UNIVERSITY OF TWENTE.

faculty of engineering technology

Engineering Fluid Dynamics Group

Multi-level techniques for convection dominated flows

D. Krijgsman

January 2010

TS-088

Multi-level techniques for convection dominated flows

Dinant Krijgsman

January 7, 2010

Faculty of Engineering Technology
Engineering Fluid Dynamics Group
University of Twente, Enschede, the Netherlands

Summary

In the past decades computational fluid dynamics (CFD) has become a crucial analysis tool in engineering fluid mechanics. Detailed information about the flow and relevant flow parameters can be obtained, thereby reducing the need for often expensive and time consuming experiments. CFD computations involve solving a set of partial differential equations (i.e. the Euler or Navier-Stokes equations) on grids which can be very dense (up to 100 millions cells). To represent the governing equation on the grid they have to be discretized. The resulting problem to be solved using a computer is usually solving a large, sparse system of (non-linear) algebraic equations. Standard algorithms (i.e. Gauss-Seidel or Jacobi iteration) for solving these systems iteratively are computationally very expensive, due to their slow convergence, especially when meshes with many grid points are used.

Multi-level techniques have the prospect of greatly improving the performance of iterative techniques. These methods make use of coarser grids to accelerate convergence on the target grid, thereby reducing the calculation time required to get a converged result within the desired accuracy. For these methods it is required that error components, that are slow to converge on the target grid, can accurately be represented and reduced on a coarser grid and transferred between these grids. Multi-level methods have shown huge reductions in the required amount of computational time for many problems.

Convection dominated flow problems, amongst others, cannot be solved efficiently, when standard multi-grid techniques are used. For this type of problems, some error components cannot be removed by simple relaxations, neither can they be accurately represented on a coarser grid. In this research two approaches are investigated to restore multi-grid efficiency for convection dominated flow problems. In the first approach, using the standard geometrical setting, the coarse grid operator is changed. Both improved and Galerkin based coarse grid operators are able to accurately represent the problematic components on the coarse grid and thereby achieving multi-grid efficiency. In the second approach algebraic multi-grid techniques (AMG) are used. These techniques no longer use standard coarse grids, but rather operator-dependent grids. The coarse grids are chosen such that all components that are slow to converge on the target grid can be accurately represented on the coarser grid. Both approaches are incorporated in a multi-level algorithm for solving the convection-diffusion equation. Detailed results for a scalar model convection-diffusion problem are presented, showing a major increase in performance in comparison with standard iterative and multi-level techniques. The next step is to extend the algorithm to systems of equations.

Preface

This thesis is the final part of my masters program in Mechanical Engineering, followed within the group of Engineering Fluid Dynamics, at the University of Twente. It all started about six years ago when it became necessary to choose a study. At that time I was already quite interested in mathematics, but not just all math problems, only the ones for which the result, after a long calculation, has a direct practical meaning. Therefore, I decided to follow the bachelor program in Mechanical Engineering. The study involves quite a lot of math, but the goal is never the math itself, but usually something like the required thickness of a beam, or the maximum load a crane is able to lift. During my Bachelor this interest in mathematics only grew, but the need for practical answers reduced somewhat, therefore the choice of master became more difficult. For a long time I was unsure about switching completely to the department of mathematics or choosing for a more calculus oriented department within Mechanical Engineering. In the end I decided the latter and joined the group of Engineering Fluid Dynamics of Prof. dr. ir. Hoeijmakers.

This continued shift of interest towards mathematics is also reflected in this thesis about the development of efficient multi-level techniques for numerically solving (systems of) partial differential equations, where not so much the solution itself is of interest, but more the fashion in which the solution is obtained. These methods first got my attention during the lectures of Kees Venner on the subject of Computational Fluid Dynamics, for which one of the final assignment was to build a multi-level solver for the Poisson equation. Later, when I was looking for an assignment for my master thesis, I was thrilled to hear that he was looking for a student to combine his knowledge of multi-level methods with Edwin van der Weide's knowledge of different discretization schemes for partial differential equations. Therefore I would like to thank both of them for supervising my research.

At this point it is common to list others who have helped, but I know almost certainly that I will forget someone special and regret it later on. So I would like to make a more general acknowledgment to everyone who helped doing assignments for my bachelor or master study, supported me in difficult times, accompanied my over a glass of beer or just helped by making my study here in Enschede a wonderful experience.

Dinant, Enschede, January 7

Contents

Nomenclature

Roman symbols

Symbol	Description
A	Matrix of the operators
$A\left(\vec{\theta}\right)$	Complex amplitude of the fourier mode
a	Dimensionless convective speed in x-direction
b	Dimensionless convective speed in y-direction
C	Set of coarse grid points
C_i^s	Set of strongly connected neighboring coarse grid points ($C_i^s = C \cup S_i$)
c	Flow velocity
D	Diagonal part of A
d	Dimension of the problem
e^h	Error with respect to the analytical solution
e_{app}^h	Error with respect to the solution on a coarse grid
F	Set of fine grid points
F_i^s	Set of strongly connected neighboring fine grid points ($F_i^s = F \cup S_i$)
f^h	Right hand side
H	Grid spacing of the coarse grid
h	Grid spacing of the fine grid
$I_h^H\ \langle\rangle$	Interpolation operator
$I_H^h\ \langle\rangle$	Restriction operator
i	Imaginairy number $\left(i^2 = -1\right)$
$K_h^H\ \langle\rangle$	Two-grid correction operator
$L^h\ \langle\rangle$	Operator, describing the discretization of the partial differential equation
M_h^H	Two-grid iteration operator
M_s	Required number of cycles
m	Order of the opperator
m_i	Order of interpolation
m_j	Order of restriction
N	Total number of cells ($N = n^d$)
N_i	Set of neighboring points $\{j \in \Omega : j \neq i, A_{i,j} \neq 0\}$
n	Number of cells in one direction
Pe	Peclet number
Re	Reynolds number
R_i	Limmitter function
r^h	Residual
S^h	Relaxation operator
S_i	Set of strongly connected neighboring points (see equation 3.5)
S_i^T	Set of points strongly influenced by i (see equation 3.6)
s_i	i-th Row sum of A
T^{all}	Set of all high fourier frequencies
T^{high}	Set of all fourier frequencies

