al., 2000], where additional details and numerical experiments can be found. However, there is a key distinction: in much of the literature (including [Briggs et al., 2000]), multigrid is often described as an enhancement of stationary iterative methods. I find this perspective overly simplistic and somewhat misleading. Instead, we present multigrid as the combination of two complementary techniques that together yield a truly efficient solver or preconditioner.

1.1 Error, Residual, Iterative Refinement

Consider solving

$$\mathsf{A}\mathsf{u} = \mathsf{f},$$

where A is a non-singular matrix, f is the right-hand side, and u is the exact solution. We will denote an approximate solution either by v (if it is arbitrary) or by $\mathsf{u}^{(i)}$ (if we want to emphasize an iterative procedure).

Error or the *algebraic error*, is the key vector, $\mathsf{e} = \mathsf{u} - \mathsf{v}$, or $\mathsf{e}^{(i)} = \mathsf{u} - \mathsf{u}^{(i)}$. Given an approximation v with the error e, we have

$$\mathsf{v} + \mathsf{e} = \mathsf{v} + \mathsf{u} - \mathsf{v} = \mathsf{u},$$

so that we can compute the exact solution.

Residual is the computable vector given by $\mathsf{r} = \mathsf{f} - \mathsf{A}\mathsf{v}$, or $\mathsf{r}^{(i)} = \mathsf{f} - \mathsf{A}\mathsf{u}^{(i)}$. There holds

$$\mathsf{r} = \mathsf{f} - \mathsf{A}\mathsf{v} = \mathsf{A}\mathsf{u} - \mathsf{A}\mathsf{v} = \mathsf{A}(\mathsf{u} - \mathsf{v}) = \mathsf{A}\mathsf{e}.$$

From $\mathsf{e} = \mathsf{A}^{-1}\mathsf{r}$ and $\mathsf{f} = \mathsf{A}\mathsf{u}$ we have

$$\|\mathsf{e}\| \le \|\mathsf{A}^{-1}\|\|\mathsf{r}\|, \qquad \|\mathsf{f}\| \le \|\mathsf{A}\|\|\mathsf{u}\|$$

and altogether

$$\frac{\|\mathsf{e}\|}{\|\mathsf{u}\|} \le \frac{\|\mathsf{A}\|\|\mathsf{A}^{-1}\|\|\mathsf{r}\|}{\|\mathsf{f}\|} = \kappa(\mathsf{A})\frac{\|\mathsf{r}\|}{\|\mathsf{f}\|},$$

where $\kappa(\mathsf{A}) = \|\mathsf{A}\|\|\mathsf{A}^{-1}\|$ is the condition number of A. Therefore $\mathsf{r} = 0$ if and only if $\mathsf{v} = \mathsf{u}$ but a small residual norm need not guarantee a small norm of the error if $\kappa(\mathsf{A})$ is not small.

Iterative refinement is the following procedure for a given approximation $\mathsf{u}^{(i)}$

1. Compute the residual $\mathsf{r}^{(i)} = \mathsf{f} - \mathsf{A}\mathsf{u}^{(i)}$.
2. Solve $\mathsf{A}\mathsf{e}^{(i)} = \mathsf{r}^{(i)}$ approximately, giving $\mathsf{d}^{(i)}$.
3. Define a new approximation $\mathsf{u}^{(i+1)} = \mathsf{u}^{(i)} + \mathsf{d}^{(i)}$.

If the solution in 2. can be written as

$$\mathsf{d}^{(i)} = \mathsf{M}^{-1}\mathsf{r}^{(i)},$$

with some $\mathsf{M} \approx \mathsf{A}$, then

$$\mathsf{u}^{(i+1)} = \mathsf{u}^{(i)} + \mathsf{M}^{-1}\mathsf{r}^{(i)} = \mathsf{u}^{(i)} + \mathsf{M}^{-1}(\mathsf{f} - \mathsf{A}\mathsf{u}^{(i)})$$

and for the error it holds

$$\mathsf{e}^{(i+1)} = \mathsf{u} - \mathsf{u}^{(i+1)} = \mathsf{u} - \mathsf{u}^{(i)} - \mathsf{M}^{-1}(\mathsf{f} - \mathsf{A}\mathsf{u}^{(i)}) = (\mathsf{I} - \mathsf{M}^{-1}\mathsf{A})\mathsf{e}^{(i)},$$

where I denotes the identity matrix (of the proper size).

If M is independent of i, $\mathsf{u}^{(i)}$, $\mathsf{r}^{(i)}$, the above iterative scheme is called *linear* or *stationary*. Examples of such schemes are Jacobi, Gauss–Seidel, SOR or Richardson methods.

There are also nonlinear iterative methods, for example Krylov subspace methods. Nevertheless, in (standard) preconditioned Krylov methods, a preconditioned residual is computed as $\mathsf{z}^{(i)} = \mathsf{M}^{-1}\mathsf{r}^{(i)}$ in each iteration.

1.2 Model Problem and its Discretization

Consider the homogeneous Poisson problem

$$-\Delta u = f \qquad \text{in } \Omega = (0,1)^d, \tag{1.1}$$

$$u = 0 \qquad \text{on } \partial\Omega. \tag{1.2}$$

We will often consider $d = 1$ for illustrations and $d = 2$ for computations. The true power of multigrid (and multilevel methods in general) comes, in particular, with 3D problems.

The boundary conditions (1.2) can be generalized as well as reaction term σu can be added to the left-hand side of (1.1). The domain can be more complex but not too much; this often remains one of the limitations for an efficient (geometric) multigrid.

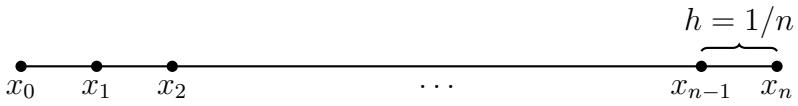


Figure 1.1: Uniform partitioning in 1D

1.2.1 Discretization in 1D

Consider the uniform partitioning of the interval $(0, 1)$ with n intervals and $n + 1$ nodes

$$x_j = jh = j/n, \quad j = 0, 1, \ldots, n.$$

Denote by v_j an approximation to value of the exact solution u in the node x_j, $v_j \approx u(x_j)$ and consider the vector of unknowns

$$\mathsf{v} = [v_1, v_2, \ldots, v_{n-1}]^T.$$

We replace the second derivative u'' in (1.1) by the central difference,

$$-u''(x_j) \approx \frac{-u(x_{j-1}) + 2u(x_j) - u(x_{j+1})}{h^2},$$

giving the system of linear equations

$$\begin{aligned} v_0 = v_n &= 0, \\ \frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} &= f(x_j) \qquad j = 1, 2, \ldots n-1. \end{aligned} \tag{1.3}$$

When denoting $f_j = f(x_j)$, $j = 1, \ldots, n-1$ and $\mathsf{f} = [f_1, \ldots, f_{n-1}]^T$, we can write the matrix form of (1.3)

$$\frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 + h^{-2} v_0 \\ f_2 \\ \vdots \\ f_{n-1} + h^{-2} v_n \end{bmatrix}. \tag{1.4}$$

1.2.2 Discretization in 2D

Consider the uniform partitioning of the interval $(0, 1)$ with $m + 1$ nodes

$$x_i = ih_x = i/m, \quad i = 0, 1, \ldots, m.$$

and $n + 1$ nodes for the y-direction

$$y_j = jh_y = j/n, \quad j = 0, 1, \ldots, n.$$

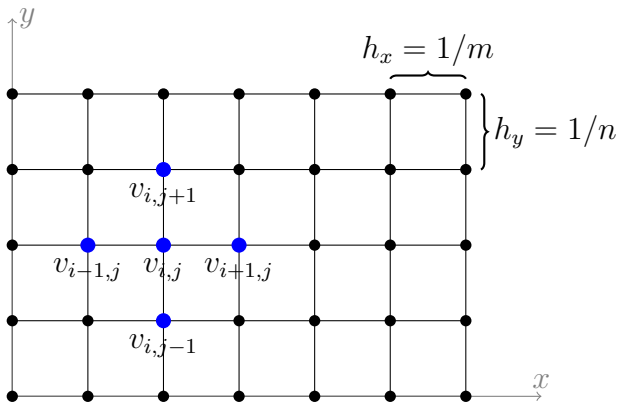


Figure 1.2: 2D partitioning with illustration of values for 2D central differences

