FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Anna Marie Minarovičová

Basic Properties of Multigrid Methods

Department of Numerical Mathematics

Supervisor of the bachelor thesis: RNDr. Jan Papež, Ph.D.
Study programme: Mathematical modelling

Prague 2024

I declare that I carried out this bachelor thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

In gratitude's tune, my sincere poem starts,
To my parents, first, whose love forever in my heart resides.
Their support, the sturdy ground on which I stand,
Their encouragement, the compass in my hand.

Next, to grandma and fam, reliable and true,
Their presence, like a rock, guided me anew.
In high school days, friends sparked the scholar's flame,
Their influence, the reason for my academic aim.

To classmates, friends, and community so grand,
Whose solidarity made challenges less unplanned.
And to Dr. Jan Papež, the best supervisor near,
Whose guidance and insight, clearly sincere.

Not just for thesis but also for the kindness that brightly shone,
when in the first university year, deciding to stay or move on.
To all who kept their support, I'm truly glad,
Your presence means the world to me, I add.

Title: Basic Properties of Multigrid Methods

Author: Anna Marie Minarovičová

Department: Department of Numerical Mathematics

Supervisor: RNDr. Jan Papež, Ph.D., Institute of Mathematics CAS

Abstract: Multigrid methods are among the most effective iterative methods for the numerical solution of partial differential equations (PDEs). In the thesis, we consider Poisson's equation as the model problem and present its discretization by the finite difference method. Discretization of PDEs gives typically large algebraic systems of linear equations. Various iterative methods can struggle to find an enough accurate approximation within the allocated time. In particular, relaxation methods such as Jacobi or Gauss-Seidel effectively reduce oscillating parts of the error but are inefficient in reducing smooth error components. Multigrid methods combine relaxation methods with correction on a coarser grid to overcome this deficiency. The problem discretized on a coarser grid is smaller and easier to solve. Typically, a recursive error correction is considered using a hierarchy of grids until the coarsest problem is small enough to get a solution quickly by a direct solver. The purpose of this thesis is to discuss the main principles and thoughts behind the multigrid methods, alongside some practical examples and numerical experiments.

Keywords: multigrid methods, numerical methods, solution of linear algebraic systems, discretization hierarchy

Název práce: Základní vlastnosti víceúrovňových metod

Autor: Anna Marie Minarovičová

Katedra: Katedra Numerické matematiky

Vedoucí bakalářské práce: RNDr. Jan Papež, Ph.D., Matematický ústav AV ČR

Abstrakt: Víceúrovňové metody patří mezi nejefektivnější iterační metody pro numerické řešení parciálních diferenciálních rovnic (PDR). V práci uvažujeme jako modelový problém Poissonovu rovnici a její diskretizaci metodou konečných diferencí. Obecně diskretizace PDR vede na velké soustavy lineárních rovnic. Různé iterační metody mohou mít potíže s nalezením dostatečně přesné aproximace v daném čase. Zejména relaxační metody, jako je Jacobi nebo Gauss-Seidel, účinně redukují oscilující části chyby, ale jsou neefektivní v redukci hladkých chybových složek. Multigridní metody kombinují relaxační metody s korekcí na hrubší síti, aby překonaly tento nedostatek. Problém diskretizovaný na hrubší síti je menší a snáze řešitelný. Oprava na hrubší síti se obvykle realizuje rekurzivně pomocí hierarchie sítí, dokud nejhrubší problém není dostatečně malý na to, aby jej bylo možné řešit přímým řešičem. Cílem této práce je diskutovat hlavní principy a myšlenky, které stojí za multigridními metodami, spolu s některými praktickými příklady a numerickými experimenty.

Klíčová slova: víceúrovňové metody, numerické metody, řešení soustav lineárních rovnic, hierarchie diskretizací

Contents

Introduction

Many real-world physical phenomena, such as the movement of viscous fluids or heat transfer, are governed by (partial) differential equations. Understanding these phenomena requires knowing the solutions to the (partial) differential equations. However, finding exact solutions is often challenging or even impossible due to the complexity of the governing equations. Therefore, numerical mathematics aims to find approximate solutions that are sufficiently accurate, allowing us to better understand the world around.

Multigrid methods are among the most effective iterative methods for the numerical solution of partial differential equations (PDEs). In this thesis, we focus on Poisson's equation as a model problem and obtain its discretization by the finite difference method. The discretization of PDEs usually yields large algebraic systems of linear equations.

While solving these systems, various iterative methods can struggle to find an accurate approximation efficiently. We examine the underlying cause of the slow convergence observed in relaxation methods, such as Jacobi or Gauss-Seidel. They effectively reduce oscillating parts of the error but are inefficient in reducing smooth error components. This is called the smoothing property of the methods. Smooth components of the error can be successfully represented on a coarse grid (a grid with lower resolution, i.e., a grid with fewer degrees of freedom). Solving the smaller system corresponding to the discretisation of the problem on the coarser grid gives an error approximation, which may significantly improve the approximate solution on the finer grid computed by the relaxation method. Multigrid methods are based on combining relaxation methods with correction on a coarser grid.

We establish the mathematical foundation of multigrid methods, employing multiple grid levels of varying coarseness within the computational domain to iteratively solve the system at different resolutions and refine the fine-grid approximate solution. This approach leverages coarse and fine-grid computations, resulting in faster convergence rates.

The thesis is structured as follows. The chapter 1 explores Poisson's equation, a common partial differential equation (PDE) in physics and engineering. Through the analytical view, we illustrate the necessity of numerical methods in solving PDEs. The chapter is based on [1, Chapter 2.2.]. In chapter 2, we focus on the finite difference method, which is our primary tool for discretising partial differential equations into systems of algebraic equations. We present its derivation and explore its implementation. Chapter two follows [2, Chapter 1]. The chapter 3 introduces iterative methods as a mean to approximate solutions for systems of linear equations. We specifically delve into the Jacobi and Gauss-Seidel methods, discussing their properties. These relaxation methods serve as a foundation for Multigrid methods. The chapter draws primarily from [3, Chapter 4.1] and [2, Chapter 2]. In chapter 4, the thesis presents the essential components of multigrid methods, including smoothing processes, restriction, and interpolation techniques for transferring between grids. Practical examples illustrate their roles and interactions within the multigrid framework. The chapter 5 presents several multigrid schemes, such as the V-cycle and W-cycle. Algorithmic details are discussed. Chapters 4 and 5 are based on [2, Chapters 1-3]. An introductory theoretical analysis of multigrid methods is presented in chapter 6, discussing their general formulation, convergence properties, and mathematical conditions for convergence. This chapter serves as a bridge between practical implementation and theoretical understanding. The chapter follows [4, Chapter 4]. Finally, chapter 7 consists of numerical experiments designed to empirically test the effectiveness and convergence speed of multigrid methods in various settings.

1 Poisson's Equation

In this chapter, we formulate Poisson's equation, a partial differential equation used later to demonstrate the principles of multigrid methods.

Let $\Omega \subseteq \mathbb{R}^d$ be an open and bounded set, $d = \{1, 2, 3\}$ is the dimension of the problem. For function $u \in C^2$, $u : \Omega \to \mathbb{R}$, define the Laplace operator Δ as

$$\Delta u = \text{div grad } u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \cdots + \frac{\partial^2 u}{\partial x_d^2}.$$

Given $f : \Omega \to \mathbb{R}$, the partial differential equation seeking for $u : \Omega \to \mathbb{R}$

$$-\Delta u = f, \qquad \text{in } \Omega \tag{1.1}$$

is called the Poisson's equation. Laplace's equation is a special case of (1.1) where the right-hand side function, also called the source term, is zero, i.e.,

$$-\Delta u = 0. \tag{1.2}$$

The problem is fully determined with proper boundary conditions. In the thesis, we consider Dirichlet boundary conditions for simplicity. Given $u_0 : \partial\Omega \to \mathbb{R}$ we require additionally to (1.1) that

$$u = u_0 \qquad \text{on } \partial\Omega. \tag{1.3}$$

1.1 Poisson's and Laplace's Equation in Physics

Poisson's equation or Laplace's equation appear in a wide variety of physical contexts. In a typical interpretation, u denotes the density of some quantity (e.g. chemical concentration) in equilibrium.

Let us denote V any volume within Ω, ∂V the boundary of the volume V, $\mathbf{n}$ the normal vector of the boundary and $d\vec{\mathbf{S}} = \mathbf{n} dS$ the surface element. If $\mathbf{F}$ is the flux density, then assuming the equilibrium, which implies that the net flux of u through ∂V equals zero, the Gauss-Ostrogradsky theorem [1, Section 2.2] yields

$$0 = \int_{\partial V} \mathbf{F} \; d\vec{\mathbf{S}} = \int_V \text{div}\mathbf{F} \; dV.$$

Since V was arbitrary,

$$\text{div}\mathbf{F} = 0$$

in Ω. If $\mathbf{F}$ is proportional to $-\text{grad}(u)$, then

$$-\text{div grad } u = -\Delta u = 0.$$

In electromagnetism, $\mathbf{F}$ is the vector of the electric field, and u is the electrostatic potential, typically denoted by $\mathbf{E}$, φ, respectively. Then, for an electric field in a vacuum it holds

$$\Delta\varphi = -\frac{\rho}{\varepsilon_0}.$$

Outside the electric charge, where $\rho = 0$, we have

$$\Delta\varphi = 0.$$

In hydrodynamics, aerodynamics or fluid mechanics, Laplace's equation represents the irrotational fluid flow, where the unknown function u, usually denoted by ϕ, is the scalar potential.

The behaviour of the gravitational field is given by

$$\Delta\phi = 4\pi G\rho,$$

where ϕ is the gravitational potential, G is gravitational constant and ρ is the mass density.

The heat transfer is typically modelled by the time-dependent heat equation

$$\frac{\partial u}{\partial t} - \Delta u = f. \tag{1.4}$$

When the solution of (1.4) eventually reaches equilibrium, then

$$\frac{\partial u}{\partial t} = 0,$$

and the heat transfer u satisfies Poisson's equation.

1.2 Analytical Solution

Solution to (1.1) with Dirichlet boundary condition (1.3) can be sought analytically. This requires a fundamental solution to Laplace's equation (1.2)

$$\Phi(x) = \begin{cases} -\frac{|x|}{2} & d = 1, \\ -\frac{1}{2\pi}\ln(x) & d = 2, \\ \frac{1}{4\pi|x|} & d = 3, \end{cases}$$

which can be obtained using radial symmetry [1, Section 2.2]. The fundamental solution is radially symmetric due to the rotational invariance of Laplace's equation (1.2) [1, Section 2.2]. Now denote shifted fundamental solution as $\Phi_x(y) = \Phi(y-x)$ and the correction function for a fixed $x \in \Omega$ satisfying

$$\Phi^*_x(y) = \begin{cases} \Delta\Phi^*_x(y) = 0 & \text{in } \Omega, \\ \Phi^*_x(y) = \Phi_x(y) & \text{on } \partial\Omega. \end{cases}$$

Then define Green's function as $G_x = \Phi_x(y) - \Phi^*_x(y)$. Using G_x, we can express the analytical solution of the Poisson's equation in the integral form

$$u(x) = -\int_\Omega G_x \cdot f - \int_{\partial\Omega} u_0 \cdot \nabla G_x d\vec{\mathbf{S}}, \tag{1.5}$$

see, e.g., [5, Section 9.6].

In practice, the specific form for the source term f or the boundary condition u_0 simplifies finding the analytical solution [6, Chapter 7.2.2]. Although, for arbitrarily shaped domains, the conformal mappings are a potential solution method, they may be difficult to describe mathematically [7, Chapter 4].

Therefore, in many real-world scenarios, in contrast to idealised scenarios, analytic solutions are only sporadically found as the problems involve complex geometries, boundary conditions [8, p.5] and right-hand side functions [9, p.1]. Therefore (1.5) in these conditions may be unsolvable and numerical methods must be used to approximate the solution.

2 Finite Difference Method

For practical computations, the infinite-dimensional problem (1.1) has to be discretised, i.e. transformed into a finite-dimensional problem. The transformation could be done by a finite difference method, where the second space derivatives are approximated using a second-order central difference. Central difference is derived from the Taylor polynomial using uniform discretisation of the domain Ω.

2.1 One Dimension

Divide the interval $\Omega = [0, 1]$ into n uniform subintervals in one dimension. That means the division is $0 = x_0 < x_1 < \cdots < x_n = 1$ where $x_k = k \cdot h$, $k = 0, \ldots, n$, leaving $h = 1/n$ as the step size. A general interval $\Omega = [a, b]$ can be then obtained by scaling $x'_i = x_i(b - a) + a$. Denote the function values in interval points as $f(x_k) = f_k$, $k = 1, \ldots, n - 1$, an approximation of the exact solution in these points as $u_k \approx u(x_k)$, $k = 1, \ldots, n-1$, with $u(x_0) = u(a) = u(x_n) = u(b) = 0$, representing the zero Dirichlet condition. Then the approximate solution can be written as a vector $\mathbf{u} = (u_1, \ldots, u_{n-1})^{\mathrm{T}}$.

First show that the approximation of the second derivative in a one-dimensional problem can be given by the formula

$$u_{xx}(x_j) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2},$$

which is derived from the Taylor polynomial.

Denote the derivative of the function u with respect to the variable x as u_x. Analogously, denote the second derivative as u_{xx}. From the Taylor expansion at $x_k + h$ and $x_k + h$

$$\begin{aligned} u(x_k + h) &= u(x_k) + u_x(x_k) \cdot h + u_{xx}(x_k) \cdot \frac{h^2}{2} + O(h^3) \\ u(x_k - h) &= u(x_k) - u_x(x_k) \cdot h + u_{xx}(x_k) \cdot \frac{h^2}{2} + O(h^3) \end{aligned}$$

by summing, we get

$$\begin{aligned} u(x_k + h) + u(x_k - h) &= 2u(x_k) + u_{xx}(x_k) \cdot h^2 + O(h^3) \\ u_{xx}(x_k) &= \frac{u(x_k + h) - 2u(x_k) + u(x_k - h)}{h^2} + O(h) \end{aligned}$$

hence

$$u_{xx}(x_k) \approx \frac{u_{k-1} - 2\ u_k + u_{k+1}}{h^2}.$$

The formulation of the approximation of the problem (1.1) with (1.3) in one dimension

$$-u_{xx}(x_j) = f(x_j), \quad u(a) = u(b) = 0$$

is therefore

$$\begin{gathered} \frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = f_j, \qquad j = 1, 2, \ldots, n, \\ u_0 = u_n = 0. \end{gathered} \tag{2.1}$$

2.2 Two Dimensions

In two dimensions, discretize a rectangular domain $\Omega = (0,1)\times(0,1)$ into a grid $m\times n$.Denote $h_x = 1/m$, $h_y = 1/n$ and define grid points as $(x_i, y_k) = (i\cdot h_x, k\cdot h_y)$, $i = 1,\dots,m-1$, $k = 1,\dots,n-1$. Let $u_{i,k} \approx u(x_i, y_k)$ represent the approximate solution at the grid points. Using the same reasoning as in one dimension,

$$\begin{aligned} u_{xx}(x_i, y_k) &= \frac{u(x_i - h_x, y_k) - 2\ u(x_i, y_k) + u(x_i + h_x, y_k) + O(h_x^3)}{h_x^2} \\ u_{yy}(x_i, y_k) &= \frac{u(x_i, y_k - h_y) - 2\ u(x_i, y_k) + u(x_i, y_k + h_y) + O(h_y^3)}{h_y^2}. \end{aligned}$$

Substituting the derivatives in Poisson's equation (1.1) by second-order finite differences leads to

$$\begin{aligned} \frac{-u_{i-1,k} + 2\ u_{i,k} - u_{i+1,k}}{h_x^2} + \frac{-u_{i,k-1} + 2\ u_{i,k} - u_{i,k+1}}{h_y^2} &= f(x_i, y_k) \\ \text{for} \quad i = 1,\dots,m-1, \quad k = 1,\dots,n-1 & \end{aligned} \tag{2.2}$$

and

$$u_{0,k} = u_{i,0} = u_{m,k} = u_{i,n} = 0 \quad \text{for} \quad 0 \le i \le m,\ 0 \le k \le n$$

as boundary conditions.