TV	Total variation
U	Set of yet undecided grid points
u^*	Analytical solution
u^h	(current) solution
v^h	Error with respect to the exact discrete solution
W_i	Set of weakly connected neighboring points ($W_i = N_i \setminus S_i$)
W_{cycle}	Required computational work for one cycle
WU	Requried work for one relaxation sweep on the finest level

Greek symbols

Symbol	Description
α	Flow angle
$\delta_{i,j}$	Correction for the relaxation of point (i,j)
ϵ	For Poisson problems: diffusion parameter For convection-diffusion problems: Ratio of diffusion over convection [m]
ϵ_{str}	Factor for determining strong influence
ϵ_{tr}	Factor for truncation of interpolation
ϵ_*	Non dimensional diffusion parameter ($\epsilon_* = \epsilon/hc$)
η	Coordinate perpendicular to the flow
γ	Cycle parameter (1 for V-cycles, 2 for W-cycles)
κ	Parameter for κ-scheme discretizations
λ_i	Measure of importante of point i to become the next coarse grid point
μ	Assymptotic convergence speed
μ_{loc}	Smoothing factor
ν_1	Number of pre-relaxations
ν_2	Number of post-relaxation
Ω	The domain
$\partial\Omega$	Boundary of the domain
ω	Relaxation parameter
ϕ	Gridfunction
ρ	Two-grid convergence factor
$\vec{\phi}$	Gridfunctionvector
τ^h	Truncation error
τ_h^H	Relative truncation error
$\theta_{1,2}$	Frequencies in x- and y-direction, respectively
ξ	Coordinate in streamwise direction

Accents

Accent	Description
$\hat{x}$	Next approximation
$\tilde{x}$	Current approximation
$\bar{x}$	Either current or next approximation, based on the relaxation sheme used

Grid indicators

Symbol	Description
x^h	Variable on the fine grid
x^H	Variable on the coarse grid
x^{2H}	Variable on an even coarser grid

Chapter 1

Introduction

In the past decades computational fluid dynamics (CFD) has become a crucial analysis tool in engineering fluid mechanics. Detailed information about the flow and relevant flow parameters can be obtained, thereby reducing the need for often expensive and time consuming experiments. CFD computations involve solving a set of partial differential equations (i.e. the Euler or Navier-Stokes equations) on grids which can be very dense (up to 100 millions of cells). To represent the governing equations on the grid they have to be discretized. The resulting problem to be solved using a computer is usually solving a large, sparse system of (non-linear) algebraic equations. Standard algorithms (i.e. Gauß-Seidel or Jacobi iteration) for solving these systems iteratively are computationally very expensive due to their slow convergence, especially when meshes with many grid points are used. Multi-level techniques have the prospect of greatly increasing the performance of iterative techniques.

1.1 Multi-grid

Multi-Level or Multi-Grid techniques have been developed for efficiently solving large systems of algebraic equations, for example resulting from the discretization of systems of partial differential equations in scientific computing. These methods make use of coarser grids to accelerate convergence on a target grid, thereby reducing the calculation time needed to get a converged result. In this section, the basic elements of a multi-grid algorithm for the fast solution of a discretized partial differential equation are described.

Consider the discretized problem described by:

$$L^h \left\langle u^h \right\rangle = f^h \tag{1.1}$$

with $L^h \langle\rangle$ the operator (a discrete approximation of a differential equation), u^h the unknown solution vector and f^h the known right hand side. Assume some iterative method (e.g. Gauß-Seidel or Jacobi-relaxation) is used to solve the system of equations. These methods are usually perfectly capable of reducing high-frequency error components, but low-frequency error components are reduced much slower (see section 2.1.3). Therefore these relaxations are referred to as smoothers, because they make the error smooth by reducing high-frequency components. After a small number of these relaxation sweeps (called pre-relaxations) on the target grid, an approximation $\tilde{u}^h$ to the exact solution u^h is obtained. Now the (smooth) error (v^h) and residual ($\tilde{r}^h$) are defined as:

$$v^h = u^h - \tilde{u}^h \tag{1.2}$$

$$\tilde{r}^h = f^h - L^h \left\langle \tilde{u}^h \right\rangle \tag{1.3}$$

Combining equation (1.1) and (1.3) gives:

$$\tilde{r}^h = L^h \left\langle u^h \right\rangle - L^h \left\langle \tilde{u}^h \right\rangle \tag{1.4}$$

For a system of linear algebraic equations, this can be rewritten into:

$$\tilde{r}^h = L^h \left\langle u^h - \tilde{u}^h \right\rangle = L^h \left\langle v^h \right\rangle \tag{1.5}$$