This gives us $(m+1) \times (n+1)$ nodes of the mesh (x_i, y_j), $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, n$. Denote by $v_{i,j}$ an approximation to value of the exact solution u in the node (x_i, y_j),

$$v_{i,j} \approx u(x_i, y_j), \qquad i = 0, 1, \ldots, m, \;\; j = 0, 1, \ldots, n.$$

Recall, that the model problem (1.1) in 2D reads

$$-\Delta u = -\frac{\partial^2}{\partial x^2} u - \frac{\partial^2}{\partial y^2} u = f.$$

Replacing the second derivatives $\partial^2 u / \partial x^2$ and $\partial^2 u / \partial y^2$ by central differences, as in 1D, gives

$$\frac{-v_{i-1,j} + 2v_{i,j} - v_{i+1,j}}{h_x^2} + \frac{-v_{i,j-1} + 2v_{i,j} - v_{i,j+1}}{h_y^2} = f_{i,j} = f(x_i, y_j),$$
$$1 \le i \le m-1, \quad 1 \le j \le n-1, \quad (1.5)$$

with homogeneous boundary conditions represented by

$$v_{0,j} = v_{m,j} = v_{i,0} = v_{i,n} = 0, \qquad 0 \le i \le m, \quad 0 \le j \le n. \quad (1.6)$$

The system has $(m-1) \times (n-1)$ unknowns that can be written into a vector "row-" or "column-wise". We will consider the later one. Denote

$$\mathsf{v}_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,n-1}]^T$$
$$\mathsf{f}_i = [f_{i,1}, f_{i,2}, \ldots, f_{i,n-1}]^T, \qquad 1 \le i \le m-1,$$

and write (1.5) (now already with boundary conditions (1.6)) as

$$\begin{bmatrix} \mathsf{B} & -a\mathsf{I} & & \\ -a\mathsf{I} & \mathsf{B} & -a\mathsf{I} & \\ & \ddots & \ddots & \ddots \\ & & -a\mathsf{I} & \mathsf{B} \end{bmatrix} \begin{bmatrix} \mathsf{v}_1 \\ \mathsf{v}_2 \\ \vdots \\ \mathsf{v}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathsf{f}_1 \\ \mathsf{f}_2 \\ \vdots \\ \mathsf{f}_{n-1} \end{bmatrix}. \quad (1.7)$$

Here $a\mathsf{I} = h_x^{-2}\mathsf{I} \in \mathbb{R}^{(m-1)\times(m-1)}$ and

$$\mathsf{B} = h_y^{-2}\begin{bmatrix} 2\left(\frac{h_x^2+h_y^2}{h_x^2}\right) & -1 & & \\ -1 & 2\left(\frac{h_x^2+h_y^2}{h_x^2}\right) & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2\left(\frac{h_x^2+h_y^2}{h_x^2}\right) \end{bmatrix}.$$

In the case with $h = h_x = h_y$, $\mathsf{B} = h^{-2}$ tridiag$(-1, 2, -1)$, which is the 1D Poisson matrix from (1.4).

Exercise. Generalize the finite-difference scheme to 3D.

1.3 Elements of the Multigrid

1.3.1 Coarse-space correction

Use of several grids (levels) is a key aspect of a multigrid. In our setting with finite-difference discretization, we understand by a coarser grid (or level) a subset of degrees of freedom (DOFs). In a *geometric multigrid*, the subset corresponds to an actual discretization grid with a different discretization parameter; in the previous section denoted by h. *Algebraic multigrids* are based on creating the grids artificially to obtain some (algebraic) properties.

The simplest case, for a presentation as well as an analysis, is a *two-grid* method. The problem is posed on the fine grid, while the coarse grid is used for improving the efficiency of solving the fine problem. Important assumption is that the problem on the coarse level is sufficiently smaller than the one on the fine level.

To introduce a two-grid method, we need

- (Discrete) problems on the fine and coarser levels. These are represented[1] by the matrices A_1 and A_0. We also need the right-hand side f_1 for the fine-level problem.
- The operators between the levels (intergrid operators):

  interpolation I_0^1 from the coarse-grid vectors to fine-grid ones; "from a short vector to the long one"

  restriction R_1^0, "from a long vector to the short one"

Denoting by N_1 the number of degrees of freedom on the fine level and by N_0 the number of DOFs on the coarse levels, the operators (matrices) discussed above have the following dimensions

$$\mathsf{A}_1 : \mathbb{R}^{N_1} \to \mathbb{R}^{N_1}, \qquad \mathsf{A}_0 : \mathbb{R}^{N_0} \to \mathbb{R}^{N_0},$$
$$\mathsf{I}_0^1 : \mathbb{R}^{N_0} \to \mathbb{R}^{N_1}, \qquad \mathsf{R}_1^0 : \mathbb{R}^{N_1} \to \mathbb{R}^{N_0}.$$

[1] A common notation for two-grid methods uses subscripts "h" and "H". We will not follow this since the notation with numbers is straightforwardly applicable to the case with more levels.

We assume that $N_0 \ll N_1$.

Given an approximation $\mathsf{u}_1^{(i)}$ to the solution u_1 of the fine-level problem $\mathsf{A}_1\mathsf{u}_1 = \mathsf{f}_1$, compute the associated residual $\mathsf{r}_1^{(i)} = \mathsf{f}_1 - \mathsf{A}_1\mathsf{u}_1^{(i)}$. The idea of the *coarse-space correction* is to solve the problem with A_0 (corresponding to the coarse level) and this use to improve (correct) the approximation $\mathsf{u}_1^{(i)}$ on the fine level. Therefore, we solve

$$\mathsf{A}_0\mathsf{d}_0^{(i)} = \mathsf{R}_1^0\mathsf{r}_1^{(i)} \tag{1.8}$$

and hope that

$$\mathsf{I}_0^1\mathsf{d}_0^{(i)} \approx \mathsf{e}_1^{(i)} = \mathsf{u}_1 - \mathsf{u}_1^{(i)}.$$

In the notation from the previous section,

$$\mathsf{u}_1^{(i+1)} = \mathsf{u}_1^{(i)} + \mathsf{I}_0^1 \left(\mathsf{A}_0\right)^{-1} \mathsf{R}_1^0\mathsf{r}_1^{(i)}.$$

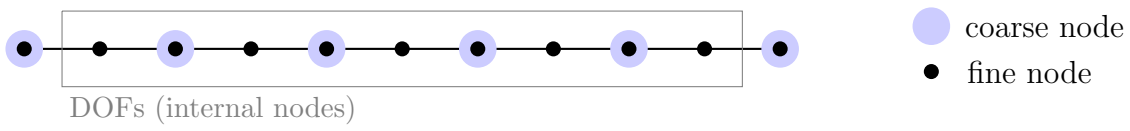
The analysis of multigrid methods typically assumes that (1.8) is solved exactly. We will also stick with this assumption for (most of) the course.

> Natural and key question arises, when does the coarse-space correction work? Vaguely, we need the problem $\mathsf{A}_0\mathsf{d}_0^{(i)} = \mathsf{R}_1^0\mathsf{r}_1^{(i)}$ to be close to $\mathsf{A}_1\mathsf{d}_1^{(i)} = \mathsf{r}_1^{(i)}$, to capture "all its important features". Algebraically, the matrices I_0^1 and R_1^0 are rectangular so that they have (nontrivial) ranges and kernels. Then
>
> - the norm $\|\mathsf{I}_0^1\mathsf{d}_0^{(i)} - \mathsf{e}_1^{(i)}\|$ can only be (relatively) small, if $\mathsf{e}_1^{(i)}$ is close to $Im(\mathsf{I}_0^1)$, the range of I_0^1,
> - $\mathsf{R}_1^0\mathsf{r}_1^{(i)}$ (nearly) vanishes if $\mathsf{r}_1^{(i)}$ is close to $Ker(\mathsf{R}_1^0)$, the kernel of R_1^0.
>
> These loosely formulated observations motivate our following development on how to complement the coarse-space correction to develop a truly efficient solver. Later, in the convergence analysis, we will see rigorous formulations.

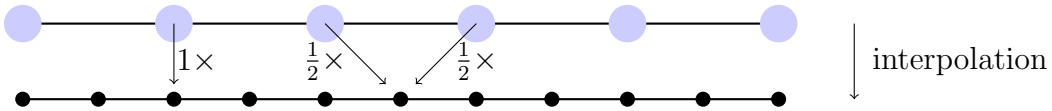
1.3.2 Interpolation and restriction for finite differences

1D

Given a fine grid (partitioning) with the step size $h_1 = 1/n$, it is natural to consider the coarse grid as "every second node", i.e., a partitioning with $h_0 = 2h_1$. Since the value in the boundary nodes is determined by the boundary condition (equal to zero), for the number of degrees of freedom, internal nodes, we have $N_1 = 2N_0 - 1$.