2.3 Three Dimensions

In three dimensions, for ease of presentation, we consider a cubic domain $\Omega = (0,1)\times(0,1)\times(0,1)$ discretized on a uniform grid, i.e. a grid of $n\times n\times n$.Denoting the grid spacing as $h = 1/n$, the grid points are $(x_i, y_k, z_l) = (i\cdot h, k\cdot h, l\cdot h)$, where $i, k, l = 1,\dots,n-1$. To represent the approximate solution at the grid points, denote $u_{i,k,l} \approx u(x_i, y_k, z_l)$ and $f(x_i, y_k, z_l) = f_{ikl}$. By following the same reasoning as above, one can proceed with the numerical approximation of the derivatives

$$\begin{aligned} u_{xx}(x_i, y_k, z_l) &= \frac{u(x_i - h, y_k, z_l) - 2\ u(x_i, y_k, z_l) + u(x_i + h, y_k, z_l) + O(h^3)}{h^2} \\ u_{yy}(x_i, y_k, z_l) &= \frac{u(x_i, y_k - h, z_l) - 2\ u(x_i, y_k, z_l) + u(x_i, y_k + h, z_l) + O(h^3)}{h^2} \\ u_{zz}(x_i, y_k, z_l) &= \frac{u(x_i, y_k, z_l - h) - 2\ u(x_i, y_k, z_l) + u(x_i, y_k, z_l + h) + O(h^3)}{h^2}. \end{aligned}$$

Substituting the derivatives in Poisson's equation (1.1) by second-order finite differences and using (1.3) leads to

$$\begin{aligned} \frac{-u_{i-1,k,l} - u_{i,k-1,l} - u_{i,k,l-1} + 6\ u_{i,k,l} - u_{i+1,k,l} - u_{i,k+1,l} - u_{i,k,l+1}}{h^2} &= f_{ikl} \\ u_{0,k,l} = u_{i,0,l} = u_{i,k,0} = u_{n,k,l} = u_{i,n,l} = u_{i,k,0} = 0 \quad \text{for} \quad i,k,l \in \{0,\dots n\}. & \end{aligned} \tag{2.3}$$

2.4 Matrix Notation

We now present matrix representations of the linear algebraic systems (2.1), (2.2) and (2.3).

In one dimension, the system looks as follows

$$\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} \end{bmatrix}.$$

In two dimensions, we get the system with $(m-1)(n-1)$ unknowns. Denote the unknowns on the ith row of the grid as $\mathbf{w}_j = (u_{j,1}, u_{j,2}, \ldots, u_{j,n-1})^{\mathrm{T}}$ and the values of the function f on the ith row of the grid as

$$\mathbf{F}_j = (f(x_j, y_1), f(x_j, y_2), \ldots, f(x_j, y_{n-1}))^{\mathrm{T}}.$$

The system of the equations in two dimensions is then given by block tridiagonal matrix in the form

$$\begin{bmatrix} B & -A & & \\ -A & B & \ddots & \\ & \ddots & \ddots & -A \\ & & -A & B \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{m-1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_{m-1} \end{bmatrix},$$

where A is a $(n-1) \times (n-1)$ scaled identity matrix

$$A = \frac{1}{h_x^2}\, I_{n-1}$$

and B is a $(n-1) \times (n-1)$ tridiagonal matrix

$$B = \frac{1}{h_y^2}\begin{bmatrix} 2 \cdot \frac{h_x^2+h_y^2}{h_x^2} & -1 & & \\ -1 & 2 \cdot \frac{h_x^2+h_y^2}{h_x^2} & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \cdot \frac{h_x^2+h_y^2}{h_x^2} \end{bmatrix}.$$

In three dimensions, the system with $(n-1)^3$ unknowns. Denote by

$$\mathbf{w}_j = (u_{j,1,1}, u_{j,1,2}, \ldots, u_{j,1,n-1}, u_{j,2,1}, u_{j,2,2}, \ldots, u_{j,2,n-1} \ldots, u_{j,n-1,n-2}, u_{j,n-1,n-1})^{\mathrm{T}}$$

the unknowns for a fixed x-coordinate and the values of the function f as

$$\mathbf{F}_j = (f_{j,1,1}, f_{j,1,2}, \ldots, f_{j,1,n-1}, f_{j,2,1}, f_{j,2,2}, \ldots, f_{j,2,n-1} \ldots, f_{j,n-1,n-2}, f_{j,n-1,n-1})^{\mathrm{T}}$$

where $f_{i,j,k} = f(x_i, y_j, z_k)$. The problem (2.3) is then represented by

$$\frac{1}{h^2}\begin{bmatrix} D & -I_{(n-1)^2} & & \\ -I_{(n-1)^2} & D & \ddots & \\ & \ddots & \ddots & -I_{(n-1)^2} \\ & & -I_{(n-1)^2} & D \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_{n-1} \end{bmatrix},$$

where D is a $(n-1)^2 \times (n-1)^2$ block tridiagonal matrix

$$D = \begin{bmatrix} E & -I_{n-1} & & \\ -I_{n-1} & E & \ddots & \\ & \ddots & \ddots & -I_{n-1} \\ & & -I_{n-1} & E \end{bmatrix}, \qquad E = \begin{bmatrix} 6 & -1 & & \\ -1 & 6 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 6 \end{bmatrix}.$$

with E a $(n-1) \times (n-1)$ tridiagonal matrix.

3 Iterative Methods

The solution of the system of linear equations $Au = f$ cannot often be found as the system is too large or the exact solution does not exist. Even if the exact solution is obtainable, a less expensive approximation to the solution may be sufficient in some cases.

Iterative methods aim to approximate the solution of a problem by improving an initial guess $u^{(0)}$ through a sequence of iterations $u^{(i)} \to u$, $i = 0, 1, \dots, k$. The method iterates until the error between the current approximation and the actual solution reaches a predefined tolerance.

3.1 Error, Residual and Residual Equation

Let $A \in \mathbb{R}^{n\times n}$, $f \in \mathbb{R}^n$, then

$$Au = f \tag{3.1}$$

represents the system of n linear equations with the exact solution $u \in \mathbb{R}^n$. Given $u^{(i)}$, a computed approximation to u, there are two important measures of $u^{(i)}$ as an approximation to u. The (algebraic) error is given by

$$e^{(i)} = u - u^{(i)}.$$

The error is also a vector, and a proper vector norm can be used to measure its magnitude. Commonly used norms for this purpose are the maximum and Euclidean norms [2, Chapter 2].

The error is typically impossible to be computed. Another measure is necessary to evaluate the quality of the approximate solution. A computable vector is the residual given by

$$r^{(i)} = f - Au^{(i)}.$$

It should be noted that a small residual does not necessarily indicate a small error.

An important equation that shows the relationship between the error and the residual is the residual equation

$$Ae^{(i)} = r^{(i)},$$

which is obtained as follows

$$Ae^{(i)} = A(u - u^{(i)}) = f - Au^{(i)} = r^{(i)}.$$

In other words, given an approximation $u^{(i)}$, we can search for the solution (or new approximation) by solving the problem with the same matrix A and the residual $r^{(i)}$ as the right-hand side. Then, if $e^{(i)}$ is the computed correction, we get the solution as $u = u^{(i)} + e^{(i)}$.

3.2 Relaxation Methods

Relaxation methods provide a sequence of approximations $u^{(i)} \to u$, $i = 0, 1, \ldots, k$ through an iterative scheme in a general form

$$Mu^{(i+1)} = Nu^{(i)} + f. \tag{3.2}$$

Matrices $M, N \in \mathbb{R}^{n \times n}$, where M is a regular matrix, are obtained by splitting the matrix A as $A = M - N$. In such a case, we can also express the relationship between the error in subsequent steps as

$$e^{(i+1)} = M^{-1}Ne^{(i)}.$$

This relationship follows from the definition of the error

$$\begin{aligned} -Mu^{(i+1)} &= -M(u - e^{(i+1)}) = -Nu^{(i)} - f = -N(u - e^{(i)}) - f \\ Me^{(i+1)} &= Mu - Nu + Ne^{(i)} - f = Au - f + Ne^{(i)} = Ne^{(i)}. \end{aligned}$$

Consider the matrix decomposition

$$A = D - L - U,$$

where D is the diagonal part of A, $-U$ is the strict upper triangular part of A, and $-L$ is the strict lower triangular part of A.

The choice $M = D$ and $N = (L + U)$ in (3.2) leads to the Jacobi method. This method is of the form

$$u^{(i+1)} = D^{-1}(L + U)u^{(i)} + D^{-1}f. \tag{3.3}$$

With the iteration matrix $R_J = D^{-1}(L + U)$, we can rewrite the iteration scheme as

$$u^{(i+1)} = R_J u^{(i)} + D^{-1}f,$$

and for the error it holds

$$e^{(i+1)} = R_J e^{(i)}.$$

Weighted Jacobi method as a simple but important modification of the Jacobi method. The approximation in the new iteration results from a combination of the previous approximation and the new approximation (3.3) provided by the Jacobi method. For a parameter $\omega \in \mathbb{R}$, this is expressed as

$$\begin{aligned} u^{(i+1)} &= (1 - \omega)u^{(i)} + \omega(D^{-1}(L + U)u^{(i)} + D^{-1}f) \\ &= u^{(i)} - \omega(I - D^{-1}(L + U))u^{(i)} + \omega D^{-1}f \\ &= u^{(i)} - \omega D^{-1}(D - L - U)u^{(i)} + \omega D^{-1}f \\ &= (I - \omega D^{-1}Au^{(i)}) + \omega D^{-1}f. \end{aligned}$$

Denote $R_{J_\omega} = I - \omega D^{-1}A$, then

$$u^{(i+1)} = R_{J_\omega}u^{(i)} + \omega D^{-1}f \tag{3.4}$$

and

$$e^{(i+1)} = R_{J_\omega} e^{(i)}.$$

The optimal parameter for the weighted Jacobi method is $\omega = \frac{4}{5}$ [3, Chapter 13.2.2].

Another relaxation method is the Gauss-Seidel method corresponding to the choice $M = D - L$ and $N = U$. Then the iteration scheme is given by

$$u^{(i+1)} = (D - L)^{-1} U u^{(i)} + (D - L)^{-1} f, \tag{3.5}$$

and the relation

$$e^{(i+1)} = (D - L)^{-1} U e^{(i)}$$

holds for the error.

Jacobi, Gauss-Seidel, or relaxation methods in general can be effective in finding approximate solutions to systems of linear equations. However, the convergence of these methods can be slow, particularly for more complex systems, and alternative techniques such as multigrid or preconditioning may be required to accelerate the convergence [3, Chapter 9]

4 Multigrid Components

The relaxation methods introduced in the previous previous chapter 3 have some limitations for practical use. As we will illustrate in section 4.1, they tend to reduce error oscillations while leaving smooth components relatively unchanged. This is called a smoothing property and can slow down the convergence. In multigrid methods, this deficiency is mitigated by solving problems on a hierarchy of grids and transferring solutions between fine and coarse grids.

The transition between fine and coarse grids involves two operations: interpolation and restriction, which will be introduced in section 4.2. Interpolation maps vectors from a coarse to a fine grid, while restriction maps vectors from a fine to a coarse grid. Let us explore their mathematical formulations and implications in more detail.

4.1 Convergence and Smoothing Property

In order to examine the convergence of relaxation methods, it is convenient to focus on the homogeneous linear system $Au = 0$ and start the iteration process with a proper initial guess. For this system, the exact solution $u = 0$ is already known. Therefore, the difference between the approximation $u^{(i)}$ and the exact solution is known and equal to $-u^{(i)}$.

Consider a one-dimensional problem as stated in section 2.4 with $f = 0$ and $n = 500$ and use Jacobi method with an initial guess consisting of the functions in general form

$$v_j = \sin\left(\frac{jk\pi}{n}\right), \quad 0 \leq j \leq n, \quad 1 \leq k \leq n-1, \tag{4.1}$$

called the Fourier modes. The term k is the frequency. The Figure 4.1 shows that low values of k result in smooth waves, while high values of k lead to oscillatory waves.

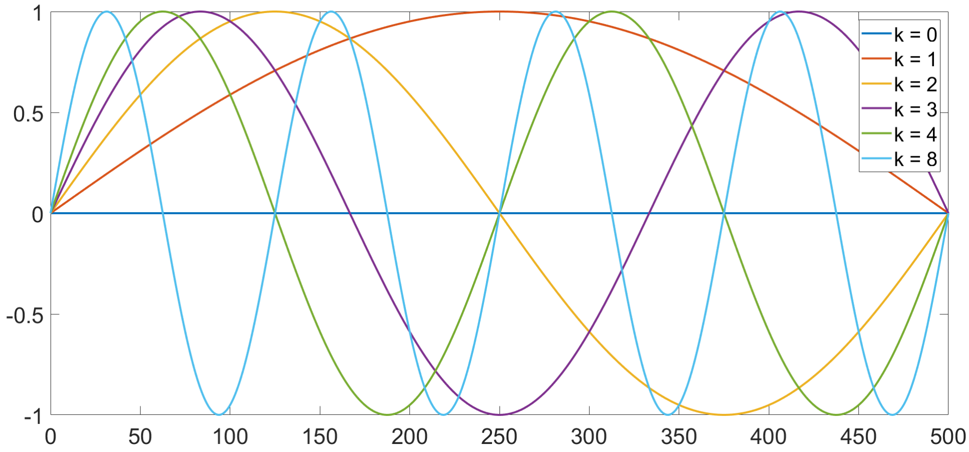


Figure 4.1 Fourier modes for multiple frequencies k.