If this equation can be solved exactly for v^h, the previous solution ($\tilde{u}^h$) can be updated using the definition of the error (equation 1.2). However, it is equally difficult to solve the new equation as it is to solve equation (1.1). Therefore equation (1.5) is solved approximately using a reduced set of equations. This set is obtained using a 'transfer' to a coarser grid. This transfer is allowed, because the error is smooth on the fine grid (due to the initial relaxations) and can accurately be represented on a coarser grid. The restricted set of equations to be solved is:

$$\begin{aligned} L^H \left\langle \tilde{v}^H \right\rangle &= I_h^H \left\langle f^h - L^h \left\langle \tilde{u}^h \right\rangle \right\rangle \\ &= I_h^H \left\langle \tilde{r}^h \right\rangle \end{aligned} \tag{1.6}$$

Here $I_h^H \langle\rangle$ is an operator to restrict the fine grid residual to the coarser grid and $L^H \langle\rangle$ a coarse grid approximation of the fine grid operator. After this reduced set of equations is solved, the new approximation to the original set of equations can be calculated using:

$$\hat{u}^h = \tilde{u}^h + I_H^h \left\langle \hat{v}^H \right\rangle \tag{1.7}$$

with $I_H^h \langle\rangle$ an operator to interpolate the coarse grid correction to the fine grid. This method of using the residual as the right-hand side for a coarse grid correction is called the Correction Scheme. From this description, one immediately obtains the two-grid correction operator (K_h^H), which give the corrections for the fine grid solution using the coarse grid:

$$K_h^H = I^h - I_H^h \left(L^H\right)^{-1} I_h^H L^h \tag{1.8}$$

When also ν_1 pre- and ν_2 post-relaxations are included, the two-grid iteration operator is obtained:

$$M_h^H = \left(S^h\right)^{\nu_2} K_h^H \left(S^h\right)^{\nu_1} \tag{1.9}$$

with $S^h \langle\rangle$ the relaxation operator. When the coarse grid still contains many points, it can still be time consuming to solve the new restricted set of equations. Therefore the approach can be used recursively, yielding a Multi-Level cycle. As an example a graphical representation of the data flow in a three level cycle is shown in figure 1.1. First pre-relaxation are applied, then the residuals are transferred to a coarser grid. When a better coarse grid solution is obtain they are interpolated to the fine grid and used to correct the fine grid solution.

When performing Multi-Level cycles it can be advantageous to perform additional cycles on coarser grids. Therefore the cycle parameter γ is introduced. This parameter determines how accurately each coarse grid problem is solved before returning to the fine grid. $\gamma = 1$ cycles are usually referred to as V-cycles, whereas $\gamma = 2$ cycles are referred to as W-cycles. A flow diagram of V(ν_1,ν_2) and W(ν_1,ν_2) cycles for a case of four grids is shown in figure 1.2.

When the initial problem is defined, the following individual components still needs to be specified.

- γ cycle parameter
- Ω_H the coarse grid
- $I_h^H \langle\rangle$ the fine-to-coarse restriction operator
- $I_H^h \langle\rangle$ the coarse-to-fine interpolation operator
- $L^H \langle\rangle$ the coarse grid operator
- $S^h \langle\rangle$ the relaxation operator including the number of pre- and post-relaxations

The choice of these components may have a strong influence on the efficiency of the resulting multi-grid solver. There exist some standard components which usually give good results, but also a number of further analysis tools can be used to estimate the convergence properties of the algorithm and to help choose the proper multi-grid components.

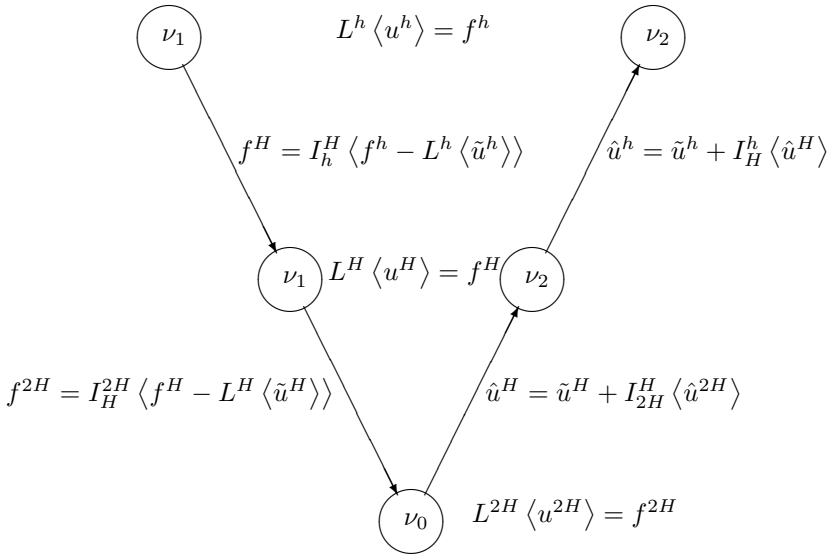
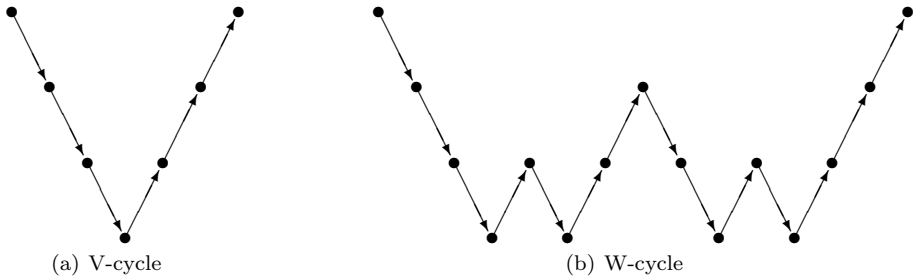