For the *interpolation*, we determine the values in all (fine) nodes from the values in the coarse nodes. A natural choice is illustrated in the figure below.

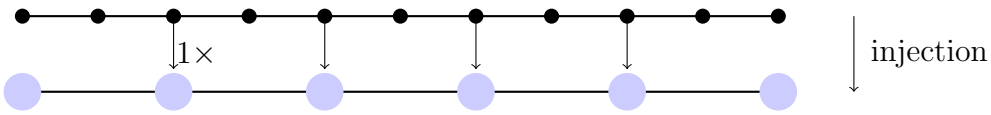


In the matrix notation,

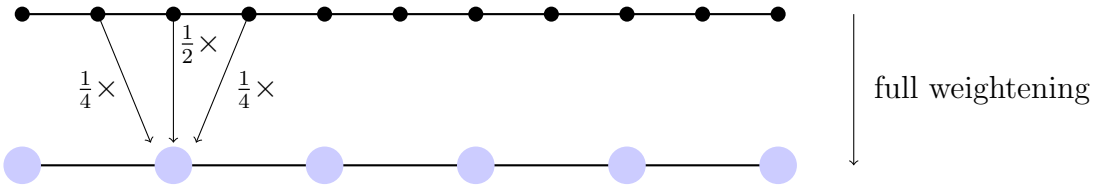
$$\mathsf{I}_0^1 = \begin{bmatrix} 1/2 & & & & & & \\ 1 & & & & & & \\ 1/2 & 1/2 & & & & & \\ & 1 & & & & & \\ & 1/2 & \ddots & & & & \\ & & & \ddots & 1/2 & & \\ & & & & 1 & & \\ & & & & 1/2 & & \end{bmatrix}$$

To determine the *restriction*, there are two options:

injection where one simply takes only the values in the nodes that are both on fine and coarse grid (and the rest is ignored)



full-weightening where the coarse value is determined also from the neighboring fine nodes



The matrix notation for the full-weightening restriction is then

$$\mathsf{R}_1^0 = \begin{bmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & & 1/4 & 1/2 & 1/4 \\ & & & & & & \ddots \end{bmatrix}$$

The fact that $\mathsf{R}_1^0 = c(\mathsf{I}_0^1)^T$ for some constant $c \in \mathbb{R}$ is called *variational property*. For the full-weightening restriction in 1D, $c = 1/2$. In general, the variational property simplifies the

analysis because of preserving the symmetry. Moreover, if[2]

$$\mathsf{A}_0 = \mathsf{R}_1^0 \mathsf{A}_1 \mathsf{I}_0^1,$$

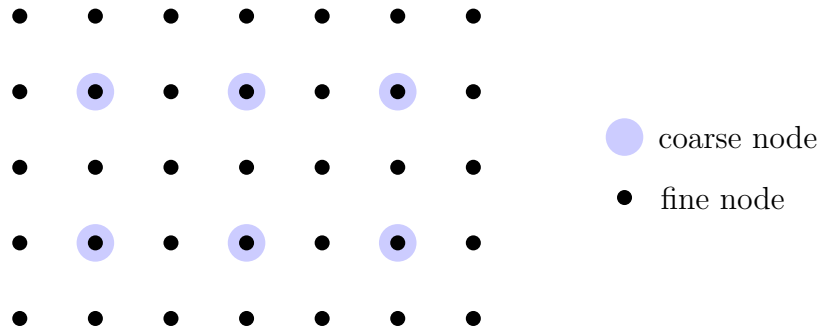
then for the coarse-space correction we have

$$\begin{aligned} \mathsf{u}^{(i+1)} &= \mathsf{u}^{(i)} + \mathsf{I}_0^1 (\mathsf{R}_1^0 \mathsf{A}_1 \mathsf{I}_0^1)^{-1} \mathsf{R}_1^0 \mathsf{r}^{(i)}, \\ \mathsf{e}^{(i+1)} &= \left(\mathsf{I} - \mathsf{I}_0^1 (\mathsf{R}_1^0 \mathsf{A}_1 \mathsf{I}_0^1)^{-1} \mathsf{R}_1^0 \mathsf{A}_1\right) \mathsf{e}^{(i)}, \end{aligned}$$

where $(\mathsf{I} - \mathsf{I}_0^1 (\mathsf{R}_1^0 \mathsf{A}_1 \mathsf{I}_0^1)^{-1} \mathsf{R}_1^0 \mathsf{A}_1)$ is a projection (i.e., $\mathsf{P} = \mathsf{P}^2$).

2D

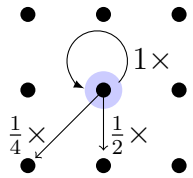
There are more choices of coarse degrees of freedom possible in 2D. The most popular that reduces the number of DOFs by factor of 4 while assuring that every fine-level node has two coarse nodes "nearby" is



For the ease of presentation, we did not display the boundary nodes.

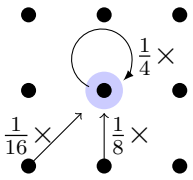
The *interpolation* can be represented by a so-called stencil, describing how the value in the coarse DOF is "spread" to the neighboring fine nodes. A natural one is

$$\begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}, \qquad \text{graphically}$$



Please note that the stencil is not a submatrix of the interpolation matrix. The values from the stencil are on a proper positions in a (sparse) column vector of the interpolation matrix.

The full-weightening *restriction* in 2D, given as the analogy to 1D case above, is as follows



[2]This happens for example for the matrices from finite element discretizations.

This gives, for the restriction matrix,

$$\mathsf{R}_1^0 = \frac{1}{4}(\mathsf{I}_0^1)^T = \left(\frac{1}{2}\right)^d (\mathsf{I}_0^1)^T.$$

1.3.3 Stationary iterative methods and smoothing property

Stationary iterative methods is a class of methods based on matrix splitting. Let

$$\mathsf{A} = \mathsf{D} - \mathsf{L} - \mathsf{U}, \tag{1.9}$$

where D is the diagonal of A, $-\mathsf{L}$ is its strictly lower part, and U is the strictly upper part of A.

Jacobi Method is based on the relationship

$$\mathsf{A}\mathsf{u} = \mathsf{f} = \mathsf{D}\mathsf{u} - (\mathsf{L}+\mathsf{U})\mathsf{u} \quad \Rightarrow \quad \mathsf{u} = \mathsf{D}^{-1}(\mathsf{L}+\mathsf{U})\mathsf{u} + \mathsf{D}^{-1}\mathsf{f}$$

giving the iterative scheme

$$\mathsf{u}^{(i+1)} = \mathsf{R}_J\mathsf{u}^{(i)} + \mathsf{D}^{-1}\mathsf{f}, \qquad \mathsf{R}_J = \mathsf{D}^{-1}(\mathsf{L}+\mathsf{U}). \tag{1.10}$$

Weighted (damped) Jacobi Method is a variant of Jacobi method where the new approximation is given as a linear combination of the previous approximation and the vector given by the (standard) Jacobi. For some $\omega \in \mathbb{R}$,

$$\begin{aligned} \mathsf{u}^{(i+1)} &= \omega(\mathsf{R}_J\mathsf{u}^{(i)} + \mathsf{D}^{-1}\mathsf{f}) + (1-\omega)\mathsf{u}^{(i)} = \mathsf{R}_\omega\mathsf{u}^{(i)} + \omega\mathsf{D}^{-1}\mathsf{f}, \\ \mathsf{R}_\omega &= \omega\mathsf{R}_J + (1-\omega)\mathsf{I} = \mathsf{I} - \omega\mathsf{D}^{-1}\mathsf{A}. \end{aligned} \tag{1.11}$$

With $\omega = 1$, $\mathsf{R}_\omega = \mathsf{R}_J$ and the weighted Jacobi becomes the standard Jacobi method.

The practical disadvantage of both the method is that they requires storing both $\mathsf{u}^{(i)}$ and $\mathsf{u}^{(i+1)}$. In some applications, it is not possible to store more than a single solution vector. It may be then useful, if the vector $\mathsf{u}^{(i)}$ is overwritten in the computation.