Now apply the relaxation scheme (3.3) with $u^{(0)} = v_j$ and observe the magnitude of error in maximum norm, i.e.

$$||e^{(i)}||_\infty = ||u^{(i)}||_\infty = \max_j |u_j^{(i)}|.$$

As we can see in Figure 4.2 and Figure 4.3, more oscillatory initial guess converges faster, whereas the smooth waves converge slowly, both in relative error norm (that is accessible to us thanks to the choice of the right-hand side) and relative residual norm. When we plot, how many iterations of the method are needed to reduce the error by three orders of magnitude, see Figure 4.4, we get an analogous information.

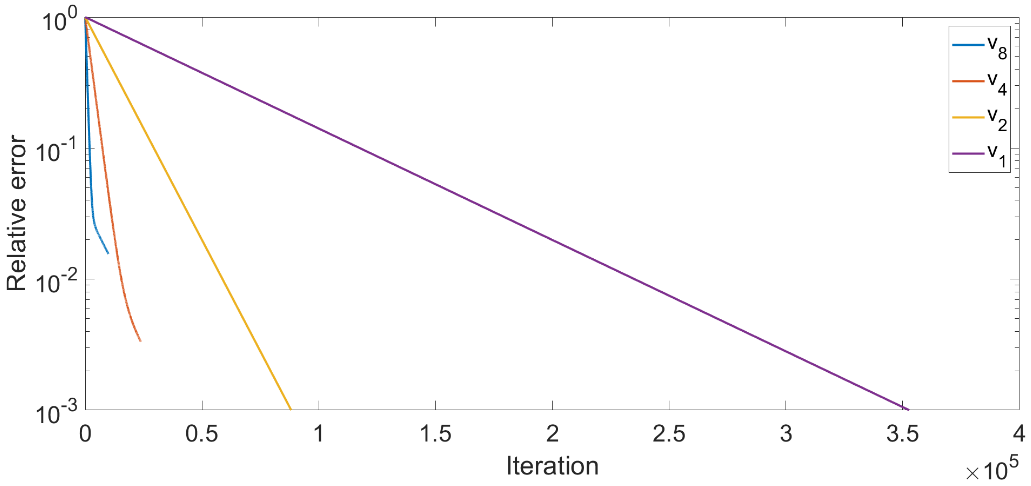


Figure 4.2 The relative maximum norm of the error in iterations of the Jacobi method. The initial error $e^{(0)}$ is equal to Fourier modes v_1, v_2, v_4 and v_8 from (4.1) respectively.

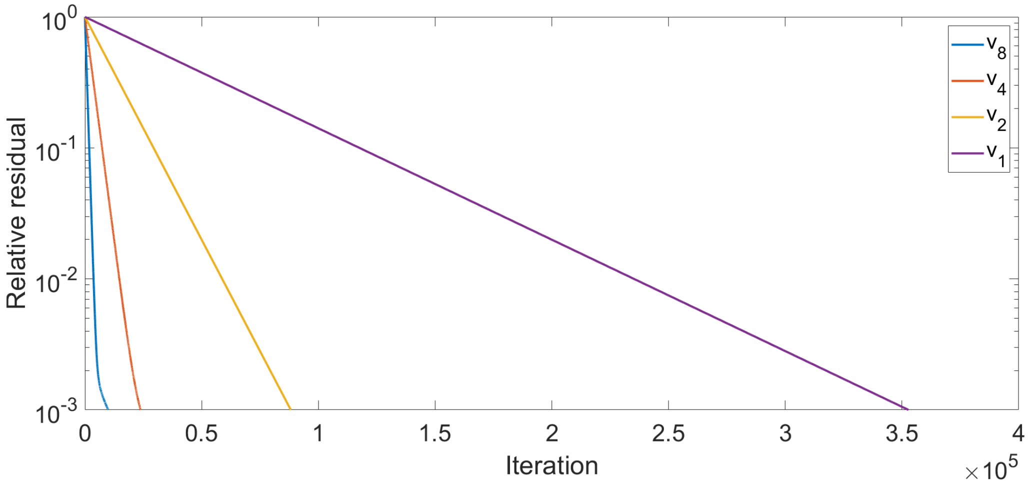


Figure 4.3 The relative maximum norm of the residual in iterations of the Jacobi method. The initial error $e^{(0)}$ is equal to Fourier modes v_1, v_2, v_4 and v_8.

This experiment illustrates that the iterative scheme effectively eliminates the oscillatory components of the error while leaving the low-frequency or smooth components relatively unchanged. This is called a smoothing property of a relaxation method, and many relaxation schemes possess this property. The smoothing property is a limitation of standard relaxation methods. However, this limitation can be overcome, and the remedy is one of the pathways to multigrid [2, Chapter 2].

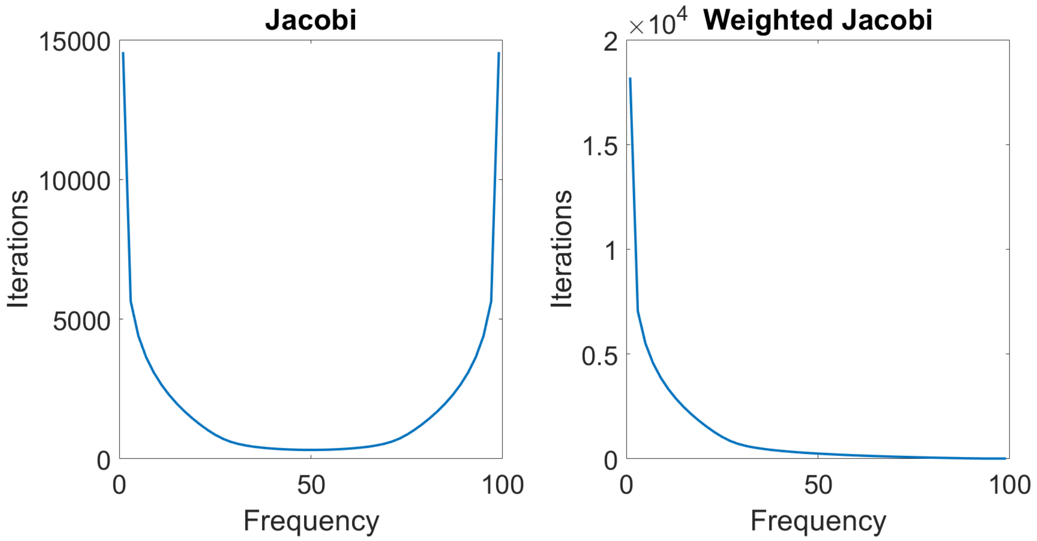


Figure 4.4 Number of iterations necessary to reduce error by three orders of magnitude for Jacobi (left) and Weighted Jacobi with $\omega = \frac{4}{5}$ (right) with initial guess $u^{(0)}$ equal to Fourier modes $v_1, v_2, v_3, \dots, v_n$.

4.2 Fine and Coarse Grid

For large-scale system, obtaining the solution may be challenging and time inefficient as iterative methods may eventually reach a point of stagnation. Denote Ω_h the discretization of the model problem, representing our fine grid, where the approximate solution is sought, and introduce a coarse grid Ω_{2h} with a double step size (resulting in half the points in 1D, a quarter in 2D, and one-eighth in 3D). The model problem then can be discretised on Ω_{2h}, where solution is cheaper to compute. This coarse-grid solution can then be projected onto the fine grid and used as a correction of the approximate solution obtained on the fine grid. This process may demonstrate efficacy under certain conditions.

In this chapter, we describe the process of transitioning between grids. This process is called interpolation when transitioning from a coarse grid to a fine grid and restriction for transition from the fine and coarse grid. Finally, the correction using the coarse-grid solution is being discussed.

4.2.1 Interpolation

There are several interpolation methods. For multigrid purposes, the simplest one - linear interpolation is usually used ([10, p. 620]; [11, p. 15]; [12, p. 1]; [2, Chapter 2]; [13, p. 19]). For describing the linear interpolation in one dimension, let us denote the k-th component of the vector $\mathbf{v}_{2h}$ on a coarse grid Ω_{2h} as $(v_{2h})_k$ and the j-th component of the vector $\mathbf{v}_h$ on a fine grid Ω_h as $(v_h)_j$. Then the linear interpolation of vector $\mathbf{v}_{2h}$ to $\mathbf{v}_h$ is given by the relation

$$\begin{aligned} (v_h)_{2j} &= (v_{2h})_j, \\ (v_h)_{2j+1} &= \frac{1}{2}\left((v_{2h})_j + (v_{2h})_{j+1}\right), \quad 0 \le j \le \frac{n}{2} - 1. \end{aligned} \tag{4.2}$$

The components of the vector $\mathbf{v}_{2h}$ become the even components of the vector $\mathbf{v}_h$,

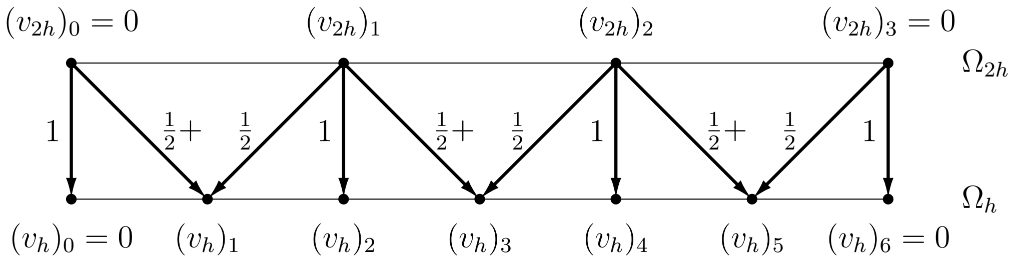


Figure 4.5 The visual representation of interpolation relation (4.2).

and the odd components are obtained as their arithmetic averages. We can also write this operation using the matrix operator $I_h^{2h} : \mathbb{R}^{\frac{n}{2}-1} \to \mathbb{R}^{n-1}$. This is a $(n-1) \times \left(\frac{n}{2} - 1\right)$ matrix, for example for $n = 6$

$$I_h^{2h}\mathbf{v}_{2h} = \frac{1}{2}\begin{bmatrix}1 & 0\\ 2 & 0\\ 1 & 1\\ 0 & 2\\ 0 & 1\end{bmatrix}\begin{bmatrix}(v_{2h})_1\\ (v_{2h})_2\end{bmatrix} = \begin{bmatrix}(v_h)_1\\ (v_h)_2\\ (v_h)_3\\ (v_h)_4\\ (v_h)_5\end{bmatrix} = \mathbf{v}_h.$$

In two dimensions, the linear interpolation of vector

$$\begin{aligned}\mathbf{v}_{2h} &= \left((\mathbf{w}_{2h})_1, \ldots, (\mathbf{w}_{2h})_{m-1}\right)^{\mathrm{T}} = \\ &= \left((v_{2h})_{1,1}, (v_{2h})_{1,2}, \ldots, (v_{2h})_{1,n-1}, \ldots, (v_{2h})_{m-1,1}, \ldots, (v_{2h})_{m-1,n-1}\right)^{\mathrm{T}}\end{aligned}$$

from Ω_{2h} to a vector $\mathbf{v}_h$ from Ω_h is given by the following relations

$$\begin{aligned}(v_h)_{2i,2j} &= (v_{2h})_{i,j}, \\ (v_h)_{2i+1,2j} &= \frac{1}{2}\left((v_{2h})_{i,j} + (v_{2h})_{i+1,j}\right), \\ (v_h)_{2i,2j+1} &= \frac{1}{2}\left((v_{2h})_{i,j} + (v_{2h})_{i,j+1}\right), \\ (v_h)_{2i+1,2j+1} &= \frac{1}{4}\left((v_{2h})_{i,j} + (v_{2h})_{i+1,j} + (v_{2h})_{i,j+1} + (v_{2h})_{i+1,j+1}\right)\end{aligned}$$

for $0 \le i \le \frac{m}{2} - 1, 0 \le j \le \frac{n}{2} - 1$. This operation has the matrix form $I_h^{2h} : \mathbb{R}^{(\frac{m}{2}-1)\cdot(\frac{n}{2}-1)} \to \mathbb{R}^{(m-1)\cdot(n-1)}$, analogously to one dimension.

In three dimensions, the linear interpolation is given by

$$\begin{aligned}
(v_h)_{2i,2j,2k} &= (v_{2h})_{i,j,k},\\
(v_h)_{2i+1,2j,2k} &= \frac{1}{2}\left((v_{2h})_{i,j,k} + (v_{2h})_{i+1,j,k}\right),\\
(v_h)_{2i,2j+1,2k} &= \frac{1}{2}\left((v_{2h})_{i,j,k} + (v_{2h})_{i,j+1,k}\right),\\
(v_h)_{2i,2j,2k+1} &= \frac{1}{2}\left((v_{2h})_{i,j,k} + (v_{2h})_{i,j,k+1}\right),\\
(v_h)_{2i+1,2j+1,2k} &= \frac{1}{4}\left((v_{2h})_{i,j,k} + (v_{2h})_{i+1,j,k} + (v_{2h})_{i,j+1,k} + (v_{2h})_{i+1,j+1,k}\right),\\
(v_h)_{2i+1,2j,2k+1} &= \frac{1}{4}\left((v_{2h})_{i,j,k} + (v_{2h})_{i+1,j,k} + (v_{2h})_{i,j,k+1} + (v_{2h})_{i+1,j,k+1}\right),\\
(v_h)_{2i,2j+1,2k+1} &= \frac{1}{4}\left((v_{2h})_{i,j,k} + (v_{2h})_{i,j+1,k} + (v_{2h})_{i,j,k+1} + (v_{2h})_{i,j+1,k+1}\right),\\
(v_h)_{2i+1,2j+1,2k+1} &= \frac{1}{8}\Big((v_{2h})_{i,j,k} + (v_{2h})_{i+1,j,k} + (v_{2h})_{i,j+1,k} + (v_{2h})_{i,j,k+1} +\\
&\quad + (v_{2h})_{i+1,j+1,k} + (v_{2h})_{i+1,j,k+1} + (v_{2h})_{i,j+1,k+1} + (v_{2h})_{i+1,j+1,k+1}\Big)
\end{aligned}$$

for $0 \le i,j,k \le \frac{n}{2} - 1$. In analogy to one and two dimensions, this operation has the matrix form $I_h^{2h} : \mathbb{R}^{(\frac{n}{2}-1)\cdot(\frac{n}{2}-1)\cdot(\frac{n}{2}-1)} \to \mathbb{R}^{(n-1)\cdot(n-1)\cdot(n-1)}$.

4.2.2 Restriction

The restriction is used for the transition between Ω_h and Ω_{2h}. There are multiple ways to restrict a vector $\mathbf{v}_h$ to a vector $\mathbf{v}_{2h}$. The simplest one - injection is defined through an operator $R_{2h}^h : \mathbb{R}^{n-1} \to \mathbb{R}^{\frac{n}{2}-1}$, which satisfies the relation

$$(v_{2h})_j = (v_h)_{2j}, \quad 0 \le j \le \frac{n}{2} - 1,$$

i.e., the coarse-grid vector values are obtained directly from the corresponding fine-grid values.