Figure 1.1: Flow diagram for a three level Multi-Level Cycle using the Correction Scheme



(a) V-cycle (b) W-cycle

Figure 1.2: Flow diagrams for the case of 4 grids

1.2 Objective

For some elliptic problems, like the ones governed by Laplaces equation, multi-grid implementation is straightforward and optimal efficiency is relatively easy to obtain. However, for systems of equations resulting from the discretization of the equations governing fluid flows, i.e. the Navier Stokes or Euler equations, the standard methodology is often not very efficient, especially for convection-dominated flows. Several problems need to be tackled before textbook multi-grid efficiency can be achieved for these equations: [12]

- The flow equations are a system of nonlinear PDE's
- Boundary layers may require the use of extremely fine and highly anisotropic grids near zero-slip boundaries of the computational domain.
- For high-Reynolds-number flows, dominance of convection results in a nearly singular pertubated problem
- Second-order accuracy is usually needed to obtain sufficiently accurate approximations.

In this thesis research is focused on the problem of dominating convection. For high Reynolds number flows ($Re \rightarrow \infty$), the flow equations are no longer elliptic like, but become hyperbolic like. A model problem for this type of problem is the steady convection-diffusion equation in two dimensions [12]:

$$-\epsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial au}{\partial x} + \frac{\partial bu}{\partial y} = f\left(x, y\right) \tag{1.10}$$

This equation is also singular perturbated, because for $\epsilon = 0$ it is no longer elliptic, but hyperbolic. The combination of advection and diffusion is a model problem for the momentum equations. The objective of this research is to produce an efficient and robust solver for the convection-diffusion equation as a step to ultimately solving the full Navier Stokes equation for high-Reynolds-number flows.

The main problem in solving the convection-diffusion equation efficiently is the coarse grid operator. For the convection-diffusion equation, some error components cannot be solved by simple relaxations, neither can they be accurately represented on a coarser grid by standard coarse grid operators. In this research two approaches are investigated to achieve textbook multi-grid efficiency. In the first approach, using the standard geometrical setting, the coarse grid operator is modified. Both improved and Galerkin based coarse grid operators are able to accurately represent the problematic error components on the coarse grid and thereby resulting in textbook multi-grid efficiency. In the second approach algebraic multi-grid techniques (AMG) are employed. These techniques no longer use standard coarse grids, but rather operator-dependent grids. The coarse grids are chosen such that all error components that are slow to converge on the target grid can accurately be represented on the coarser grid.

Steps taken to provide methods with good multi-grid performance for the convection-diffusion equation are:

- Develop a 2D multi-grid Poisson problem solver, in order to become more familiar with the multi-level approach. Show and predict the performance of the algorithm for different test cases.
- Illustrate problems associated with dominating convection when standard multi-grid components are used.
- Develop a 2D multi-grid convection-diffusion equation solver with possibilities for improved and Galerkin based coarse grid operators, so that good performance is achieved for problems associated with dominant convection. Show and predict performance for different test cases.
- Become familiar with AMG by applying some (standard) test cases. Show performance and coarsening for different test cases.

- Further develop an AMG method, currently tested for elliptic problems, such that it is also capable of efficiently solving convection dominated problems. Show performance and the coarsening for different cases.

1.3 Thesis outline

In chapter 2 the basics of geometric multi-grid will be discussed. Several components of the multi-grid cycle are explained and some tools to predict the convergence of the algorithm are given. Also extensions to full multi-grid (FMG) and the full approximation scheme (FAS), for non linear problems, are given.

For algebraic multi-grid (AMG) the choice of coarse grids and interpolations is based on the problem itself. In order to split the grid variables into coarse and fine grid variables, the concepts of algebraic smoothness and strong influence and dependence are crucial. These concepts are discussed in chapter 3, along with the algorithms for determining the splitting, interpolation and restriction operators.

In chapter 4 geometric multi-grid techniques will be applied to the Poisson problem. Conventional Poisson problems can be solved relatively easily, but the anisotropic-Poisson problem already requires a renewed analysis. Converge rates for both problems are estimated and compared to the rates of convergence actually achieved.

In chapter 5 geometric multi-grid techniques will be applied to the convection-diffusion problem. Both the improved and Galerkin based coarse grid operators are discussed and applied to a convection dominated problem. Convergence rates are estimated and compared to the rates of convergence actually achieved.

AMG methods applied to the Poisson problem are discussed in chapter 6. Different types of coarsening are obtained for different discretizations of the Poisson problem. Anisotropic problems automatically are handled well by coarsening in only one direction. Convergence rates for several problems are presented.

In chapter 7 AMG methods will be applied to convection dominated problems. The coarsening process is shown for different flow patterns and convergence rates for several problems are presented.

Finally in chapter 8 concluding remarks will be presented for both the geometric and algebraic multi-grid approach. Some recommendations will be given for aspects that need to be investigated further in order to produce an efficient solver for the scalar convection-diffusion equation and later for the Navier-Stokes or Euler equations.

Chapter 2

Geometric multi-grid

In this chapter different aspects of geometric multi-grid algorithms are described. Geometric multi-grid has originally been developed for solving elliptic partial differential equations. For this kind of equations smooth error components generally show slow convergence on the target grid. Therefore these components need to be accurately represented and solved on the coarse grid. Standard coarsening allows for an accurate representation of these smooth functions. Standard coarsening doubles the mesh-size h in each direction, at each coarser level and thus reduces the number of grid points by a factor four for two dimensional problems. Methods like this, in which a fixed coarse grid is used, are referred to as geometric multi-grid approaches, as opposed to algebraic multi-grid approaches, in which the fine grid operator is used to determine the coarse grid points and the interpolation and restriction operators (see chapter 3).