Gauss–Seidel Method is based on

$$(\mathsf{D}-\mathsf{L})\mathsf{u} = \mathsf{U}\mathsf{u} + \mathsf{f} \quad \Rightarrow \quad \mathsf{u} = (\mathsf{D}-\mathsf{L})^{-1}\mathsf{U}\mathsf{u} + (\mathsf{D}-\mathsf{L})^{-1}\mathsf{f}$$

giving the iterative scheme

$$\mathsf{u}^{(i+1)} = \mathsf{R}_{GS}\mathsf{u}^{(i)} + (\mathsf{D}-\mathsf{L})^{-1}\mathsf{f}, \qquad \mathsf{R}_{GS} = (\mathsf{D}-\mathsf{L})^{-1}\mathsf{U}. \tag{1.12}$$

The implementation of Gauss–Seidel allows to store only one approximate vector and update consecutively its components. Then, in contrast to Jacobi methods, the ordering of degrees of freedom may affect the behavior of the solver.

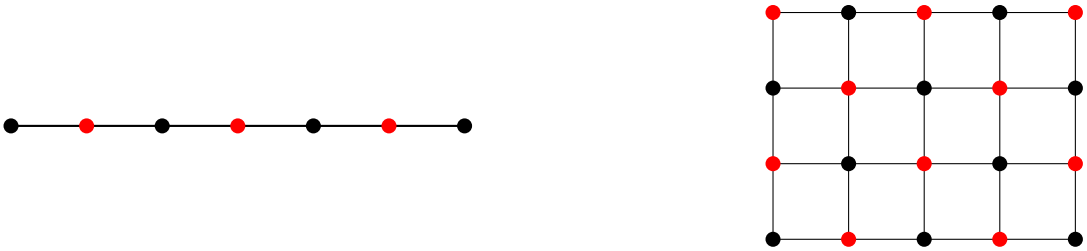


Figure 1.3: Splitting the nodes for red-black Gauss–Seidel in 1D and 2D

Red-black Gauss–Seidel Method is a variant of GS to enhance its parallelization. The idea is to split the degrees of freedom into two groups ("red" and "black") as in Figure 1.3.

The advantage of such splitting is that the "red" values are updated using the neighboring "black" values and vice versa. Instead of updating the elements of the vector $\mathsf{u}^{(i+1)}$ one by one, we can do the update in two steps; update first all the values in one of the groups and then update the second one.

There are also other stationary methods (for example Richardson method, Successive over-relaxation method (SOR), Symmetric successive over-relaxation (SSOR)) that we do not discuss in detail.

Smoothing property

For an iterative process

$$\mathsf{u}^{(i+1)} = \mathsf{R}\mathsf{u}^{(i)} + \mathsf{g},$$

satisfying $\mathsf{u} = \mathsf{R}\mathsf{u} + \mathsf{g}$ (i.e. such that u is its fixed point), the two consecutive errors satisfy

$$\mathsf{e}^{(i+1)} = \mathsf{R}\mathsf{e}^{(i)}. \tag{1.13}$$

Recursively

$$\mathsf{e}^{(i+1)} = \mathsf{R}^{i+1}\mathsf{e}^{(0)}. \tag{1.14}$$

We saw an analogous relation for iterative refinement with $\mathsf{R} = \mathsf{I} - \mathsf{M}^{-1}\mathsf{A}$.

From the elementary course on numerical analysis,

$$\lim_{i\to\infty} \mathsf{R}^i = 0 \quad \Longleftrightarrow \quad \rho(R) < 1,$$

where $\rho(\cdot)$ is the spectral radius. In other words, R^i tends to zero iff $|\lambda| < 1$ for each λ from the spectrum of R.

Let (λ, v) be the eigenpair of R. Then

$$\mathsf{R}^i\mathsf{v} = \mathsf{R}^{i-1}\mathsf{R}\mathsf{v} = \lambda\mathsf{R}^{i-1}\mathsf{v} = \cdots = \lambda^i\mathsf{v}$$

and $\|\mathsf{R}^i\mathsf{v}\| = |\lambda|^i\|\mathsf{v}\|$. Therefore $\|\mathsf{R}^i\mathsf{v}\|/\|\mathsf{v}\|$ converges to 0 faster for $|\lambda|$ small (closer to 0) and converges slower for $|\lambda|$ large (closer to 1).

When decomposing $\mathsf{e}^{(i)}$ into the eigensubspaces of R, some components will decay more rapidly as $i \to \infty$, while others will persist longer.

In applications, the eigenvectors of R associated with small eigenvalues (in the magnitude) are typically oscillating and those associated with large eigenvalues are smooth. Therefore

$$\mathsf{e}^{(i)} = \mathsf{R}^i \mathsf{e}^{(0)} \quad \text{is for large } i \text{ a smooth vector.}$$

This is therefore called a *smoothing property (of stationary methods).*

The smoothing property has two important consequences:

- The error will be after few steps dominated by the eigenvectors associated with large eigenvalues of R (that are smooth).
- For appropriate starting vectors, smoothers (stationary iterative methods) converge fast. Such vectors are dominated by the eigenvectors associated with the smallest eigenvalues of R.

It is however more practical to relate smoothness to the problem to be solved and not to a particular method. For $(\lambda_k, \mathsf{v}_k)$ an eigenpair of A, there holds

$$\|\mathsf{v}_k\|_{\mathsf{A}}^2 = \mathsf{v}_k^T \mathsf{A} \mathsf{v}_k = \mathsf{v}_k^T \lambda_k v_k = \lambda_k \|\mathsf{v}_k\|^2.$$

For model Poisson problem, the A norm corresponds to an energy that is minimized by the exact solution u. Then it is said that a *smooth* vector has small energy, while an *oscillating* vector has high energy. Therefore smooth eigenvectors are those corresponding to small eigenvalues while oscillating eigenvectors to the large ones.

Since A is diagonalizable, there exists an orthonormal basis of $\mathbb{R}^N$ formed from its eigenvectors. We can then focus on how the smoothing (application of a stationary iterative method) and coarse-space correction act on the eigenvectors of A.

Eigenpairs of 1D finite-difference matrices and smoothing property for (weighted) Jacobi

Recall

$$\mathsf{R}_\omega = \omega \mathsf{R}_J + (1-\omega)\mathsf{I} = \mathsf{I} - \omega \mathsf{D}^{-1} \mathsf{A}.$$

For A from (1.4), $\mathsf{D} = 2\mathsf{I}$ and therefore[3]

$$\mathsf{D}^{-1} = \frac{1}{2}\mathsf{I}, \quad \mathsf{R}_\omega = \mathsf{I} - \frac{\omega}{2}\mathsf{A}.$$

From this one can see that for $(\lambda_k, \mathsf{v}_k)$ an eigenpair of A, v_k is also an eigenvector of R_ω and

$$\mathsf{R}_\omega \mathsf{v}_k = \left(1 - \frac{\omega}{2}\lambda_k\right) \mathsf{v}_k.$$

[3] In 2D and 3D, $\mathsf{D} = \alpha \mathsf{I}$ for some α so that analogous result holds too.

Exercise. Verify by computation that for A from (1.4) of order n, the eigenpairs are

$$\lambda_k(\mathsf{A}) = 4\sin^2\left(\frac{k\pi}{2n}\right), \qquad 1 \le k \le n-1, \tag{1.15}$$

$$\mathsf{v}_{k,j} = \sin\left(\frac{jk\pi}{n}\right) \qquad 1 \le k \le n-1, 1 \le j \le n-1. \tag{1.16}$$

Using (1.15), the eigenvalues of R_ω are

$$\lambda_k(\mathsf{R}_\omega) = 1 - 2\omega\sin^2\left(\frac{k\pi}{2n}\right), \qquad 1 \le k \le n-1. \tag{1.17}$$

Note that with a finer partitioning, n increases, leading to the spectrum of R_ω approaching one, which generally slows down the convergence of the method.

We now illustrate the above findings in experiments. First, we plot in Figure 1.4, left, the function

$$\lambda(x) = 1 - 2\omega\sin^2\left(\frac{x\pi}{2n}\right), \quad x \in (0, n), \tag{1.18}$$

here with $n = 64$ and several choices of ω. Namely, $\omega = \frac{1}{3}$, $\frac{1}{2}$, $\frac{1}{2}$, and 1. Please note that the choice of n does not play a role here; it is only important to define the smallest eigenvalue in (1.17) corresponding to $x = 1$. From the figure we can identify the modes (given by integer values of x) where the absolute value of the function λ is the smallest, which indicate that the damped Jacobi is reducing these modes the most efficiently. For (standard) Jacobi, this is around $x = n/2$, for a damped variant, this x increases; for example, it equals to n for $\omega = 1/2$.