Although injection is simple and fast to compute, a more common restriction operator is the full weighting, see, e.g., [3, Chapter 13.3.2]. This is the operator $R_{2h}^h : \mathbb{R}^{n-1} \to \mathbb{R}^{\frac{n}{2}-1}$, where the coarse-grid vector values are weighted averages of neighbouring fine-grid points.

In one dimension R_{2h}^h is defined as

$$(v_{2h})_j = \frac{1}{4}\left((v_h)_{2j-1} + 2(v_h)_{2j} + (v_h)_{2j+1}\right), \quad 1 \le j \le \frac{n}{2} - 1, \tag{4.3}$$

in two dimensions as

$$\begin{aligned}
(v_{2h})_{i,j} = \frac{1}{16}\Big(&(v_h)_{2i-1,2j-1} + (v_h)_{2i-1,2j+1} + (v_h)_{2i+1,2j-1} + (v_h)_{2i+1,2j+1} +\\
&+ 2(v_h)_{2i-1,2j} + 2(v_h)_{2i+1,2j} + 2(v_h)_{2i,2j-1} + 2(v_h)_{2i,2j+1} +\\
&+ 4(v_h)_{2i,2j}\Big), \qquad 1 \le i \le \frac{m}{2} - 1,\ 1 \le j \le \frac{n}{2} - 1,
\end{aligned}$$

and in three dimensions as

$$\begin{aligned}(v_{2h})_{i,j,k} = \frac{1}{64}\Big(&(v_h)_{2i-1,2j-1,2k-1} + (v_h)_{2i-1,2j-1,2k+1} + (v_h)_{2i-1,2j+1,2k-1} + \\ &+ (v_h)_{2i+1,2j-1,2k-1} + (v_h)_{2i+1,2j+1,2k+1} + (v_h)_{2i+1,2j+1,2k-1} + \\ &+ (v_h)_{2i+1,2j-1,2k+1} + (v_h)_{2i-1,2j+1,2k+1} + 2(v_h)_{2i+1,2j+1,2k} + \\ &+ 2(v_h)_{2i+1,2j,2k+1} + 2(v_h)_{2i,2j+1,2k+1} + 2(v_h)_{2i-1,2j-1,2k} + \\ &+ 2(v_h)_{2i-1,2j,2k-1} + 2(v_h)_{2i,2j-1,2k-1} + 2(v_h)_{2i-1,2j+1,2k} + \\ &+ 2(v_h)_{2i+1,2j-1,2k} + 2(v_h)_{2i-1,2j,2k+1} + 2(v_h)_{2i+1,2j,2k-1} + \\ &+ 2(v_h)_{2i,2j-1,2k+1} + 2(v_h)_{2i,2j+1,2k-1} + 4(v_h)_{2i,2j,2k-1} + \\ &+ 4(v_h)_{2i,2j,2k+1} + 4(v_h)_{2i,2j-1,2k} + 4(v_h)_{2i,2j+1,2k} + \\ &+ 4(v_h)_{2i-1,2j,2k} + 4(v_h)_{2i+1,2j,2k} + 8(v_h)_{2i,2j,2k}\Big)\end{aligned}$$

for $1 \le i, j, k \le \frac{n}{2} - 1$.

For $n = 6$ the operator in one dimension is of the form

$$R_{2h}^h \mathbf{v}_h = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} (v_h)_1 \\ (v_h)_2 \\ (v_h)_3 \\ (v_h)_4 \\ (v_h)_5 \end{bmatrix} = \begin{bmatrix} (v_{2h})_1 \\ (v_{2h})_2 \end{bmatrix} = \mathbf{v}_{2h}.$$

The reason for choosing the full weighting restriction is that

$$R_{2h}^h = 2^d \cdot (I_h^{2h})^T, \tag{4.4}$$

where d is the dimension [3, Chapter 13.3.2]. The relation (4.4) called variational property and it is often used in the proofs of the convergence of the methods.

Figure 4.6 illustrates the interpolation (4.2) and restriction (4.3). Here the vector is composed of four Fourier modes with frequencies $k = 1, 2, 4, 6$, i.e., $v = \frac{1}{4}\left(\sin\left(\frac{j\pi}{12}\right) + \sin\left(\frac{2j\pi}{12}\right) + \sin\left(\frac{4j\pi}{12}\right) + \sin\left(\frac{6j\pi}{12}\right)\right)$.

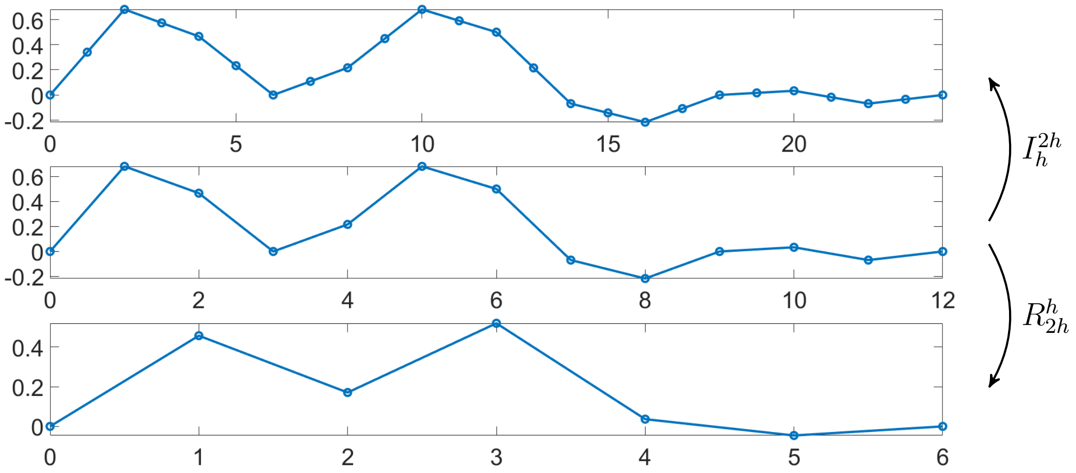


Figure 4.6 Linear interpolation and full-weighting restriction of a vector.

4.2.3 Coarse-grid correction

The suggested approach is to initially relax on the fine grid before stagnation in convergence occurs. Then, the residual equation is used and is transformed onto a coarse grid using the restriction the residual. On a coarse grid, the problem for error is solved. Relaxation on the coarse grid is more cheaper due to fewer unknowns requiring updates. Then, the obtained error is interpolated into a fine grid and used as a correction to refine the approximation from relaxation on the fine grid.

Let Ω_h and Ω_{2h} be the fine and coarse grids, respectively, with the discretized model problem represented by the matrices A_h and A_{2h}. Clearly, solving a system with A_{2h} cannot, in general, replace solving the residual equation with A_h. Therefore, using coarse-grid to correct the error $e_h^{(i)}$ associated with Ω_h can only work for some errors. In particular, coarse grids can represent smooth vectors but are unsuitable for oscillatory ones. This complements the smoothing property of relaxation methods that efficiently eliminate oscillating error components while leaving smoother ones relatively unchanged. The correction scheme employs the residual equation, restricting presumably smooth residual and error on the coarse grid.

Let $u_h^{(i)}$, $e_h^{(i)}$, and $r_h^{(i)}$ represent an approximation, the associated error and residual, respectively. The correction on a coarse grid solves

$$A_{2h}e_{2h}^{(i)} = R_{2h}^{h} r_h^{(i)},$$

where R_{2h}^{h} is the restriction operator mapping Ω_h to Ω_{2h}. An approximation to $e_h^{(i)}$ is then given by interpolating $e_{2h}^{(i)}$ onto Ω_h by I_h^{2h}, the interpolation operator from Ω_{2h} to Ω_h. Altogether, the coarse-grid correction is given by

$$I_h^{2h} e_{2h}^{(i)} = I_h^{2h} A_{2h}^{-1} R_{2h}^{h} r_h^{(i)} \approx e_h^{(i)}.$$

Since R_{2h}^{h} has a non-trivial kernel, the first requirement for an efficient coarse-space correction is that the residual mostly lies outside of the kernel $\mathrm{Ker}(R_{2h}^{h})$. From the construction of the restriction, the kernel contains the oscillatory vectors and the restriction preserves smooth vectors. This is illustrated in Figure 4.7.

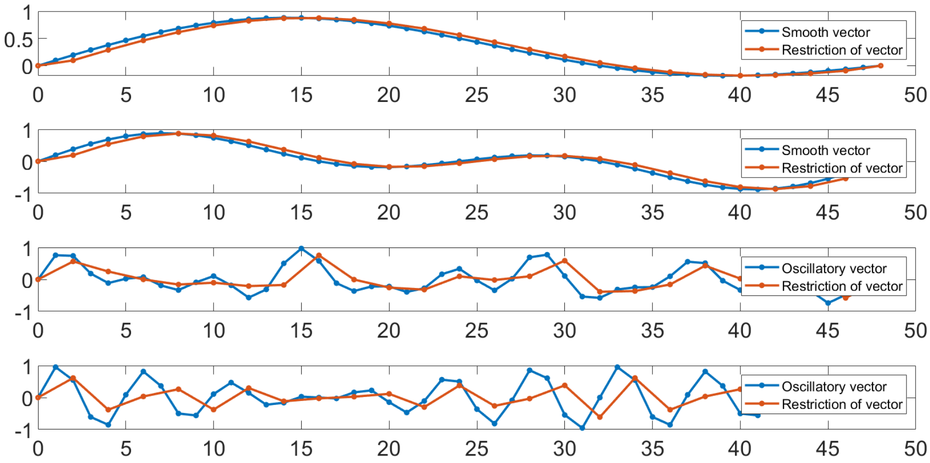


Figure 4.7 Restriction of smooth and oscillatory vectors.

Similarly for the interpolation, in an efficient coarse-space correction $I_h^{2h} e_{2h}^{(i)}$ is close to $e_h^{(i)}$. A necessary condition for that is that $e_h^{(i)}$ is in the range of the interpolation operator I_h^{2h}. Given that the range of the interpolation operator contains smooth vectors, a coarse-space correction can only work for a smooth error $e_h^{(i)}$.

To illustrate the limitations of the coarse-grid correction, compare $e_h^{(i)}$ with $I_h^{2h} A_{2h}^{-1} R_{2h}^h A_h e_h^{(i)}$ for both smooth and oscillating vectors $e_h^{(i)}$. This is done in Figure 4.8.

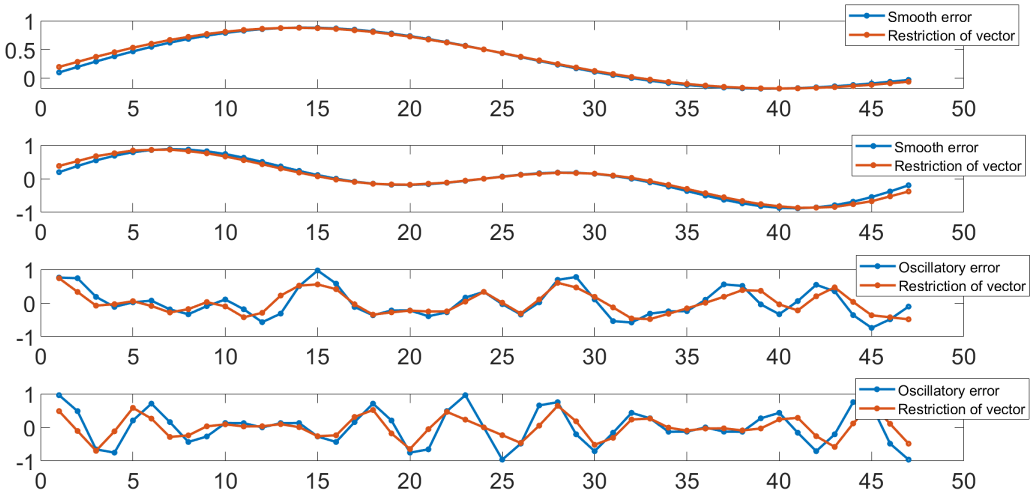


Figure 4.8 Function $I_h^{2h} A_{2h}^{-1} R_{2h}^h A_h e_h^{(i)}$ for smooth and oscillatory errors.

Another observation appears when Fourier modes are transformed between grids, particularly from a fine grid Ω_h to a coarse grid Ω_{2h}. Consider a Fourier mode in 1D on a fine grid (4.1), then its representation on a coarse grid becomes $v_j = \sin\left(\frac{jk\pi}{\frac{n}{2}}\right) = \sin\left(\frac{2jk\pi}{n}\right)$. Hence, the frequency doubles, as $\sin(2x)$ has twice the frequency of $\sin(x)$. This is illustrated in Figure 4.9. Since Fourier modes form a basis, it is possible to express the error in this basis. Suppose the relaxation process starts to stagnate due to the smoothing property, where oscillatory parts of the error have been smoothed out and smooth parts have prevailed. In that case, the error components become more oscillatory when the problem is transferred to a coarse grid. This plays no role if the coarse-grid problem on Ω_{2h} is solved by a direct method. However, if the coarse-grid correction is called recursively, which is typically done when the coarse-grid problem is still large, the observation from Figure 4.9 suggests that some iterations of a relaxation method should be done before the coarse-grid correction on Ω_{4h}.

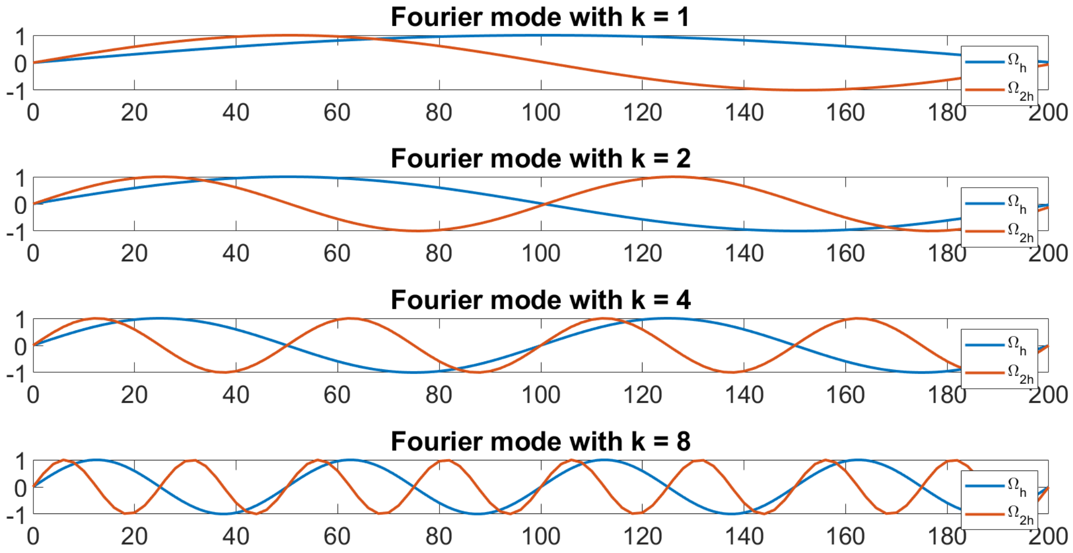


Figure 4.9 Fourier modes with $k = 2$ and $k = 8$ on a fine and coarse grid.