2.1 Geometric multi-grid components

In this section, some examples of the different multi-grid components are introduced. These examples are certainly not the only choices, but are the ones used most frequently and which in many cases lead to an efficient algorithm, i.e. rapid convergence and solution to the level of discretization accuracy in $\mathcal{O}(n)$ operations, with n the number of grid points.

2.1.1 Coarse grid choices and inter-grid operators

The order of interpolation is equal to $k+1$ if an interpolation is exact for all polynomials of degree k. In general a restriction operator can be defined as the transpose of an interpolation operator. In that case the order of a restriction operator is equal to the order of the interpolation operator. If m is the discretization order of the operator L^h and m_i and m_j are the order of interpolation and restriction, respectively, then the orders of the transfer operators should (at least) fulfill: [12]

$$m_i + m_j \geq m \tag{2.1}$$

Interpolation and restriction operators are closely related to the choice of the coarse grid. In this section only transfer operators for standard coarsening are considered, for which all coarse grid points coincide with fine grid points. This kind of grid usually results from a cell-vertex centered discretization.

Interpolation

The purpose of interpolation is to obtain a fine grid representation u^h of a given coarse grid variable u^H:

$$u^h = I_H^h \left\langle u^H \right\rangle \tag{2.2}$$

For interpolation of the corrections in the correction scheme a simple linear interpolation scheme is usually sufficient, because the error is supposed to be smooth. For the !D case this interpolation on a uniform grid is shown in figure 2.1. It can be represented using the following stencil:

$$I_H^h = \frac{1}{2}\rceil \quad 1 \quad 2 \quad 1 \quad \lfloor \tag{2.3}$$

In this notation, the stencil entries correspond to weights in a distribution process, therefore the brackets are reversed.

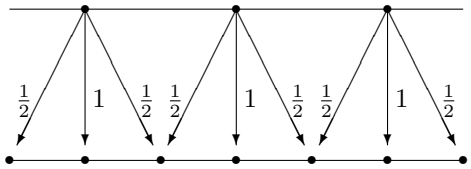


Figure 2.1: Linear interpolation

This means that a quantity at a coarse grid point fully contributes to a coinciding point on the fine grid and contributes half of its value to the direct neighbors. On the boundary the operator will be similar except that there is only one direct neighbor on the fine grid. The error associated with this linear interpolation is $\mathcal{O}\left(h^2\right)$ and it is referred to as second-order. Interpolation in more dimensions can usually be carried out one dimension at the time, using the 1D operators. The resulting stencil is the dyadic product of the 1D operators. For two dimension this becomes:

$$I_H^h = \frac{1}{4}\rceil \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \lfloor \tag{2.4}$$

For the interpolation of the coarse grid error in the coarse grid correction cycle, the use of a second-order interpolation operator is usually sufficient, because of smoothness of the components. However, for the interpolation of a coarse grid solution to the finer grid to serve as a first approximation, see section 2.4, a second-order scheme is insufficient. Higher-order schemes can be derived using e.g. Lagrange interpolation formula. For example cubic interpolation is shown in figure 2.2, which can be represented using the following stencil:

$$I_H^h = \frac{1}{16}\rceil \quad -1 \quad 0 \quad 9 \quad 16 \quad 9 \quad 0 \quad -1 \quad \lfloor \tag{2.5}$$

For points next to the boundary different stencils are required. These stencils can be obtained

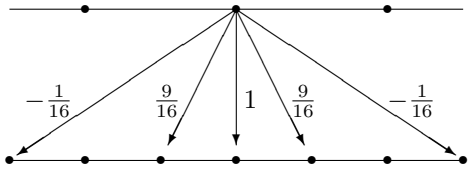


Figure 2.2: Cubic interpolation

using Lagranges interpolation formula. Interpolation in more dimensions can most efficiently be carried out by applying the 1D operators consecutively.

Restriction

The purpose of restriction is to obtain a coarse grid representation r^H of a given fine grid variable r^h:

$$r^H = I_h^H \left\langle r^h \right\rangle \tag{2.6}$$

A simple way of restricting is by injecting the fine grid quantity into the coarse grid. This type of restricting is called injection and can be described with the following stencil for 1D cases (see figure 2.3):

$$I_h^H = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{2.7}$$

A more general case of defining the restriction follows directly from the definition of interpolation:

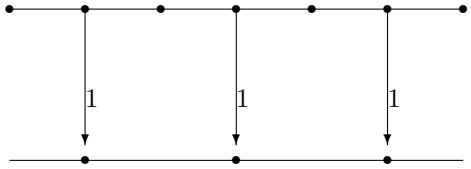


Figure 2.3: Injection

$$I_h^H = \left(\frac{h}{H}\right)^d \left(I_h^H\right)^T \tag{2.8}$$

For example the operator for the 1D case resulting from linear interpolation becomes (see figure 2.4):

$$I_h^H = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{2.9}$$

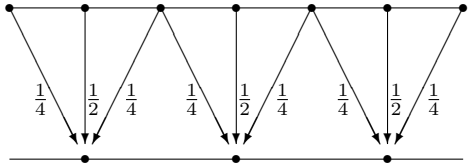


Figure 2.4: Full weighting

This restriction operator is referred to as full weighting, because it weights the values (of the coinciding fine grid point and its two neighbors) on the fine grid to obtain a coarse grid value.