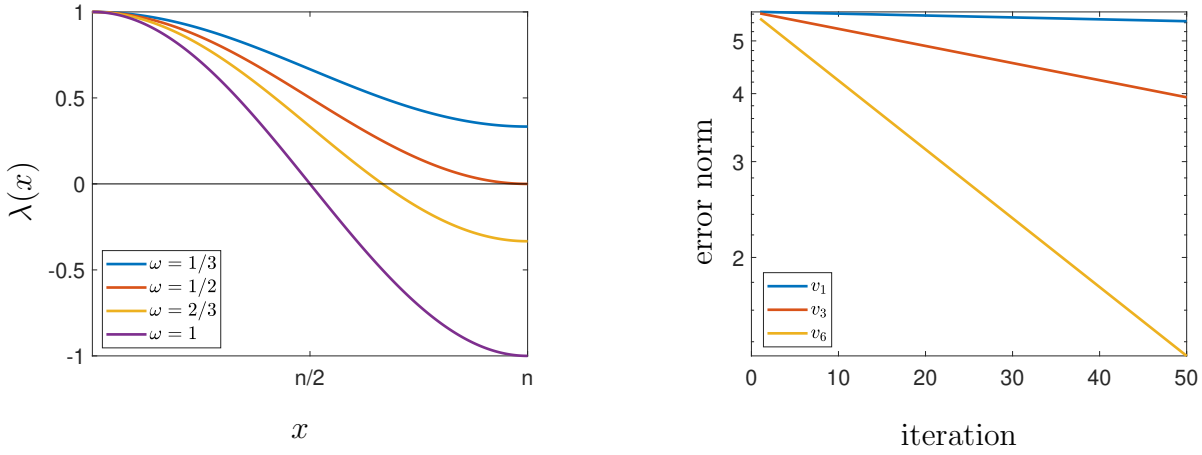


Figure 1.4: Function (1.18) describing the spectrum of R_ω in 1D (left). Euclidean norm of the error in the iterations of weighted Jacobi with $\omega = 2/3$ when the initial error is equal to v_1, v_3, respectively v_6 (right).

Varying convergence speed for different eigenvectors from (1.16) is illustrated in the right part of Figure 1.4, where the euclidean norm of the error $\|\mathsf{e}^{(i)}\|$ is plot. Therein, we set

$n = 64$, and solve the problem $\mathsf{A}\mathsf{u} = 0$ by damped Jacobi with $\omega = 2/3$ and starting with v_k with $k = 1, 3, 6$. This choice of the problem and the initial guess means that the initial error is equal to v_k. We clearly see that the error corresponding to a smaller k is reduced significantly less efficiently.

Finally, we plot the contraction of the error

$$\frac{\|\mathsf{e}^{(5)}\|}{\|\mathsf{e}^{(0)}\|} \tag{1.19}$$

in 5 iterations of (damped) Jacobi, together with the number of iterations needed to reduce the euclidean norm of the error $\|\mathsf{e}^{(i)}\|$ by two order of magnitude for the initial error equal to v_k, $k = 1, \ldots, n$, $n = 64$. For the ease of presentation, the y-axis is cut at 100; in fact, more than 5700 iterations are needed for $k = 1$ and damped Jacobi. In Figure 1.5, we give the results for Jacobi method and in Figure 1.6 for the damped Jacobi with $\omega = 2/3$. One can see that Jacobi method has difficulties to reduce the error in the subspaces corresponding to the eigenvectors with small and large k. The damped Jacobi handles better eigenvectors with large k but still fails for those with k small. These are low oscillating vectors; cf. (1.15).

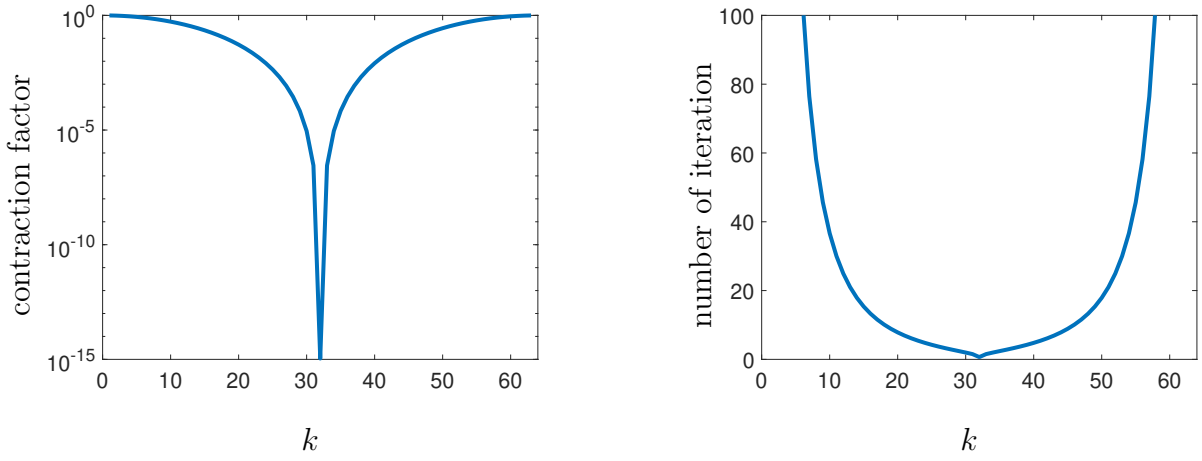


Figure 1.5: Jacobi method: contraction factor (1.19) (left) and the number of iterations needed to reduce the euclidean norm of the error by 100 (right). The initial error is given by the eigenvectors v_k from (1.16). For small and large indices the number of iterations exceeds 100.

Gauss–Seidel method

For Gauss–Seidel, in general, the eigenvectors of A are not the eigenvectors of R_{GS}. Therefore the analysis of smoothing property is more subtle and we only illustrate it below in Figure 1.7.

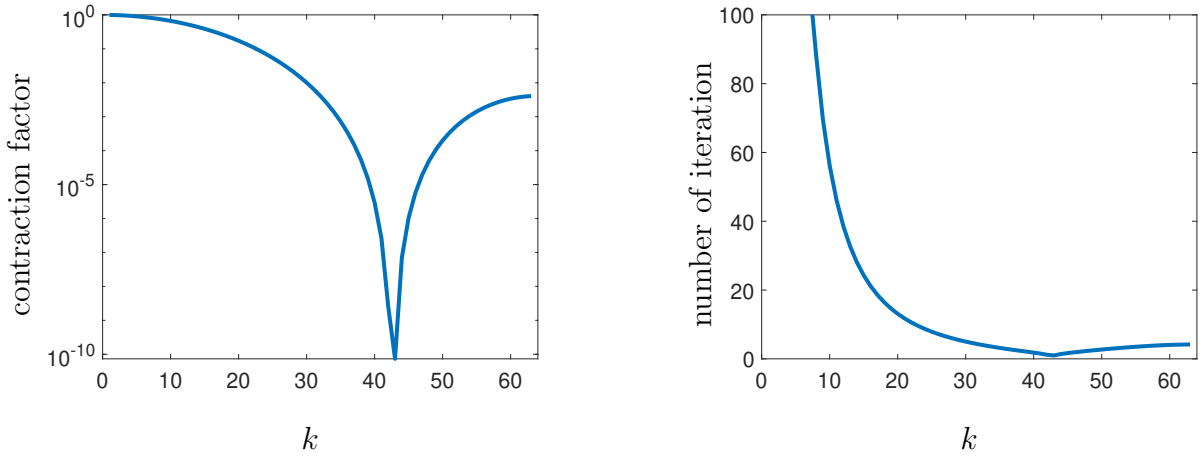


Figure 1.6: Damped Jacobi method with $\omega = 2/3$: contraction factor (1.19) (left) and the number of iterations needed to reduce the euclidean norm of the error by 100 (right). The initial error is given by the eigenvectors v_k from (1.16).