5 Multigrid

In a previous chapter, we introduced the key elements of multigrid methods: smoothing, a transfer between a fine and coarse grid and coarse grid solving. Now combine the introduced components giving a basic multigrid schemes.

5.1 Two-Grid Scheme

The simplest scheme is a two-grid scheme. Therein we assume that the problem on the fine grid is too large to be solved directly, while the problem on the coarse grid can be solved using a direct method. Given matrices A_h, A_{2h}, a vector f_h, a restriction operator R_{2h}^{h} and a interpolation operator I_h^{2h}, the scheme is as follows:

relax on $A_h u_h = f_h$ on Ω_h with initial guess $u_h^{(0)}$ to obtain an approximation $u_h^{(i)}$,

compute the residual $r_h^{(i)} = f - A_h u_h^{(i)}$ and restrict it to the coarse grid as $r_{2h}^{(i)} = R_{2h}^{h} r_h^{(i)}$

solve directly the residual equation $A_{2h} e_{2h} = r_{2h}^{(i)}$ and obtain e_{2h},

interpolate e_{2h} to Ω_h as $\tilde{e}_h = I_h^{2h} e_{2h}$ and correct the approximation obtained on Ω_h as $u_h^{(i)} \leftarrow u_h^{(i)} + \tilde{e}_h$,

relax on $A_h u_h = f_h$ on Ω_h with initial guess $u_h^{(i)}$.

5.2 Another Multigrid Schemes

The two-grid correction scheme relies on the assumption that the residual equation on a coarse grid Ω_{2h} can be solved directly. This cannot often be satisfied and, therefore, other schemes must be considered. This is done by introducing additional grid levels until the coarsest problem is of the size that allows an efficient direct solution.

5.2.1 V-Cycle Scheme

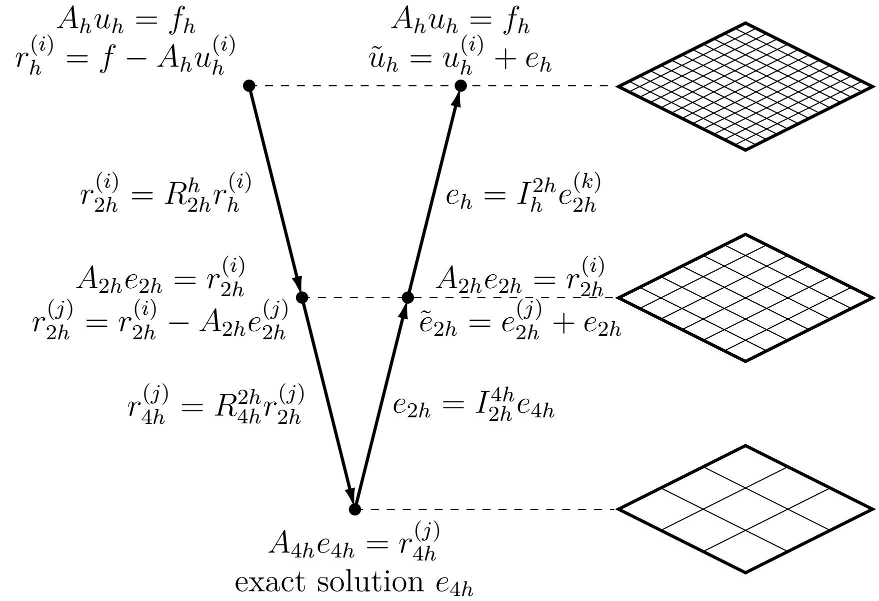
The basic idea behind the V-cycle scheme is to extend Two-Grid Correction Scheme by recursively applying it on finer grids. This gives the following algorithm:

relax on $A_h u_h = f_h$ on Ω_h with some initial guess $u_h^{(0)}$ to obtain an approximation $u_h^{(i)}$, compute the residual $r_h^{(i)} = f - A_h u_h^{(i)}$,

> restrict to Ω_{2h}, relax on $A_{2h} e_{2h} = r_{2h}^{(i)}$ with initial guess $e_{2h}^{(0)} = 0$, to obtain approximation $e_{2h}^{(j)}$, compute the residual $r_{2h}^{(j)} = r_{2h}^{(i)} - A_{2h} e_{2h}^{(j)}$,
>
> > restrict to Ω_{4h}, relax on $A_{4h} e_{4h} = r_{4h}^{(j)}$ with initial guess $e_{4h}^{(0)} = 0$, to obtain approximation $e_{4h}^{(k)}$, compute the residual $r_{4h}^{(k)} = r_{4h}^{(j)} - A_{4h} e_{4h}^{(k)}$,

```
         .
         .
         .
           restrict to $\Omega_{Lh}$ (where $L = 2^p, p \in \mathbb{N}$ and $p+1$ is the number
           of grids used), solve directly $A_{Lh}e_{Lh} = r_{Lh}^{(l)}$ and obtain exact
           solution $e_{Lh}$
         .
         .
         .
      interpolate $e_{8h}$ to $e_{4h}$ and correct $\tilde{e}_{4h} = e_{4h}^{(k)} + e_{4h}$,
      relax on $A_{4h}e_{4h} = r_{4h}^{(j)}$ with initial guess $e_{4h}^{(0)} = \tilde{e}_{4h}$ to obtain $e_{4h}$
    interpolate $e_{4h}$ to $e_{2h}$ and correct $\tilde{e}_{2h} = e_{2h}^{(k)} + e_{2h}$,
    relax on $A_{2h}e_{2h} = r_{2h}^{(i)}$ with initial guess $e_{2h}^{(0)} = \tilde{e}_{2h}$ to obtain $e_{2h}$,
  interpolate $e_{2h}$ to $e_h$ and correct $\tilde{u}_h = u_h^{(i)} + e_h$,
  relax on $A_h u_h = f_h$ with initial guess $u_h^{(0)} = \tilde{u}_h$.
```

The V-cycle algorithm is named due to the order in which the grids are visited, as illustrated in the Figure 5.1.



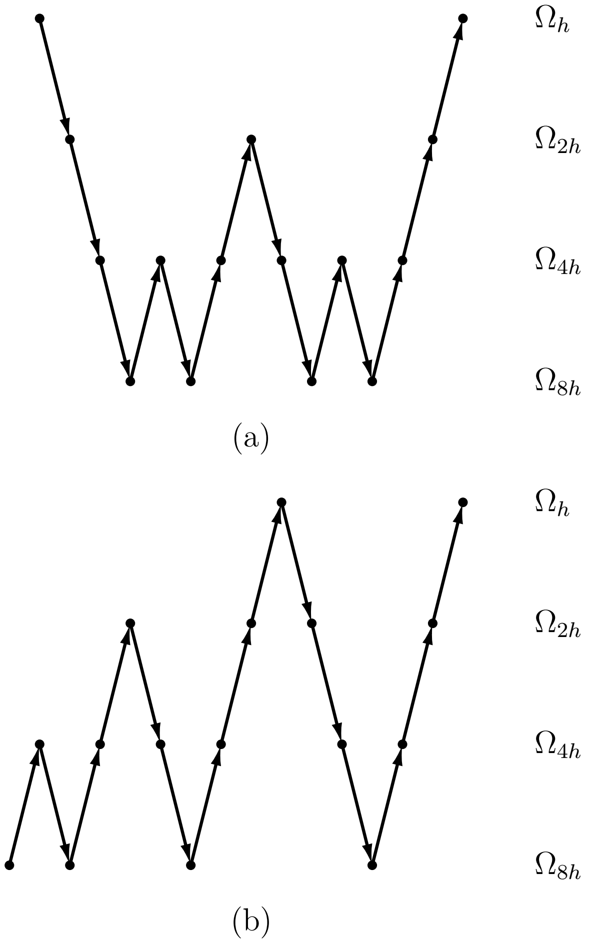
$A_h u_h = f_h$
$r_h^{(i)} = f - A_h u_h^{(i)}$
$A_h u_h = f_h$
$\tilde{u}_h = u_h^{(i)} + e_h$
$r_{2h}^{(i)} = R_{2h}^{h} r_h^{(i)}$
$e_h = I_h^{2h} e_{2h}^{(k)}$
$A_{2h}e_{2h} = r_{2h}^{(i)}$
$r_{2h}^{(j)} = r_{2h}^{(i)} - A_{2h}e_{2h}^{(j)}$
$A_{2h}e_{2h} = r_{2h}^{(i)}$
$\tilde{e}_{2h} = e_{2h}^{(j)} + e_{2h}$
$r_{4h}^{(j)} = R_{4h}^{2h} r_{2h}^{(j)}$
$e_{2h} = I_{2h}^{4h} e_{4h}$
$A_{4h}e_{4h} = r_{4h}^{(j)}$
exact solution e_{4h}

Figure 5.1 The scheme representing a V-cycle.

5.2.2 Full Multigrid V-cycle and W cycle

Other commonly used multigrid schemes are W-cycle and Full Multigrid V-cycle (FMG scheme). For simplicity, we only illustrate these schemes in Figure 5.2 as an analogy to Figure 5.1 corresponding to V-cycle scheme.

As we can see in the Figure 5.2, W-cycle is a modification of the V-cycle scheme where two recursive calls of coarse-space correction are considered instead



(a)

(b)

Figure 5.2 The scheme representing a (a) W-cycle and (b) FMG cycle.

of one as in V-cycle. This increases the computational cost of a single step. On the other hand, more operations (intergrid transfers, smoothing) are done on coarser levels where they are cheaper, with respect to the finest grid.

In contrast to V- and W-cycle, the full multigrid is not an iterative scheme, it is performed only once going from bottom to top. It starts with the direct solution of the problem on the coarsest grid. This solution is then interpolated to the finer grid, where it is used as an initial guess for the relaxation scheme. Then again the coarsest grid is used to correct the solution from the finer grid using the residual equation. This process is repeated as illustrated in Figure 5.2 till the solution on the finest grid is obtained.

6 An Abstract Multigrid Setting and Convergence

This chapter aims to develop an abstract generalisation of the multigrid methods discussed in previous chapters. In the second part, we formulate the algebraic convergence criteria of a symmetric multigrid method.

6.1 An Algebraic Multigrid Formulation

Consider any discretisation of the domain giving a sequence of finite-dimensional spaces $V_0, V_1, \dots, V_k$, with inner products denoted by $(\cdot,\cdot)_i, i = 0, \dots, k$ respectively, satisfying $\dim(V_0) < \dim(V_1) < \cdots < \dim(V_k)$. Consider the following operators:

restriction operator

$$R^i_{i-1} : V_i \to V_{i-1}, \qquad i = 1, \dots, k,$$

interpolation operator

$$I^{i-1}_i : V_i \to V_{i+1}, \qquad i = 0, \dots, k-1,$$

discrete operator

$$A_i : V_i \to V_i, \qquad i = 0, \dots, k.$$

Then, the Multigrid method solves the problem

$$A_k u_k = f_k, \tag{6.1}$$

where $u_k, f_k \in V_k$. This method utilises auxiliary problems at lower levels, represented as $A_i u_i = f_i$ for $i = 0, \dots, k$, with $u_i, f_i \in V_i$. Now, we describe the the multigrid algorithm at level $l > 0$, denoted $MG_l(u^{(0)}_l, f_l)$ with initial guess $u^{(0)}_l$ to solve $A_l u_l = f_l$, as:

pre-smoothing

$$u^{(i+1)}_l = u^{(i)}_l - \mathcal{D}_{l,i+1}(A_l u^{(i)}_l - f_l),\ i = 0, \dots, \kappa_1 - 1, \tag{6.2}$$

restriction

$$f_{l-1} = R^l_{l-1}(A_l u^{(\kappa_1)}_l - f_l), \tag{6.3}$$

coarse-grid correction

for $l - 1 = 0$ we directly solve the system and obtain an exact solution $\tilde{u}_0 = MG_0(u^{(0)}_0, f_{l-1}) = A^{-1}_0 f_0$, where initial guess $u^{(0)}$ is not used,

for $l-1>0$ we call recursively $u_{l-1}^{j} = MG_{l-1}(u_{l-1}^{(j-1)}, f_{l-1}),\ j = 1,\dots,\gamma,$
where $u_{l-1}^{(0)} \in V_{l-1}$ and γ is the number of repetitions of the MG_{l-1} algorithm (specifically $\gamma = 1$ for V-cycle and $\gamma = 2$ for W-cycle),

correction

$$y_l^{(0)} = u_l^{(\kappa_1)} + I_l^{l-1} u_{l-1}^{\gamma}, \tag{6.4}$$

post-smoothing

$$y_l^{(i+1)} = y_l^{(i)} - \mathcal{D}_{l,i+\kappa_1+1}(A_l y_l^{(i)} - f_l),\ i = 0,\dots,\kappa_2 - 1, \tag{6.5}$$

denote this obtained approximation as a result of MG_l algorithm

$$\tilde{u}_l = MG_l(u_l^{(0)}, f_l) = y_l^{(\kappa_2)}.$$

$\mathcal{D}_{l,i},\ i = 1,\dots,\kappa_1+\kappa_2$ represents a linear operator denoting an iterative method. The Multigrid algorithm is iterated until prescribed tolerance is achieved.

6.2 The Operator of Error Suppression

The operator of error suppression describes the evolution of the error in one repetition of multigrid algorithm. The convergence of the algorithm can be proved by studying and properly bounding the operator.

Consider a fixed value of i within the range from 1 to k. Given that u_i represents the exact solution to the problem $A_i u_i = f_i$, we can denote the error of the initial guess as $e_i^{(0)} = u_i - u_i^{(0)}$. Then we can preform the MG_i algorithm for $e_i^{(0)}$ to obtain corresponding error $\tilde{e}_i$ for the final approximation $\tilde{u}_i$. Now denote the operator of error suppression $B_i : e_i^{(0)} \to \tilde{e}_i$ [4, chapter 4.2.2].

Lemma 1. *The operator of error suppression B_i is linear for any $e_i^{(0)}$, independent of f_i, $u_i^{(0)}$ and has the form*

$$B_i = J_i^{(\kappa_2)}(I - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i)J_i^{(\kappa_1)}, \tag{6.6}$$

where

$$J_i^{(\kappa_1)} = (I - \mathcal{D}_{i,\kappa_1}A_i)\dots(I - \mathcal{D}_{i,1}A_i)$$
$$J_i^{(\kappa_2)} = (I - \mathcal{D}_{i,\kappa_1+\kappa_2}A_i)\dots(I - \mathcal{D}_{i,\kappa_1+1}A_i),$$

for I denoting the identity operator.