Injection and full weighting differ in the accuracy in which they represent specific aspects of the fine grid vector in the coarse grid result. For example, injection does not preserve the integral over the function, whereas full weighting does. Restriction in two or more dimension is, just as for interpolation, the dyadic product of the 1D operators. For 2D this becomes:

$$I_h^H = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.10}$$

In practice restriction can be carried out by applying restricting one dimension at the time. A more efficient method for restriction, is by scanning all fine grid points and distributing the fine grid value, multiplied with the interpolation weight, to all coarse grid points involved in the interpolation to this fine grid point.

2.1.2 Coarse grid operator

An important aspect of the multi-grid cycle is the coarse grid operator. The coarse grid serves to approximate and solve the low-frequency components that were slow to converge on the fine grid. Therefore, the coarse grid operator has to be a good approximation to the fine grid operator for those low-frequency components. A natural choice is to use the direct analog of L^h on the grid

Ω^H. However, for some problems (i.e. problems with characteristic directions) this approach is insufficient. In such cases the coarse grid operator can be formed by a restriction of equation (1.5):

$$I_h^H \left(L^h v^h \right) = I_h^H \left(\tilde{r}^h \right) \tag{2.11}$$

Substitution of $v^h = I_H^h v^H$ yields:

$$I_h^H \left(L^h I_H^h v^H \right) = I_h^H \left(\tilde{r}^h \right) \tag{2.12}$$

Rearranging the brackets yields:

$$L^H = I_h^H L^h I_H^h \tag{2.13}$$

This coarse grid operator is referred to as the Galerkin coarse grid operator. Interpolation and restriction operators required for the calculation of the Galerkin coarse grid operator are not necessarily the same as used for interpolation and restriction of the residuals, nor are they necessarily the transpose of each other.

2.1.3 Relaxation

Relaxation processes are defined as computing a new approximation to the exact solution given a current approximation. This process is repeated until a desired accuracy is obtained. The corrections for single point schemes can usually be written as:

$$\hat{u}_{i,j}^h = \tilde{u}_{i,j}^h + \omega \delta_{i,j}^h \tag{2.14}$$

with $\tilde{u}_{i,j}^h$ the current approximation, $\hat{u}_{i,j}^h$ the new approximation, ω the relaxation parameter and $\delta_{i,j}^h$ the correction. For simple one-point smoothers, this correction is obtained by dividing the residual by the weight of the central point in the discretization.

$$\delta_{i,j}^h = \left(\left. \frac{\partial L^h \langle \rangle}{\partial u^h} \right|_{u_{i,j}^h} \right)^{-1} \bar{r}_{i,j}^h \tag{2.15}$$

where the residual is calculated using:

$$\bar{r}_{i,j}^h = f_{i,j}^h - L^h \left\langle \bar{u}^h \right\rangle_{i,j} \tag{2.16}$$

The relaxation factor ω can be used for under- or over-relaxation. For linear equations with $\omega = 1$ the new approximation to $u_{i,j}^h$ is such that the discrete equation at point (i, j) is satisfied. For $\omega < 1$ it is referred to as damped or under relaxation.

Depending on the definition of $\bar{r}_{i,j}^h$ in equation (2.15) one can distinguish simultaneous and successive displacement schemes. In the first case $\bar{r}_{i,j}^h$ is computed using 'old' values only. These are simultaneous displacement schemes and are generally referred to as Jacobi or Picard iteration. Alternatively successive displacement can be used, in which already updated values at points previously relaxed can be used to calculate new corrections. This is referred to as Gauss-Seidel relaxation. For this method the order in which points are relaxed (i.e. lexicographic or checkerboard ordering) can be important for the performance. The choice of relaxation technique for a particular problem depends on data capacity and convergence speed. Simultaneous displacement schemes generally require more memory storage space than successive displacement schemes, as both the new and the old approximation need to be stored. However, simultaneous displacement schemes can be fully parallelized, whereas this is not always the case with successive displacement schemes.

2.2 Analysis tools

Several tools exist to predict the performance of multi-level algorithms. Local mode analysis, two-grid analysis, simplified two-grid analysis and first difference approximation are explained in more detail below.

2.2.1 Local Mode Analysis

The performance of a relaxation scheme can be measured in terms of the reduction of the error after each relaxation sweep. Let the error of a given approximation be:

$$\tilde{v}_{i,j}^h = u_{i,j}^h - \tilde{u}_{i,j}^h \tag{2.17}$$

A relation between the error before and after relaxation is obtained by combining equation (2.17), (2.14) and (2.15):

$$\hat{v}_{i,j}^h = \tilde{v}_{i,j}^h - \omega \left(\left. \frac{\partial L^h \langle\rangle}{\partial u^h} \right|_{u_{i,j}^h} \right)^{-1} \left(f_{i,j}^h - L^h \left\langle u^h - \bar{v}^h \right\rangle_{i,j} \right) \tag{2.18}$$

If the operator $L^h \langle\rangle$ is linear, as is the case for the Poisson equation, the solution and the error can be separated so that $f_{i,j}^h$ and $u_{i,j}^h$ can be eliminated. The resulting equation is:

$$\hat{v}_{i,j}^h = \tilde{v}_{i,j}^h - \omega \left(\left. \frac{\partial L^h \langle\rangle}{\partial u^h} \right|_{u_{i,j}^h} \right)^{-1} L^h \left\langle \bar{v}^h \right\rangle_{i,j} \tag{2.19}$$