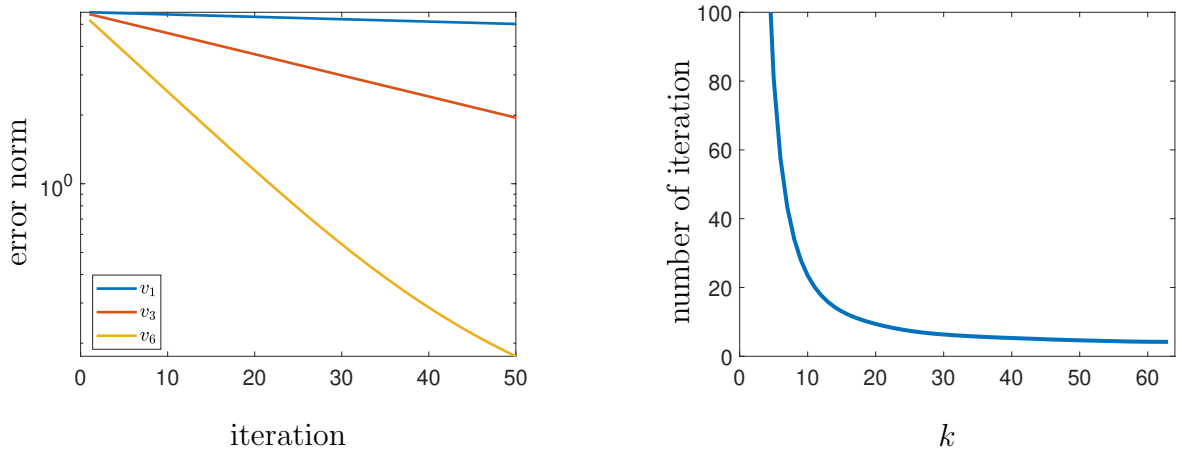


Figure 1.7: Gauss–Seidel method: Euclidean norm of the error when the initial error is equal to v_1, v_3, respectively v_6 (left). Number of iterations needed to reduce the euclidean norm of the error by 100; the initial error is given by the eigenvectors v_k from (1.16) (right).

1.4 Multigrid Method

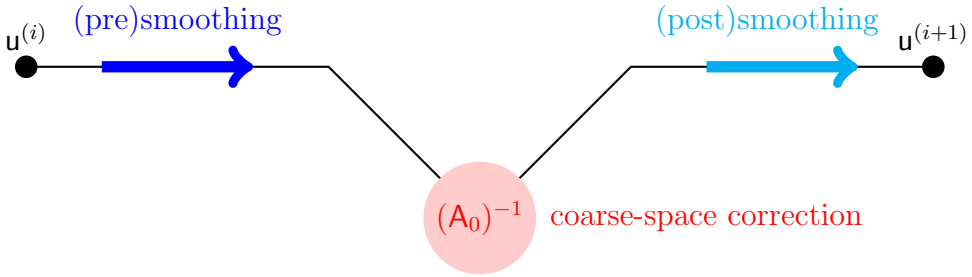
The multigrid solver (or preconditioner) efficiently combines smoothing and coarse-space correction. In a two-level setting, we can consider three variants:

Var. I: start with smoothing and correct a smooth error on the coarse level

Var. II: perform the coarse-space correction, giving an oscillating error that is then smoothed out

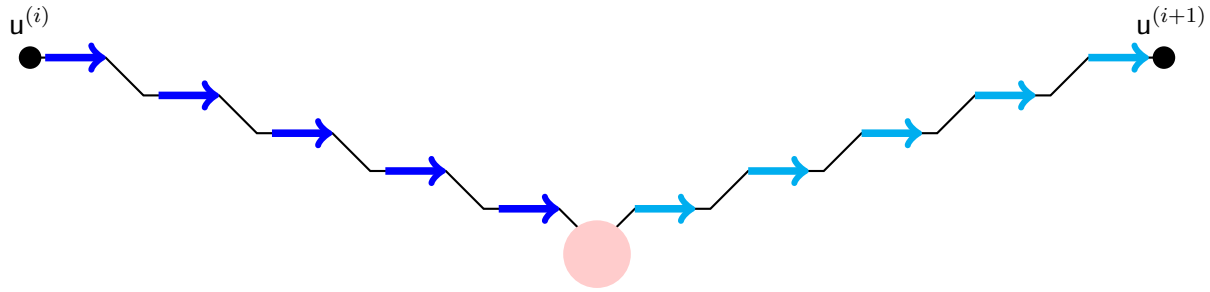
Var. III: combine **I.** (*presmoothing*) and **II.** (*postsmoothing*)

The latter variant can be illustrated by the scheme as

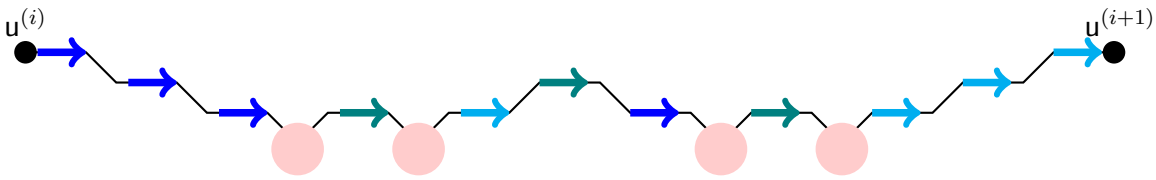


Generalization to a hierarchy of grids is based on the idea not to solve the problem in the coarse-space correction by a direct solver (mathematically by inverting the matrix), but to use a recursion until the coarsest-level problem is small enough.

The basic multigrid scheme is called *V-cycle* and can be illustrated as

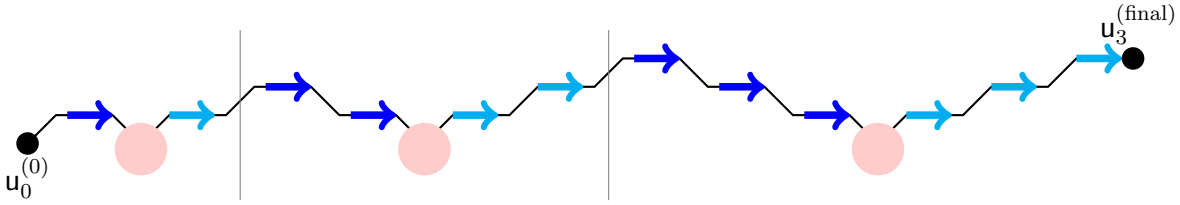


The recursive call can be more complex, giving, for example *W-cycle*



We will later see a rigorous algorithmic description of both V- and W-cycles.

Finally, we illustrate a FMG (full multigrid) that can be regarded as a non-iterative procedure, which starts on the coarsest level



Exercise. Write a pseudocode for the three MG variants described above. What additional functions does FMG require beyond those used in V- or W-cycles?

Chapter 2

Abstract formulation of a multigrid and its convergence

In this section, we present a general, abstract formulation of multigrid, derive an operator that describes error reduction, and establish several sufficient conditions for the convergence of the multigrid solver. This presentation closely follows [Shaidurov, 1995, Section 4].

2.1 An abstract multigrid algorithm

Consider

- a sequence of finite-dimensional spaces

  $$M_0, M_1, \ldots, M_k$$

  with the inner products $(\cdot,\cdot)_i$ on M_i, $i = 0, 1, \ldots, k$,

- interpolation operators

  $$I_i^{i+1} : M_i \to M_{i+1}, \qquad i = 0, 1, \ldots, k-1,$$

- restriction operators

  $$R_i^{i-1} : M_i \to M_{i-1}, \qquad i = 1, 2, \ldots, k,$$

- invertible operators

  $$A_i : M_i \to M_i, \qquad i = 0, 1, \ldots, k,$$

- smoothers

  $$D_{i,\ell}^{PRE}, D_{i,j}^{POST} : M_i \to M_i, \qquad i = 1, 2, \ldots, k, \quad \ell = 1, 2, \ldots, m_1, \quad j = 1, 2, \ldots, m_2.$$

Given $f_k \in m_k$, we seek $u_k \in M_k$ such that

$$A_k u_k = f_k.$$

We define MG_i-algorithm for solving auxiliary problems $A_i v_i = g_i$ for arbitrary right-hand side g_i and an initial approximation $w_i^{(0)}$.

MG_i-algorithm $w_i^{(1)} = MG_i(w_i^{(0)}, g_i)$

If $i = 0$, $w_i^{(1)} = A_0^{-1} g_i$ and leave, otherwise

(A1) presmoothing: set $v^{(0)} = w_i^{(0)}$ and

$$v^{(\ell)} = v^{(\ell-1)} + D_{i,\ell}^{PRE}(g_i - A_i v^{(\ell-1)}), \qquad \ell = 1, 2, \ldots, m_1.$$

(A2) restriction:

$$g_{i-1} = R_i^{i-1}\left(g_i - A_i v^{(m_1)}\right),$$

(A3) coarse-space solution: set $\tilde{w}^{(0)} = 0$ and call γ-times

$$\tilde{w}^{(s)} = MG_{i-1}(\tilde{w}^{(s-1)}, g_{i-1}) \qquad s = 1, 2, \ldots, \gamma,$$

(A4) correction:

$$y^{(0)} = v^{(m_1)} + I_{i-1}^{i} \tilde{w}^{(\gamma)},$$

(A5) postmoothing:

$$y^{(j)} = y^{(j-1)} + D_{i,j}^{POST}(g_i - A_i y^{(j-1)}), \qquad j = 1, 2, \ldots, m_2.$$

Finally, set $w_i^{(1)} = y^{(m_2)}$.