The matrices $J_i^{(\kappa_1)}$ and $J_i^{(\kappa_2)}$ represent the error smoothing through the chosen smoother (typically Jacobi, weighted Jacobi, or Gauss-Seidel method) accomplished in κ_1 or κ_2 steps of the respective method in (6.2) and (6.5).

Proof. Proof by induction with respect to i.

For i = 0 the problem $A_0u_0 = f_0$ is solved directly giving the exact solution $u_0^{(1)} = MG_0(u_0^{(0)}, f_0)$ with a corresponding error $e_0 = 0$. Therefore, B_0 is a null operator satisfying all the conditions of the lemma.

For i > 0 assume that B_{i-1} is an operator satisfying all the conditions. It is linear for any $e_{i-1}^{(0)}$, independent of f_{i-1}, $u_{i-1}^{(0)}$ and has the form $B_{i-1} = J_{i-1}^{(\kappa_2)}(I - I_{i-1}^{i-2}(I - B_{i-2}^{\gamma})A_{i-2}^{-1}R_{i-2}^{i-1}A_{i-1})J_{i-1}^{(\kappa_1)}$.

In the pre-smoothing phase, denote the error of iterative approximation as $e_i^{(l)} = u_i - u_i^{(l)}$. By using (6.2) and subtracting u_i we get

$$\begin{aligned} u_i^{(l+1)} - u_i &= u_i^{(l)} - u_i - \mathcal{D}_{i,l+1}(-A_ie_i^{(l)} + A_iu_i - f_i) \\ -e_i^{(l+1)} &= -e_i^{(l)} + \mathcal{D}_{i,l+1}A_ie_i^{(l)} \\ e_i^{(l+1)} &= (I - \mathcal{D}_{i,l+1}A_i)e_i^{(l)}. \end{aligned}$$

Then for $l = \kappa_1$ using the induction

$$\begin{aligned} &e_i^{(\kappa_1)} = (I - \mathcal{D}_{i,l+1}A_i)e_i^{(\kappa_1-1)} = \dots = (I - \mathcal{D}_{i,\kappa_1}A_i)\dots(I - \mathcal{D}_{i,1}A_i)e_i^{(0)} \\ &e_i^{(\kappa_1)} = J_i^{(\kappa_1)}e_i^{(0)}. \end{aligned}$$

In the restriction phase, using (6.3),

$$f_{i-1} = R_{i-1}^{i}A_ie_i^{(\kappa_1)}.$$

The problem $A_{i-1}u_{i-1} = f_{i-1}$ has the exact solution u_{i-1}, i. e.,

$$u_{i-1} = A_{i-1}^{-1}R_{i-1}^{i}A_ie_i^{(\kappa_1)}. \tag{6.7}$$

In the coarse-grid solution phase, the initial guess is zero, $u_{i-1}^{(0)} = 0$. Therefore, the initial error is equal to the exact solution $e_{i-1}^{(0)} = u_{i-1}$. In this stage the MG_{i-1}-algorithm is repeated γ times. After one repetition we get the result $\tilde{u}_{i-1}$ with an error $\tilde{e}_{i-1}$, i. e.,

$$\tilde{e}_{i-1} = B_{i-1}e_{i-1}^{(0)} = B_{i-1}u_{i-1}.$$

After γ repetitions, the obtained approximation is u_{i-1}^{γ} and the error equals to

$$\tilde{e}_{i-1}^{\gamma} = B_{i-1}e_{i-1}^{\gamma-1} = \dots = B_{i-1}^{\gamma}e_{i-1}^{(0)} = B_{i-1}^{\gamma}u_{i-1}.$$

Therefore

$$u_{i-1}^{\gamma} = u_{i-1} - \tilde{e}_{i-1}^{\gamma} = u_{i-1} - B_{i-1}^{\gamma}u_{i-1} = (I - B_{i-1}^{\gamma})u_{i-1}. \tag{6.8}$$

In the correction and the post-smoothing phases, we get the approximation $y_i^{(0)}$ with the error $\tilde{e}_i^{(0)} = u_i - y_i^{(0)}$. By subtracting exact solution u_i from both side in equation (6.4)

$$-\tilde{e}_i^{(0)} = -e_i^{(\kappa_1)} + I_i^{i-1}u_{i-1}^{\gamma}.$$

Using (6.7) and (6.8),

$$\begin{aligned}\tilde{e}_i^{(0)} &= e_i^{(\kappa_1)} - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i)e_i^{(\kappa_1)} = \\ &= (I - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i))e_i^{(\kappa_1)}\end{aligned}$$

In the post-smoothing phase, proceed analogously to the first part

$$\tilde{e}_i^{(l+1)} = (I - \mathcal{D}_{i,l+\kappa_1+1}A_i)\tilde{e}_i^{(l)}.$$

Then for $l = \kappa_2$ using the induction

$$\begin{aligned}\tilde{e}_i^{(\kappa_2)} &= (I - \mathcal{D}_{i,\kappa_1+\kappa_2}A_i)\tilde{e}_i^{(\kappa_1+\kappa_2-1)} = \\ &= (I - \mathcal{D}_{i,\kappa_1+\kappa_2}A_i)\dots(I - \mathcal{D}_{i,\kappa_1+1}A_i)\tilde{e}_i^{(0)} = \\ &= J_i^{(\kappa_2)}\tilde{e}_i^{(0)}.\end{aligned}$$

As $\tilde{e}_i = \tilde{e}_i^{(\kappa_2)}$

$$\begin{aligned}\tilde{e}_i &= J_i^{(\kappa_2)}\tilde{e}_i^{(0)} = \\ &= J_i^{(\kappa_2)}(I - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i))e_i^{(\kappa_1)} = \\ &= J_i^{(\kappa_2)}(I - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i))J_i^{(\kappa_1)}e_i^{(0)}.\end{aligned}$$

therefore $B_i : e_i^{(0)} \to \tilde{e}_i$ is in a form $B_i = J_i^{(\kappa_2)}(I - I_i^{i-1}(I - B_{i-1}^{\gamma})A_{i-1}^{-1}R_{i-1}^{i}A_i)J_i^{(\kappa_1)}$. The operator B_i is linear for any $e_i^{(0)}$, independent of f_i and $u_i^{(0)}$. □

Given the form of the error suppression operator, a sufficient condition for convergence of the multigrid method is that the spectral norm $\varrho(B_i)$ is strictly less than one [14, Theorem 1.3.4]. In general, this is hard to verify and this condition is replaced by requiring that a proper norm of B_i is strictly less than one, [14, Theorem 1.3.2].

6.3 Convergence of the Symmetric V-cycle

In this section, we explore the convergence and convergence criteria within a specific case, the symmetric V-cycle. We expect that the symmetric V-cycle with $\gamma = 1$ and $\kappa_1 = \kappa_2$ satisfies the following:

The operators A_i are self-adjoint and positive definite. (6.9)

The interpolation and restriction operators meet a variational property

$$I_i^{i-1} = \frac{1}{c^*} \cdot (R_{i-1}^{i})^*, \qquad c^* \in \mathbb{R} > 0. \tag{6.10}$$

$$J_i^{(\kappa_1)} = J_i^{(\kappa_2)^*} \tag{6.11}$$

in terms of inner products $(\cdot,\cdot)_i$. Finally,

$$A_{i-1} = R_{i-1}^{i}A_iI_i^{i-1}, \tag{6.12}$$

$$J_iA_i = A_iJ_i \tag{6.13}$$

where $J_i = J_i^{(\kappa_1)}$.

Lemma 2. *V-cycle multigrid method from subsection 5.2.1 with Jacobi or weighted Jacobi iteration meets the conditions* (6.9) - (6.13).

Proof. All the matrices A from section 2.4 are real symmetric operators, therefore they are self-adjoint.

$$(Ax, y)_i = (Ax)^T y = x^T A^T y = x^T(Ay) = (x, Ay)_i.$$

Matrices are also irreducibly diagonally dominant ($\forall k : |a_{k,k}| \geq \sum_{j \neq k} |a_{k,j}|$ and $\exists k : |a_{k,k}| > \sum_{j \neq k} |a_{k,j}|$) with real positive diagonal entries. Therefore they are positive definite.

$$\begin{aligned} x^T Ax &= \sum_{k=1}^{n} a_{k,k} x_k^2 + \sum_{k=1}^{n} \sum_{\substack{j=1 \\ j \neq k}}^{n} a_{k,j} x_j x_k \geq \sum_{k=1}^{n} \Big(a_{k,k} x_k^2 - \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{k,j}||x_j||x_k| \Big) = \\ &= \sum_{k=1}^{n} \Big(|a_{k,k}| x_k^2 - \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{k,j}||x_j||x_k| \Big) > \sum_{k=1}^{n} \Big(\sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{k,j}| x_k^2 - \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{k,j}||x_k||x_j| \Big) = \\ &= \sum_{k=1}^{n} \sum_{j>k}^{n} (|a_{k,j}|(x_k^2 + x_j^2 - 2|x_k||x_j|)) = \sum_{k=1}^{n} \sum_{j>k}^{n} |a_{k,j}|(x_k - x_j)^2 \geq 0 \end{aligned}$$

If we set interpolation and restriction operators as $I_i^{i-1} = I_h^{2h}$, $R_{i-1}^i = R_{2h}^h$ as defined in section 4.2, then those operators satisfy variational property (6.10) with $c^* = 2^d$ as shown subsection 4.2.2 as well as (6.12) $A_{i-1} = R_{i-1}^i A_i I_i^{i-1}$, which is defines the coarse grid operator.

To prove $J_i^{(\kappa_1)} = J_i^{(\kappa_2)^*}$ we need to find a form of matrix $\mathcal{D}_{l,i}$ in (6.2) and (6.5).

Weighted Jacobi iteration method is for $\omega \in \mathbb{R}$

$$u_l^{(i+1)} = (I - \omega D_l^{-1} A) u_l^{(i)} + \omega D_l^{-1} f_l = u_l^{(i)} - \omega D_l^{-1}(A_l u_l^{(i)} - f_l).$$

Therefore $\mathcal{D}_{l,i} = \omega D_l^{-1}, i = 0, \ldots, \kappa$, where $\kappa = \kappa_1 = \kappa_2$. Then

$$\begin{aligned} J_i^{(\kappa)} &= (I - \omega D_i^{-1} A_i) \ldots (I - \omega D_i^{-1} A_i) \\ J_i^{(\kappa)^*} &= ((I - \omega D_i^{-1} A_i) \ldots (I - \omega D_i^{-1} A_i))^* \\ &= (I - \omega D_i^{-1} A_i)^* \ldots (I - \omega D_i^{-1} A_i)^* \\ &= (I^* - (\omega D_i^{-1} A_i)^*) \ldots (I^* - (\omega D_i^{-1} A_i)^*) \\ &= (I - \omega A_i^* D_i^{-*}) \ldots (I - \omega A_i^* D_i^{-*}) \\ &= (I - \omega A_i D_i^{-1}) \ldots (I - \omega A_i D_i^{-1}) \end{aligned}$$

Since D_i^{-1} is a diagonal matrix with the same elements along the diagonal (in particular, $\frac{h^2}{2}$ in one dimension, $\frac{h^2}{4}$ in two dimensions and $\frac{h^2}{6}$ in three dimensions), then the commutativity property holds $A_i D_i^{-1} = D_i^{-1} A_i$, as the multiplication by a multiple of identity matrix is analogous to scaling matrix A_i by a scalar. Therefore

$$J_i^{(\kappa)^*} = (I - \omega D_i^{-1} A_i) \ldots (I - \omega D_i^{-1} A_i) = J_i^{(\kappa)}$$

and (6.11) holds (even for Jacobi method with $\omega = 1$).

The last property (6.13), i.e. $J_iA_i = A_iJ_i$, can be proved using the commutativity property used in a previous part as

$$\begin{aligned}
J_iA_i &= (I - \omega D_i^{-1}A_i)\dots(I - \omega D_i^{-1}A_i)A_i \\
&= (I - \omega D_i^{-1}A_i)\dots(A_i - \omega D_i^{-1}A_iA_i) = \\
&= (I - \omega D_i^{-1}A_i)\dots(A_i - \omega D_i^{-1}A_iA_i) \\
&= (I - \omega D_i^{-1}A_i)\dots(A_i - A_i\omega D_i^{-1}A_i) \\
&= (I - \omega D_i^{-1}A_i)\dots(I - \omega D_i^{-1}A_i)A_i(I - \omega D_i^{-1}A_i) \\
&= \dots = A_i(I - \omega D_i^{-1}A_i)\dots(I - \omega D_i^{-1}A_i) \\
&= A_iJ_i.
\end{aligned}$$

Therefore, all the assumptions hold for a symmetric multigrid as defined above. □

If using a Gauss-Seidel iteration, the condition (6.13) is not satisfied. That is the reason we limit our consideration to the Jacobi, or weighted Jacobi method moving forward.

With the inner product $(\cdot,\cdot)_i$ in V_i let us introduce A_i-inner product denoted as $[u,v]_i = (A_iu,v)_i,\ u,v \in V_i$, with its corresponding norm $[\![u]\!]_i = \sqrt{[u,u]_i}$ and a matrix norm

$$[\![M]\!]_i = \sup_{v \in V_i \setminus \{0\}} \frac{[\![Mv]\!]_i}{[\![v]\!]_i}.$$

As mentioned in section 6.2, the multigrid method convergences when the norm of the error suppression operator is less than one. This introduced norm is used in the proof of convergence as it simplifies the proof.

To obtain convergence, however, we will have to add one more condition, namely for the size of smoothing of the solution in the pre/post-smoothing phase.

Clearly, we require the error norm after the smoothing phase to be smaller than the error norm before the smoothing phase ($[\![v]\!]_i^2 - [\![J_iv]\!]_i^2 \geq 0$). That is so that the error does not increase.

Furthermore, we expect the algorithm to satisfy the following. When the vector is effectively smoothed out during the smoothing phase, indicated by a large $[\![v]\!]_i^2 - [\![J_iv]\!]_i^2$, the subsequent correction on a fine grid may not require as much efficiency, $[\![Q_iJ_iv]\!]_i^2$ could be larger. But, if the vector is not smoothed enough during the smoothing phase, $[\![v]\!]_i^2 - [\![J_iv]\!]_i^2$ being close to zero, a coarse grid correction must be efficient, leaving $[\![Q_iJ_iv]\!]_i^2$ minimised.