Different approaches exist to analyze the convergence behavior of iterative methods. One of them is the so called Local Mode Analysis (LMA). This method disregards the effect of the boundary conditions, which for elliptic problems is justified as the influence of the boundary only extends a limited distance into the domain. It is assumed that the error can be written as a sum of Fourier components:

$$\begin{aligned} \tilde{v}_{i,j}^h &= \sum_{\theta_1} \sum_{\theta_2} \tilde{A} \left(\vec{\theta} \right) e^{\theta_1 i + \theta_2 i} \\ \hat{v}_{i,j}^h &= \sum_{\theta_1} \sum_{\theta_2} \hat{A} \left(\vec{\theta} \right) e^{\theta_1 i + \theta_2 i} \\ \vec{\theta} &= (\theta_1, \theta_2)^T \end{aligned} \tag{2.20}$$

Substituting the Fourier components for the errors (equation 2.20 and 2.20) into this equation and assuming that relaxation only maps Fourier components onto itself (i.e. when they are the eigenvectors of the relaxation matrix), makes it possible to determine the error amplification factor (the new error divided by the current error for each Fourier component):

$$\mu \left(\vec{\theta} \right) = \left| \frac{\hat{A} \left(\vec{\theta} \right)}{\tilde{A} \left(\vec{\theta} \right)} \right| \tag{2.21}$$

Convergence of an iterative method requires $\mu < 1$ for all Fourier frequency components $\vec{\theta} \in T^{all}$ where $T^{all} = [-\pi, \pi]^2 \setminus (-h, h)^2$ (i.e. all frequency components except for $|\theta_1| < h$ and $|\theta_2| < h$), with h the mesh-size in the corresponding direction. The asymptotic convergence speed of a basic iterative scheme can be determined by:

$$\max(\mu) \text{ for } \vec{\theta} \in T^{all} \tag{2.22}$$

An estimate of the asymptotic convergence speed of multi-grid cycles can be found by considering the convergence of high-frequency components only. These components need to be reduced by the relaxation, so that the residual becomes smooth and can accurately be represented on a coarse grid. The smoothing factor (μ_{loc}) takes only the smoothing of higher frequencies into account and is given by:

$$\mu_{loc} = \max(\mu) \text{ for } \vec{\theta} \in T^{high} \tag{2.23}$$

where $T^{high} = [-\pi, \pi]^2 \setminus [-\pi/2, \pi/2]^2$ (i.e. all frequency components except for $|\theta_1| \leq \pi/2$ and $|\theta_2| \leq \pi/2$). [12]

2.2.2 Two-grid analysis

The two-grid analysis, developed by Brandt [4], is used to gain insight into the convergence properties of a multi-grid algorithm. This analysis (just as the LMA) is based on writing the solution and the error as a Fourier series. The two-grid analysis takes into account the effects of all operators (especially the two-grid operator, equation 1.9) on the set of Fourier functions (or grid functions):

$$\phi^h\left(\vec{\theta},\vec{x}\right) = e^{i\vec{\theta}\cdot\vec{x}/h} = e^{\frac{i}{h}(\theta_1 x_1+\theta_2 x_2)} \tag{2.24}$$

with $\theta_1,\theta_2 \in [-\pi,\pi]$, and its coarse grid equivalent:

$$\phi^H\left(\vec{\theta},\vec{x}\right) = e^{i\vec{\theta}\cdot\vec{x}/H} = e^{\frac{i}{H}(\theta_1 x_1+\theta_2 x_2)} \tag{2.25}$$

These grid functions are separated in high and low-frequency components on grid Ω^H with respect to Ω^h. The distinction is based on the phenomenon that only those functions, with frequencies $\theta_1,\theta_2 \in [-\pi/2,\pi/2]$ are distinguishable on Ω^H. For each low-frequency component $\theta_1,\theta_2 \in [-\pi/2,\pi/2]$ three other (high) frequency components alias on Ω^H with $\phi^h(\vec{\theta'},\vec{x})$ and are not distinguishable on Ω^H. These four linearly independent grid functions are called harmonics. The influence of the operators on these four harmonics at once is considered:

$$\vec{\theta}^{(0,0)} = (\theta_1,\theta_2) \qquad \vec{\theta}^{(1,1)} = \left(\bar{\theta_1},\bar{\theta_2}\right) \qquad \vec{\theta}^{(1,0)} = \left(\bar{\theta_1},\theta_2\right) \qquad \vec{\theta}^{(0,1)} = \left(\theta_1,\bar{\theta_2}\right)$$

where

$$\bar{\theta_i} = \begin{cases} \theta_i + \pi & \text{if } \theta_i < 0 \\ \theta_i - \pi & \text{if } \theta_i \geq 0 \end{cases} \tag{2.26}$$

These four grid function are combined in one grid functionvector:

$$\vec{\phi}^h\left(\vec{\theta},\vec{x}\right) = \begin{bmatrix} \phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right) \\ \phi^h\left(\vec{\theta}^{(1,1)},\vec{x}\right) \\ \phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right) \\ \phi^h\left(\vec{\theta}^{(0,1)},\vec{x}\right) \end{bmatrix} \tag{2.27}$$

Now each grid variable can be written as a sum over the Fourier functions:

$$\begin{aligned} u^h &= \sum_{\vec{\theta}} \vec{A}^h\left(\vec{\theta}\right)\cdot\vec{\phi}^h\left(\vec{\theta},\vec{x}\right) \\ u^H &= \sum_{\vec{\theta}} A^H\left(\vec{\theta}\right)\phi^H\left(\vec{\theta},\vec{x}\right) \end{aligned} \tag{2.28}$$