We shall now describe the evolution of the error within one call of MG_i. Given the problem $A_i v_i = g_i$, the initial guess $w^{(0)}$, we have

$$\begin{aligned} e^{(0)} &= v_i - w^{(0)} \\ e^{(1)} &= v_i - w^{(1)}, \qquad w_i^{(1)} = MG_i(w_i^{(0)}, g_i). \end{aligned}$$

Define *error suppression operator*

$$B_i : e^{(0)} \mapsto e^{(1)}.$$

Denote

$$\begin{aligned} J_i^{PRE} &= (I - D_{i,m_1}^{PRE} A_i)(I - D_{i,m_1-1}^{PRE} A_i) \cdots (I - D_{i,1}^{PRE} A_i), \\ J_i^{POST} &= (I - D_{i,m_2}^{POST} A_i)(I - D_{i,m_2-1}^{POST} A_i) \cdots (I - D_{i,1}^{POST} A_i). \end{aligned}$$

In the previous part, we saw that the error after smoothing with D satisfies (in the previous notation)

$$\mathsf{e}^{(i+1)} = (\mathsf{I} - \mathsf{DA})\mathsf{e}^{(i)}$$

and the error after coarse-space correction

$$\mathsf{e}^{(i+1)} = (\mathsf{I} - \mathsf{I}_0^1\mathsf{A}_0^{-1}\mathsf{R}_1^0\mathsf{A}_1)\mathsf{e}^{(i)}.$$

Analogously, we get for two levels and MG_1,

$$B_1 = J_1^{POST}(I - I_0^1 A_0^{-1} R_1^0 A_1) J_1^{PRE}.$$

How B_i looks on finer levels with $i > 1$? The smoothing stays (up to the indices) the same and the difference shall be in the "coarse-space correction" part. We therefore expect

$$B_i = J_i^{POST} \boxed{\text{CP}} J_i^{PRE},$$

where $\boxed{\text{CP}}$ stands for a recursive call of MG_{i-1} and most probably will involve the error suppression B_{i-1}.

Let now focus on development of the error in steps (A2)–(A4):

- after (A1) we have $v^{(m_1)} \approx v_i$ with the error $\sigma^{(m_1)} = v_i - v^{(m_1)}$.
- In the restriction (A2)

$$g_{i-1} = R_i^{i-1}\left(g_i - A_i v^{(m_1)}\right) = R_i^{i-1} A_i \sigma^{(m_1)}.$$

- The coarse-space solve (A3) starts with zero initial guess, therefore the error is equal to the solution $\tilde{w^*}$. After one application of MG_{i-1} the error is $B_{i-1}\tilde{w^*}$ and there holds

$$\tilde{w}^{(1)} = \tilde{w^*} - (\tilde{w^*} - \tilde{w}^{(1)}) = \tilde{w^*} - B_{i-1}\tilde{w^*}.$$

  After γ steps of MG_{i-1} therefore

$$\tilde{w}^{(\gamma)} = (I - (B_{i-1})^\gamma)\tilde{w^*}.$$

- In the correction (A4)

$$y^{(0)} = v^{(m_1)} + I_{i-1}^i \tilde{w}^{(\gamma)}$$

  and therefore

$$\begin{aligned} v_i - y^{(0)} &= (v_i - v^{(m_1)}) - I_{i-1}^i \tilde{w}^{(\gamma)} \\ &= \sigma^{(m_1)} - I_{i-1}^i (I - (B_{i-1})^\gamma)\tilde{w^*} \\ &= \left(I - I_{i-1}^i (I - (B_{i-1})^\gamma) A_{i-1}^{-1} R_i^{i-1} A_i\right) \sigma^{(m_1)} \end{aligned}$$

Altogether, the error suppression operator is

$$B_i = J_i^{POST}\left(I - I_{i-1}^{i}(I - (B_{i-1})^{\gamma})A_{i-1}^{-1}R_i^{i-1}A_i\right)J_i^{PRE}, \qquad B_0 = 0. \tag{2.1}$$

If the operators A_i, R_i^{i-1}, I_{i-1}^{i}, $D_{i,\ell}^{PRE}$, $D_{i,j}^{POST}$ are linear, so is B_i. Additionally, B_i is independent of g_i or w_0.

To prove the convergence of the multigrid, it is sufficient show that

$$\rho(B_i) < 1.$$

Since for every (induced) matrix norm $\rho(B_i) \leq |||B_i|||$, we show that

$$|||B_i||| < 1 - \epsilon, \qquad \epsilon > 0,$$

for some appropriate norm.

To sum up,

$$|||B_i||| < 1 \quad \Longrightarrow \quad \rho(B_i) < 1 \quad \Longrightarrow \quad (B_i)^j e^{(0)} \to 0 \quad \forall w_0, g_i.$$

2.2 Convergence of a symmetric multigrid

A symmetric multigrid is an important variant of multilevel methods. In particular, it can be used as a preconditioner for the Conjugate Gradient method.

The symmetry in MG is represented by these assumptions:

(S1) operators A_i are self-adjoint and positive definite (in case of matrices, A_i are symmetric positive-definite),

(S2) $I_{i-1}^{i} = \left(R_i^{i-1}\right)^*$ in the sense of inner products $(\cdot,\cdot)_i$ on M_i,

(S3) $J_i^{PRE} = \left(J_i^{POST}\right)^*$,

(S4) variational property $A_{i-1} = R_i^{i-1}A_iI_{i-1}^{i}$,

(S5) $J_i^{PRE}A_i = A_iJ_i^{PRE}$ (which also assures that $(J_i^{PRE})^*A_i = A_i(J_i^{PRE})^*$).

For the ease of presentation, set $J_i := J_i^{PRE}$.

Thanks to (S1), operators A_i define inner products and the norm

$$(u,v)_{A_i} = (A_iu, v)_i \qquad u, v \in M_i$$
$$\|u\|_{A_i} = (u,u)_{A_i}^{1/2} \qquad u \in M_i$$

We also define an operator of (exact) coarse-space correction $Q_i : u \to w$ such that

$$w = u - I_{i-1}^{i}A_{i-1}^{-1}R_i^{i-1}A_iu.$$

Therefore

$$Q_i = I - I_{i-1}^{i}A_{i-1}^{-1}R_i^{i-1}A_i.$$

Lemma 2.1 (Properties of Q_i)**.** *Under conditions (S1), (S2), (S4) the operator Q_i is self-adjoint projector from M_i into orthogonal complement of $I_{i-1}^i M_{i-1}$ with respect to the inner product $(\cdot,\cdot)_{A_i}$.*

Proof. Let $w = Q_i u$. We show that

$$(w, z)_{A_i} = 0 \quad \forall z \in I_{i-1}^i M_{i-1}.$$

We have

$$\begin{aligned}(A_i w, z)_i &= (A_i Q_i u, z)_i \overset{z=I_{i-1}^i y}{=} (A_i Q_i u, I_{i-1}^i y)_i \\ &= (A_i u, I_{i-1}^i y)_i - (\underbrace{R_i^{i-1} A_i I_{i-1}^i}_{A_{i-1}} A_{i-1}^{-1} R_i^{i-1} A_i u, y)_i \\ &= (R_i^{i-1} A_i u, y)_i - (R_i^{i-1} A_i u, y)_i = 0.\end{aligned}$$

Operator Q_i is a projection (as we have already seen)

$$Q_i^2 = I - 2 I_{i-1}^i A_{i-1}^{-1} R_i^{i-1} A_i + I_{i-1}^i A_{i-1}^{-1} \underbrace{R_i^{i-1} A_i I_{i-1}^i}_{A_{i-1}} A_{i-1}^{-1} R_i^{i-1} A_i = I_{i-1}^i A_{i-1}^{-1} R_i^{i-1} A_i.$$

Finally, Q_i is self-adjoint because

$$\begin{aligned}(Q_i u, v)_{A_i} &= (A_i (I - I_{i-1}^i A_{i-1}^{-1} R_i^{i-1} A_i) u, v)_i \\ &= (A_i u, v)_i - (A_i u, I_{i-1}^i A_{i-1}^{-1} R_i^{i-1} A_i v)_i = (u, Q_i v)_{A_i}.\end{aligned}$$

□

Now we can state a sufficient condition for the convergence. Let there exist $c^* > 0$ such that $\forall i = 1, \ldots, k$, and $\forall v \in M_i$

$$\|v\|_{A_i}^2 - \|J_i v\|_{A_i}^2 \geq c^* \|Q_i J_i v\|_{A_i}^2. \tag{2.2}$$

After the first, introductory part of the course, we should be able to properly understand the condition (2.2). We can discuss two scenarios. First, the vector v is "oscillatory" so that it can by efficiently reduced by smoothing giving $J_i v$ with small norm. In that case, the coarse-space correction of $J_i v$ (giving the vector $Q_i J_i v$) need not be very efficient. Second case is a smooth vector that is not smoothed out and therefore $\|v\|_{A_i}$ and $\|J_i v\|_{A_i}$ are close to each other. Therefore, to fulfill (2.2), the vector $Q_i J_i v$ after the coarse-space correction must be significantly reduced.