We can express both of those conditions in one inequality

$$[\![v]\!]_i^2 - [\![J_iv]\!]_i^2 \geq c\,[\![(I - I_i^{i-1}A_{i-1}^{-1}R_{i-1}^iA_i)J_iv]\!]_i^2 \tag{6.14}$$

where $c > 0$. Denote the operator $Q_i : u \to v \in V_i$ as $Q_i := I - I_i^{i-1}A_{i-1}^{-1}R_{i-1}^iA_i$. Then the convergence criterion is

$$[\![v]\!]_i^2 - [\![J_iv]\!]_i^2 \geq c\,[\![Q_iJ_iv]\!]_i^2. \tag{6.15}$$

Theorem 3. *Suppose the conditions* (6.9) *to* (6.13) *are valid for the symmetric MG_l algorithm with $\gamma = 1$ (V-cycle). Also, let the convergence criterion* (6.15) *hold for $c > 0$. Under these conditions, for the A_i-norm of the error suppression applies*

$$[\![B_i]\!]_i \le \frac{1}{1+c}. \tag{6.16}$$

Proof. In finite-dimensional V_i

$$[\![B_i]\!]_i = \sup_{v\in V_i\setminus\{0\}} \frac{[\![B_i v]\!]_i}{[\![v]\!]_i} = \sup_{v\in V_i\setminus\{0\}} \frac{[B_i v, v]_i}{[v,v]_i}.$$

Prove the inequality (6.16) by induction for $\varepsilon > 0$

$$0 \le \frac{[B_i v, v]_i}{[v,v]_i} \le \frac{\varepsilon}{2^d} := \frac{1}{1+c} \qquad \forall v \in V_i \setminus \{0\} \tag{6.17}$$

For $\mathbf{i = 0}$ B_0 is a null operator $[\![B_i]\!]_i = 0$ and (6.17) is valid for any $c > 0$.

For $\mathbf{i > 0}$ suppose that B_{i-1} is satisfies (6.17). Then using Lemma 1 and property (6.11)

$$\begin{aligned} [B_i v, v]_i &= [J_i^*(I - I_i^{i-1}(I - B_{i-1})A_{i-1}^{-1}R_{i-1}^i A_i)J_i v, v]_i = \\ &= [J_i^* Q_i J_i v, v]_i + [J_i^* I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, v]_i = \\ &= [Q_i J_i v, J_i v]_i + \frac{1}{c^*}[B_{i-1}A_{i-1}^{-1}R_{i-1}^i A_i J_i v, A_{i-1}^{-1}R_{i-1}^i A_i J_i v]_{i-1}, \end{aligned}$$

because under assumptions (6.9), (6.10), (6.12) and (6.13)

$$\begin{aligned} &[J_i^* I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, v]_i = (A_i J_i^* I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, v)_i = \\ &\quad = (J_i^* A_i I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, v)_i = (A_i I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, J_i v)_i = \\ &\quad = (I_i^{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, A_i J_i v)_i = \\ &\quad = \frac{1}{2^d}((R_{i-1}^i)^* B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, A_i J_i v)_i = \\ &\quad = \frac{1}{2^d}(B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, R_{i-1}^i A_i J_i v)_i = \\ &\quad = \frac{1}{2^d}(A_{i-1}^{-1} A_{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, R_{i-1}^i A_i J_i v)_{i-1} = \\ &\quad = \frac{1}{2^d}(A_{i-1} B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, A_{i-1}^{-1} R_{i-1}^i A_i J_i v)_{i-1} = \\ &\quad = \frac{1}{2^d}[B_{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, A_{i-1}^{-1} R_{i-1}^i A_i J_i v]_{i-1}. \end{aligned}$$

From the induction assumption

$$\begin{aligned} 0 \le [B_i v, v]_i &\le [Q_i J_i v, J_i v]_i + \frac{\varepsilon}{2^d}[A_{i-1}^{-1} R_{i-1}^i A_i J_i v, A_{i-1}^{-1} R_{i-1}^i A_i J_i v]_{i-1} = \\ &= [Q_i J_i v, J_i v]_i + \frac{\varepsilon}{2^d}(R_{i-1}^i A_i J_i v, A_{i-1}^{-1} R_{i-1}^i A_i J_i v)_{i-1} = \\ &= [Q_i J_i v, J_i v]_i + \frac{\varepsilon}{2^d}(A_i I_i^{i-1} A_{i-1}^{-1} R_{i-1}^i A_i J_i v, J_i v)_i = \\ &= [Q_i J_i v, J_i v]_i - \frac{\varepsilon}{2^d}(A_i(-I + I - I_i^{i-1} A_{i-1}^{-1} R_{i-1}^i A_i)J_i v, J_i v)_i = \\ &= [Q_i J_i v, J_i v]_i - \frac{\varepsilon}{2^d}(A_i(-I + Q_i)J_i v, J_i v)_i = \\ &= (1 - \frac{\varepsilon}{2^d})[Q_i J_i v, J_i v]_i + \frac{\varepsilon}{2^d}[J_i v, J_i v]_i = \\ &= (1 - \frac{\varepsilon}{2^d})[\![Q_i J_i v]\!]_i + \frac{\varepsilon}{2^d}[\![J_i v]\!]_i. \end{aligned}$$

Using convergence criterion (6.15) and $c = \frac{2^d - \varepsilon}{\varepsilon}$

$$0 \le [B_i v, v]_i \le \frac{\varepsilon}{2^d} [\![v]\!]_i.$$

□

Sufficient Conditions for the Convergence

Theorem 3 guarantees the convergence under the convergence criterion (6.15). Now, we state some inequalities that may be more easily verified, directly imply (6.15) and, therefore, convergence.

Lemma 4. *Consider the "approximation" assumption*

$$[Q_i v, v]_i \le c_1 \frac{||A_i v||_i^2}{\lambda_i^{(1)}} \qquad \forall v \in V_i \tag{6.18}$$

where $c_1 > 0$ is a constant and $\lambda_i^{(1)}$ is the largest eigenvalue of A_i and the "smoothing" assumption

$$\frac{||A_i J_i v||_i^2}{\lambda_i^{(1)}} \le c_2 ([\![v]\!]_i^2 - [\![J_i v]\!]_i^2) \qquad \forall v \in V_i \tag{6.19}$$

with $c_2 > 0$.

Then these assumptions with conditions (6.9)*,* (6.10) *and* (6.12) *imply* (6.15) *with constant $c = \frac{1}{c_1 c_2}$.*

Proof. Consider vector $J_i v$. Then using using assumptions (6.18) and (6.19)

$$[\![Q_i J_i v]\!]_i^2 = [Q_i J_i v, J_i v]_i \le c_1 \frac{||A_i J_i v||_i^2}{\lambda_i^{(1)}} \le c_1 c_2 ([\![v]\!]_i^2 - [\![J_i v]\!]_i^2).$$

This gives the convergence criterion with $c = \frac{1}{c_1 c_2}$. □

In contrast to (6.15), the criteria (6.18) and (6.19) separate assumptions on the coarse grid correction (in (6.18)) and the smoother (in (6.19)). Approximation assumption (6.18) is then met by a proper discretisation and construction of the multilevel hierarchy. Its verification for particular finite element discretisation is given, e.g., in [4, Section 4]; [15, Section 6]. The choice of the smoother and number of smoothing iterations is then motivated to satisfy (6.19), respectively to lead to a small value of c_2 in (6.19). Verification of (6.19) for particular relaxation methods (for example Jacobi, weighted Jacobi) and bounds or estimates on c_2 can be found, e.g., in [4, Section 4]; [15, Section 6].

Now, focus on the smoothing criterion. In our case, the smoothing process done by Jacobi or weighted Jacobi method involves κ identical steps defined by a matrix $\mathcal{D}_i = \mathcal{D}_{i,j}, j = 0, \ldots, 2\kappa - 2 = 2\kappa_1 - 2 = 2\kappa_2 - 2$.

Define the iterative operator as $K_i = I - \mathcal{D}_i A_i$. Then $J_i = K_i^\kappa$ and the smoothing property can be expressed in an alternative form

$$\frac{||A_i v||_i^2}{\lambda_i^{(1)}} \le c_3 [(I - K_i) v, v]_i \qquad \forall v \in V_i, c_3 = const. > 0. \tag{6.20}$$

Lemma 5. *Assume that* (6.9) *and* (6.20) *holds, and consider the operator K_i to be non-negative in the A_i-inner product, symmetric and $[\![K_i]\!]_i \leq 1$. Under these conditions,* (6.19) *is valid.*

Proof. Thanks to the conditions, we can use inequality

$$[(I - K_i)K_i^{2\kappa}v, v]_i \leq \frac{1}{2\kappa}\sum_{k=0}^{2\kappa-1}[(I - K_i)K_i^k v, v]_i = \frac{1}{2\kappa}[(I - K_i^{2\kappa})v, v]_i$$

from [16, Inequality (3.16)]. Using $J_i = K_i^\kappa$

$$\begin{aligned}[(I - K_i)J_i^2 v, v]_i &\leq \frac{1}{2\kappa}[(v - J_i^2)v, v]_i \\ [(I - K_i)J_i v, J_i v]_i &\leq \frac{1}{2\kappa}([\![v]\!]_i^2 - [\![J_i v]\!]_i^2).\end{aligned}$$

Choosing vector $J_i v$ for v in assumption (6.20) we get

$$\frac{||A_i J_i v||_i^2}{\lambda_i^{(1)}} \leq c_3[(I - K_i)v, v]_i \leq \frac{c_3}{2\kappa}([\![v]\!]_i^2 - [\![J_i v]\!]_i^2),$$

which is (6.19) with $c_2 = \frac{c_3}{2\kappa}$. □

7 Numerical Experiments

In this chapter, we perform experiments illustrating certain concepts from section 6.3. Specifically, we look into the assumption that was necessary for proving the convergence, symmetry, and into the independence of the operator of error suppression on the right-hand side vector.

As a test problem consider the 3D cube $\Omega = [0,1]^3$ discretised with $n = 128$ points in each dimension (resulting in a matrix of size $2,048,383 \times 2,048,383$) and a constant right-hand side f of ones. We repeat multigrid iterations until,

$$\frac{|f - A \cdot u_{\text{solution}}|}{|f|} < 10^{-6}. \tag{7.1}$$

The relative residual is used as we do not have the access to the error; for the system of this size, direct solve was not feasible on our computational resources. We, therefore, rely on alternative measures like the relative residual to measure the convergence, although a small residual alone does not necessarily imply a small error.

All the experiments were performed in MATLAB, version R2024a, run on a portable computer with 8.00 GB (7.82 GB usable) RAM memory. Run times and numbers of repetitions of the multigrid algorithm were taken as an average from 5 runs and measure only the running times of the multigrid algorithm (in particular, matrix generation is not included in the time). The codes are available in the GitHub repository `https://github.com/AnnaMarieM/Basic_Properties_of_Multigrid_Methods.git`.

Combination of number of smoothing steps κ and number of levels L

In this experiment, we explore the symmetric variant of the V-cycle, where parameter κ represents both the number of smoothing steps in pre-smoothing κ_1 and the number of smoothing steps in post-smoothing κ_2, $\kappa = \kappa_1 = \kappa_2$. We investigate whether different combinations of numbers of smoothing steps κ and numbers of grid levels L have any determinable influence on convergence time. For parameter L values $2, 3, 4, 5, 6$ are used, that correspond to coarse-grid problems of sizes $250047, 29791, 3375, 343, 27$ respectively.

$\kappa \backslash L$	2	3	4	5	6
1	$182.5754, 11$	$6.5268, 13$	3.3169, 14	$3.3688, 15$	$3.4289, 15$
2	$139.8749, 7$	$4.4533, 7$	$2.7091, 8$	2.6154, 8	$2.8677, 9$
3	$106.6443, 5$	$3.8405, 5$	$2.8077, 6$	2.4671, 6	$2.9622, 7$
4	$88.8609, 4$	$7.4665, 5$	2.6450, 5	$2.8604, 6$	$3.3301, 6$
5	$88.5994, 4$	$4.2135, 4$	$2.8815, 5$	2.8400, 5	$3.5388, 6$
6	$89.4711, 4$	$4.2126, 4$	$3.2127, 5$	3.1751, 5	$4.4744, 5$
7	$72.0012, 3$	$4.2755, 4$	2.9624, 4	$3.5668, 5$	$4.3057, 5$

Table 7.1 Time (in seconds) and multigrid repetitions needed for the V-cycle to converge with different combinations of a number of levels and smoothing steps.

From the table, an observation can be made, that, for all values of κ, there is a trend of decreasing convergence time with an increase in the number of levels L up to four or five levels. For $L = 4$ and $L = 5$, the convergence times are usually similar. However, beyond five levels, the time starts to increase. This suggests that for this particular problem size and the data, $L = 4$ or $L = 5$ may represent optimal parameter settings.

The reason behind this optimum could be that for $L = 4$ and $L = 5$, the system becomes sufficiently small on a coarse grid (a $16 \times 16 \times 16$ grid corresponding to SLE of size 3375 for $L = 4$ and a $8 \times 8 \times 8$ grid corresponding to SLE of size 343 for $L = 5$ and), allowing for an efficient direct solution. Additionally, with fewer levels, there are fewer interpolations and restrictions performed, which may be the reason behind the inefficiency of 6 levels.

Another observation is that there is no clearly optimal value for the number of smoothing steps κ. Therefore, we perform another experiment to identify the optimal κ value for $L = 4$ and $L = 5$.

Symmetry and convergence

This section investigates the relationship between symmetry and convergence in a computational experiment when setting the number of levels to $L = 4$ and $L = 5$. Specifically, we explore the impact of varying the number of pre-smoothing and post-smoothing steps, denoted by κ_1 and κ_2 respectively, on the convergence behaviour and running time. The results are given in Table 7.2 for $L = 4$ and in Table 7.3 for $L = 5$.

$\kappa_1 \backslash \kappa_2$	1	2	3	4	5	6	7
1	4.2365, 16	3.0217, 10	2.7837, 8	3.0761, 7	3.1479, 6	3.2267, 6	3.3406, 5
2	3.9072, 11	3.3874, 8	3.0830, 7	3.2019, 6	3.2784, 6	3.3745, 5	3.0574, 5
3	3.5218, 9	2.6565, 7	2.7391, 6	3.0398, 6	3.0600, 5	3.5622, 5	3.8318, 5
4	3.6071, 8	2.8082, 6	3.4077, 6	3.4250, 5	3.4138, 5	3.8014, 5	3.6700, 5
5	3.4900, 7	3.2919, 6	3.3998, 5	2.8609, 5	3.0522, 5	3.4871, 5	2.9777, 4
6	2.9011, 6	2.8658, 5	3.3848, 5	3.4366, 5	4.2014, 5	2.9484, 4	3.3397, 4
7	3.4104, 6	2.9671, 5	3.1263, 5	3.9830, 5	3.6716, 4	3.3317, 4	4.0306, 4

Table 7.2 Time (in seconds) and multigrid repetitions needed for the V-cycle with parameter $L = 4$ to converge with a varying number of pre-smoothing and post-smoothing steps.