Theorem 2.2 (Convergence of a symmetric V-cycle MG)**.** *Let (S1) – (S5) hold together with (2.2) for MG_i algorithm with $\gamma = 1$. Then*

$$\|B_i\|_{A_i} = \sup_{v \in M_i \setminus 0} \frac{\|B_i v\|_{A_i}}{\|v\|_{A_i}} \leq \frac{1}{1 + c^*} \ (< 1). \tag{2.3}$$

Proof. Will be given on the lecture. Otherwise see [Shaidurov, 1995, Thm. 4.5]. □

A single inequality (2.2) yielding a sufficient condition for convergence is presented in [Shaidurov, 1995, Eq. (4.82)] probably for the first time. More common in the literature is to have two conditions, one for the intergrid operators and the second for the smoother. We now state these conditions and show that they together imply (2.2) and, consequently, the convergence of the multigrid.

First, consider so-called *approximation assumption*: let there exist constant $c_1 > 0$ such that

$$(Q_i v, v)_{A_i} \le c_1 \frac{\|A_i v\|_i^2}{\lambda_i^*} \qquad \forall v \in M_i, \tag{2.4}$$

where λ_i^* is either the largest eigenvalue of A_i or its upper bound.

Second, *smoothing assumption* states: let there exist $c_2 > 0$ such that

$$\frac{\|A_i J_i v\|_i^2}{\lambda_i^*} \le c_2 \left(\|v\|_{A-i}^2 - \|J_i v\|_{A-i}^2 \right), \qquad \forall v \in M_i. \tag{2.5}$$

Lemma 2.3. *Under conditions (S1), (S2), (S4) the assumptions* (2.4) *and* (2.5) *imply* (2.2) *with the constant* $c^* = (c_1 c_2)^{-1}$.

Proof. From Theorem 2.1, Q_i is a projection and self-adjoint with respect to $(\cdot,\cdot)_{A_i}$. Therefore

$$\|Q_i v\|_{A_i}^2 = (Q_i v, Q_i v)_{A_i} = (Q_i^2 v, v)_{A_i} = (Q_i v, v)_{A_i}.$$

Now choose $J_i v$ for v, giving

$$\|Q_i J_i v\|_{A_i}^2 = (Q_i J_i v, J_i v)_{A_i} \overset{(2.4)}{\le} c_1 \frac{\|A_i J_i v\|_i^2}{\lambda_i^*} \overset{(2.5)}{\le} c_1 c_2 (\|v\|_{A_i} - \|J_i v\|_{A_i}),$$

which gives (2.2).

□

2.2.1 Smoothing assumption and stationary iterative solvers

We shall now discuss when the smoothing assumption (2.5) is satisfied for stationary iterative methods. Let

$$D_{i,\ell}^{PRE} = D_{i,j}^{POST} = D_i \qquad \ell = 1, \dots, m_1, \quad j = 1, \dots, m_2, \quad \forall i.$$

Recall that from (S3), we have $m := m_1 = m_2$.

Set $K_i = I - D_i A_i$, giving $J_i = K_i^m$. Then (2.5) can be formulated for a single smoothing iteration: let there exist $c_3 > 0$ such that

$$\frac{\|A_i v\|_i^2}{\lambda_i^*} \le c_3 ((I - K_i)v, v)_{A_i} \qquad \forall v \in M_i. \tag{2.6}$$

Lemma 2.4 ([Shaidurov, 1995, Lemma 4.3.3])**.** *Let (S1) and* (2.6) *hold. Let K_i are symmetric, nonnegative in the inner product $(\cdot,\cdot)_{A_i}$ and $\|K_i\|_{A_i} \le 1$. Then* (2.5) *holds with the constant $c_2 = c_3/(2m)$.*

Proof. From [Bramble and Pasciak, 1987, p. 316], we have

$$((I-K_i)K_i^{2m}v,v)_{A_i} \le \frac{1}{2m}\sum_{j=0}^{2m-1}((I-K_i)K_i^j v,v)_{A_i}, \tag{2.7}$$

where we needed the properties of K_i (its spectrum w.r.t. the inner product $(\cdot,\cdot)_{A_i}$ stays in the interval $[0,1]$). When expanding the sum, intermediate terms vanish and we stay with

$$\frac{1}{2m}\sum_{j=0}^{2m-1}((I-K_i)K_i^j v,v)_{A_i} = \frac{1}{2m}((I-K_i^{2m})v,v)_{A_i}.$$

Using the symmetry of K_i and $K_i^m = J_i$,

$$((I-K_i)J_i v, J_i v)_{A_i} \le \frac{1}{2m}\left(\|v\|_{A_i}^2 - \|J_i v\|_{A_i}^2\right).$$

Now take $J_i v$ for v in (2.6) and use the previous inequality to get

$$\frac{\|A_i J_i v\|_i^2}{\lambda_i^*} \le c_3((I-K_i)J_i v, J_i v)_{A_i} \le \frac{c_3}{2m}\left(\|v\|_{A_i}^2 - \|J_i v\|_{A_i}^2\right).$$

□

If interested, I have written down a detailed proof of (2.7).

2.3 Convergence in a general case

In a general case, we assume

(G1) A_i are nonsingular, $i = 0, 1, \ldots, k$,

(G2) $I_{i-1}^1 = (R_i^{i-1})^*$,

(G3) $\|J_i\|_i \le c_J$, $i = 1, 2, \ldots, k$,

(G4) $\underline{c}_I \|I_{i-1}^i u\|_i \le \|u\|_{i-1} \le \overline{c}_I \|I_{i-1}^i u\|_i, \quad \forall u \in M_{i-1}$, $i = 1, 2, \ldots, k$,

(G5) the number of smoothing steps is $m = m_1$, $m_2 = 0$, and

$$\|A_i^{-1} - I_{i-1}^1 A_{i-1}^{-1} R_i^{i-1}\|_i \cdot \|A_i J_i\|_i \le \eta(m), \quad i = 1, \ldots, k,$$

where η does not depend on i a converges to 0 for $m \to \infty$.

The inequality in assumption (G5) can be replaced by

(G5a) $\| \left(A_i^{-1} - I_{i-1}^1 A_{i-1}^{-1} R_i^{i-1}\right) A_i J_i \|_i \le \eta(m), \quad i = 1, \dots, k.$

In a general case, there is no norm induced by A_i so that we will work with the norm

$$\|u\|_i = (u,u)_i^{1/2}, \quad u \in M_i, \quad i = 0, 1, \dots, k.$$

The associated operator norm is

$$\|A\|_i = \sup_{u \in M_i \setminus 0} \frac{\|Au\|_i}{\|u\|_i}.$$

Theorem 2.5 (Convergence of W-cycle in the general case)**.** *Let (G1)–(G5) (alternatively (G5a)) hold for MG_i algorithm with $\gamma = 2$, $m_1 = m$, and $m_2 = 0$. Then for arbitrary $\xi \in (0,1)$ there exists m_0 such that $\forall m \ge m_0$*

$$\|B_i\|_i \le \xi \qquad \forall i = 0, 1, \dots, k. \tag{2.8}$$

Proof. Will be given on the lecture. Otherwise see [Shaidurov, 1995, Thm. 4.11]. □

Index

Bibliography

[Bramble and Pasciak, 1987] Bramble, J. H. and Pasciak, J. E. (1987). New convergence estimates for multigrid algorithms. *Math. Comput.*, 49:311–329.

[Briggs et al., 2000] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A multigrid tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition.

[Rüde, 1993] Rüde, U. (1993). *Mathematical and computational techniques for multilevel adaptive methods*, volume 13 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

[Shaidurov, 1995] Shaidurov, V. V. (1995). *Multigrid methods for finite elements*, volume 318 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht. Translated from the 1989 Russian original by N. B. Urusova and revised by the author.