Several observations can be made from Table 7.2 and Table 7.3. Convergence times for the V-cycle with $L = 4$ have mean and variance 3.3230 and 0.1424 respectively. For $L = 5$ mean and variance are 3.5046 and 0.2075. Although the

$\kappa_1 \backslash \kappa_2$	1	2	3	4	5	6	7
1	3.7369, 17	3.0715, 11	3.5876, 9	3.6237, 7	3.7819, 7	3.3353, 6	3.5383, 6
2	4.3950, 12	3.5781, 9	2.5292, 7	2.4788, 6	2.7535, 6	3.0307, 6	2.7746, 5
3	3.0721, 10	2.5587, 7	2.8734, 7	3.4416, 6	3.6556, 6	3.3479, 5	3.1389, 5
4	3.4885, 8	3.0954, 7	4.1173, 6	4.0931, 6	3.8055, 5	3.4323, 5	3.8953, 5
5	3.3445, 7	3.7980, 6	3.4883, 6	3.2699, 5	3.9079, 5	4.0216, 5	3.7519, 5
6	4.0769, 7	3.5778, 6	3.0739, 5	3.3902, 5	3.8551, 5	3.6510, 5	3.9741, 5
7	4.0357, 7	3.5615, 6	3.2163, 5	3.5657, 5	3.9710, 5	3.7993, 5	4.1626, 5

Table 7.3 Time (in seconds) and multigrid repetitions needed for the V-cycle with parameter $L = 5$ to converge with a given number of pre-smoothing and post-smoothing steps.

fastest convergence overall is observed for $L = 5$ with parameters $\kappa_1 = 2$ and $\kappa_2 = 4$, convergence times have lower mean and variance for $L = 4$ compared to $L = 5$. This observation indicates potential advantage of the choice $L = 4$, as it generally yields lower convergence times and shows lower sensitivity to the parameters κ_1 and κ_2, as indicated by the variance.

Second, with increasing the number of pre-smoothing steps κ_1 and post-smoothing steps κ_2, convergence time tends to decrease until a certain threshold. Then, on the contrary, the convergence time is increased. This suggests that employing more smoothing steps for pre- and post-smoothing stages does not significantly improve convergence time, although it reduces the number of repetitions of the V-cycle.

Apart from few exceptions, the numbers of multigrid repetition are the same along the antidiagonals. This means that it is the total number $\kappa_1 + \kappa_2$ of smoothing steps that affects the convergence of the mutigrid. Despite a significant effort to identify and correct the issue, the timings do not share this symmetry. For the same number of multigrid repetitons and smoothing steps, the timings should be much closer; the difference seems to be caused by a deficiency of our implementation.

While it might be intuitively expected that symmetrical configurations with equal numbers of pre-smoothing and post-smoothing steps $\kappa_1 = \kappa_2$ would yield faster convergence, the results does not fully support this expectation. Thus, while symmetry was, in our case, used as a condition for convergence, it may not guarantee optimality in terms of convergence time.

Independence of the operator of error suppression and the right-hand side vector

The convergence proof presented in section 6.3 indicated that, theoretically, convergence is independent of the right-hand side. Therefore, in this experiment, we consider solutions of systems with different right-hand side vectors. We set the number of levels $L = 4$ and use $\kappa = 3$ pre- and post-smoothing steps by Weighted Jacobi (3.4) with the weighting parameter $\omega = \frac{4}{5}$.

The right-hand side vectors used in the experiment are following:

- f_1 is a vector filled with ones.
- f_2 is a vector representing the values of a relatively smooth function $f(x, y, z) = \sin(x) \cdot \cos(y) + \sin(y) \cdot \cos(z) + \sin(z) \cdot \cos(x)$ evaluated on a cubic grid.
- f_3 is a vector representing the values of a relatively smooth function $f(x, y, z) = \sin\left(\frac{x}{10}\right) + \sin\left(\frac{y}{10}\right) + \sin\left(\frac{z}{10}\right)$.
- f_4 is a vector representing the values of a function on a cubic grid $f(x, y, z) = 10 \cdot \exp\left(-\frac{(x-0.7)^2+(y-0.3)^2+(z-0.5)^2}{0.0001}\right)$ with a spike at the point $[0.7, 0.3, 0.5]$.
- f_5 and f_6 are vectors representing the values of a function with a spike $f(x, y, z) = \frac{1}{\sqrt{(x-0.7)^2+(y-0.3)^2+(z-0.5)^2+\varepsilon}}$, where $\varepsilon = 10^{-3}$ for f_5 and $\varepsilon = 10^{-5}$ for f_6, i.e. f_5 is “smoother” than f_6.

Right Side	Time (s)	Repetitions	Relative Residual
f_1	3.6316	10	$1.9254 \cdot 10^{-8}$
f_2	3.5328	10	$1.8500 \cdot 10^{-8}$
f_3	3.5556	10	$1.9904 \cdot 10^{-8}$
f_4	3.5411	10	$1.8279 \cdot 10^{-8}$
f_5	3.5428	10	$1.3608 \cdot 10^{-8}$
f_6	3.5333	10	$1.5556 \cdot 10^{-8}$

Table 7.4 Time (s), repetitions of multigrid algorithm, and relative residual for each right-hand side f_i.

Table 7.4 gives the runtime and the number of multigrid repetitions until (7.1) is satisfied for all right-hand side vectors, together with the final value of the relative residual. The results confirm the theoretical findings on the independence of the right-hand side vector to the convergence of multigrid methods. Additionally, figures 7.1 and 7.2 depict the convergence of the relative residual and the relative norm of the error [1] in the repetitions of the multigrid. The results demonstrate asymptotic convergence independent of the right-hand side vector. The method for certain right-hand side vectors yields slightly faster convergence initially, which could be attributed to the more effective elimination of oscillatory error components in the beginning. Later, all demonstrate the same asymptotic convergence.

[1] For the purpose of evaluating the norm of the error, the exact solution is approximated using an excessive number (30) of multigrid repetitions.

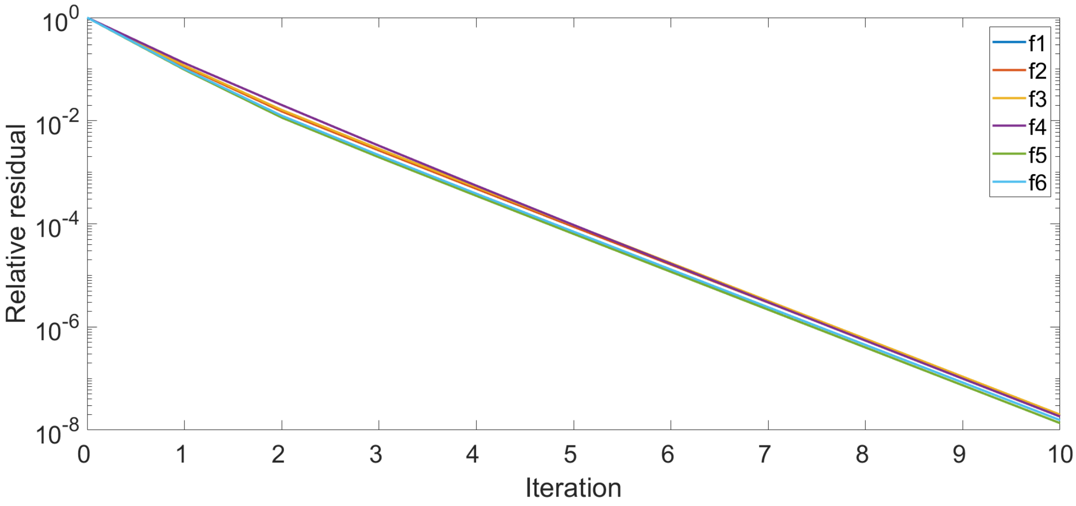


Figure 7.1 Convergence of relative residual norm for different right-hand side vectors.

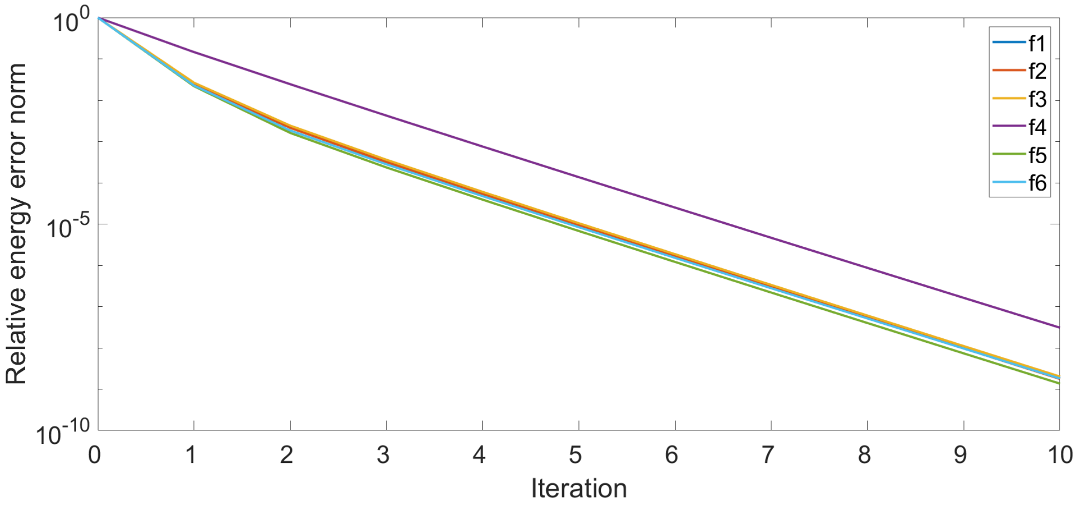


Figure 7.2 Convergence of relative energy error norm $\sqrt{\frac{(u-u^{(i)})^T A(u-u^{(i)})}{(u-u^{(0)})^T A(u-u^{(0)})}}$ for different right-hand side vectors.

Conclusion

Multigrid methods are iterative numerical methods for solving large systems of linear equations stemming from the discretization of a partial differential equation. They are based on two complementary techniques: iterative methods with a so-called smoothing property and coarse-grid correction.

Smoothing property possessed by, e.g., relaxation methods, means that the methods can effectively reduce oscillating parts of the error but are less efficient in reducing smooth error components. Smooth components of the error can be successfully represented on a coarse grid. By solving a problem on a coarse-grid and interpolating the approximation back to the finer grid, the smooth part of the error can be removed. While the size of the coarse-grid system is smaller than the one associated with the finer grid, a recursive call of coarse-grid correction is typically done until the system on the coarsest grid is small enough to be efficiently solved by a direct solver. A careful and elaborated combination of smoothing by a proper relaxation method with a hierarchy of levels for coarse-grid corrections gives a powerful and efficient multigrid solver (or a multigrid preconditioner).

The thesis presents the fundamental principles of multigrid methods and illustrates them on simple examples. Numerical experiments are then performed to study the choice of the parameters in a multigrid method and demonstrate the ability of a multigrid to efficiently solve complex numerical problems. The codes used in the thesis are available in the GitHub repository `https://github.com/AnnaMarieM/Basic_Properties_of_Multigrid_Methods.git`.

Future work could extend the thesis, for example, by exploring discretization by a finite element method or study an application of multigrid for different, potentially non-symmetric problems. Algebraic multigrid methods, which do not explicitly use the hierarchy of the grids, could also be of interest.

Bibliography

1. Evans, Lawrence C. *Partial differential equations*. Vol. 19. Second. American Mathematical Society, Providence, RI, 2010. Graduate Studies in Mathematics. isbn 978-0-8218-4974-3. Available from doi: 10.1090/gsm/019.
2. Briggs, William L.; Henson, Van Emden; McCormick, Steve F. *A multigrid tutorial*. Second. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. isbn 0-89871-462-1. Available from doi: 10.1137/1.9780898719505.
3. Saad, Yousef. *Iterative Methods for Sparse Linear Systems*. Second. Society for Industrial and Applied Mathematics, 2003. Available from doi: 10.1137/1.9780898718003.
4. Shaidurov, V.V. *Multigrid Methods for Finite Elements*. Springer Netherlands, 2013. Mathematics and Its Applications. isbn 9789401585279. Available also from: https://books.google.nl/books?id=DO_qCAAAQBAJ.
5. Selvadurai, A.P.S. Partial Differential Equations in Mechanics 2. *Springer eBooks*. 2000, pp. xviii+698. Available from doi: 10.1007/978-3-662-09205-7.
6. Polyanin, Andrei; Nazaikinskii, Vladimir. *Handbook of Linear Partial Differential Equations for Engineers and Scientists, Second Edition*. 2016. isbn 978-1466581456. Available from doi: 10.1201/b19056.
7. Schinzinger, R.; Laura, P.A.A. *Conformal Mapping: Methods and Applications*. Dover Publications, 2012. Dover Books on Mathematics. isbn 9780486150741. Available also from: https://books.google.cz/books?id=qe-7AQAAQBAJ.
8. Bairstow, Leonard; Berry, Arthur. Two-dimensional solutions of Poisson's and Laplace's equations. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*. 1919, vol. 95, no. 672, pp. 457–475. Available from doi: 10.1098/rspa.1919.0023.
9. Da Silva, Adilson David; Neto, José Dantas; Da Costa, Vladimir Santos. A numerical procedure to solve Poisson's equation in spherical coordinates. *Revista Brasileira de Ensino de Física*. 2021, vol. 43. Available from doi: 10.1590/1806-9126-rbef-2021-0019.
10. Wesseling, Pieter. A robust and efficient multigrid method. In: *Multigrid Methods: Proceedings of the Conference Held at Köln-Porz, November 23–27, 1981*. Springer, 1982, pp. 614–630.
11. McCormick, Stephen F. *Multigrid methods*. SIAM, 1987.
12. Wan, W. L.; Chan, Tony F.; Smith, Barry. An Energy-minimizing Interpolation for Robust Multigrid Methods. *SIAM Journal on Scientific Computing*. 1999, vol. 21, no. 4, pp. 1632–1649. Available from doi: 10.1137/S1064827598334277.
13. Yavneh, I. Why Multigrid Methods Are So Efficient. *Computing in Science & Engineering*. 2006, vol. 8, no. 6, pp. 12–22. Available from doi: 10.1109/MCSE.2006.125.

14. Greenbaum, A. *Iterative Methods for Solving Linear Systems.* Society for Industrial and Applied Mathematics, 1997. Frontiers in Applied Mathematics. isbn 9780898713961. Available also from: `https://books.google.cz/books?id=WwMDNLxrwocC`.

15. Hackbusch, Wolfgang; Trottenberg, Ulrich. *Multigrid Methods. Proceedings of the Conference Held at Köln-Porz, November 23-27, 1981.* 1982. isbn 3-540-11955-8.

16. Bramble, James H.; Pasciak, Joseph E. New convergence estimates for multigrid algorithms. *Mathematics of Computation.* 1987, vol. 49, pp. 311–329. Available also from: `https://api.semanticscholar.org/CorpusID:35764052`.

List of Figures

List of Tables