With the help of trigonometric identities special relations for the four grid functions can be derived:

$$\begin{aligned}
&\phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(1,1)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(0,1)},\vec{x}\right) && \text{if } x_1/h \text{ and } x_2/h \text{ even} \\
&\phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(1,1)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(0,1)},\vec{x}\right) && \text{if } x_1/h \text{ and } x_2/h \text{ odd} \\
&\phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(1,1)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(0,1)},\vec{x}\right) && \text{if } x_1/h \text{ odd, } x_2/h \text{ even} \\
&\phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(1,1)},\vec{x}\right) = \phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right) = -\phi^h\left(\vec{\theta}^{(0,1)},\vec{x}\right) && \text{if } x_1/h \text{ even, } x_2/h \text{ odd}
\end{aligned} \tag{2.29}$$

The influence of the smoothing, fine grid and coarse grid operators on the grid function vector is problem specific and will not be treated in this section, but later in the relevant chapters. For restriction and interpolation standard stencils are usually employed. The influence of these operators on the grid functions is discussed in the two following sections.

Restriction

In this section the influence of the restriction operator on the grid function vector (equation 2.27) is studied for full weighting (2.9) and injection (2.7). All four components of the grid function are mapped onto its low-frequency harmonic. The amplitude of the resulting low-frequency are calculated by the yet to be calculated Fourier symbol $\tilde{I}_h^H\left(2\vec{\theta}\right)$:

$$I_h^H\left\langle \vec{A}^h\left(\vec{\theta}\right)\cdot\vec{\phi}^h\left(\vec{\theta},\vec{x}\right)\right\rangle = \tilde{I}_h^H\left(\vec{\theta}\right)\vec{A}^h\left(\vec{\theta}\right)\phi^H\left(2\vec{\theta},\vec{x}\right) \tag{2.30}$$

Injection First consider the injection operator (equation 2.7) acting on a low-frequency grid function at coarse grid point $\vec{x}=(x_1,x_2)$.

$$\begin{aligned} I_h^H\left\langle \phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right)\right\rangle &= e^{\frac{i}{h}(\theta_1x_1+\theta_2x_2)} \\ &= e^{\frac{i}{H}(2\theta_1x_1+2\theta_2x_2)} \\ &= \phi^H\left(2\vec{\theta},\vec{x}\right) \end{aligned} \tag{2.31}$$

Now consider the injection operator acting on a high-frequency grid function in one direction at coarse gridpoint $\vec{x}=(x_1,x_2)$.

$$\begin{aligned} I_h^H\left\langle \phi^h\left(\vec{\theta}^{(1,0)},\vec{x}\right)\right\rangle &= e^{\frac{i}{h}((\theta_1\pm\pi)x_1+\theta_2x_2)} \\ &= e^{\frac{i}{H}(2\theta_1x_1+2\theta_2x_2)}e^{\pm\frac{i\pi x_1}{h}} \\ &= \phi^H\left(2\vec{\theta},\vec{x}\right) \end{aligned} \tag{2.32}$$

Here the last step is allowed because x_1/h is even for coarse grid points. This can be carried out in a similar way for the other two high-frequency grid functions, resulting in the Fourier symbol for injection of:

$$\tilde{I}_h^H\left(\vec{\theta}\right) = \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right] \tag{2.33}$$

Thus all Fourier components are mapped onto its low-frequency harmonic without a change in amplitude. For low-frequency components this is exactly what is needed. Because these low-frequency residuals are accurately described on the coarse grid, and thus accurate corrections for the fine grid can be calculated. However, the fact that all high-frequency components are mapped onto its low-frequency harmonic is unwanted, since the coarse grid will give unwanted corrections for the low-frequency harmonics. This may not harm convergence as pre-relaxations have already reduced the amplitude of high-frequency grid functions, before the residual is restricted to the coarse grid. However, in general full weighting is safer to use.

Full weighting Now consider the full weighting operator (equation 2.9) acting on a low-frequency grid function at coarse grid point $\vec{x}=(x_1,x_2)$.

$$\begin{aligned} I_h^H\left\langle \phi^h\left(\vec{\theta}^{(0,0)},\vec{x}\right)\right\rangle &= e^{\frac{i}{h}(\theta_1x_1+\theta_2x_2)}\frac{1}{16}\begin{pmatrix} & e^{-i(\theta_1+\theta_2)} & + & 2e^{i\theta_2} & + & e^{i(\theta_1+\theta_2)} \\ + & 2e^{-i\theta_1} & + & 4 & + & 2e^{i\theta_1} \\ + & e^{-i(\theta_1-\theta_2)} & + & 2e^{-i\theta_2} & + & e^{i(\theta_1-\theta_2)} \end{pmatrix} \\ &= \frac{1}{4}e^{\frac{i}{H}(2\theta_1x_1+2\theta_2x_2)}\left(1+\cos\theta_1+\cos\theta_2+\cos\theta_1\cos\theta_2\right) \\ &= \phi^H\left(2\vec{\theta},\vec{x}\right)\frac{1}{4}\left(1+\cos\theta_1\right)\left(1+\cos\theta_2\right) \end{aligned} \tag{2.34}$$

Now consider the full weighting operator acting on a high-frequency grid function at coarse grid point $\vec{x} = (x_1, x_2)$.

$$\begin{aligned} I_h^H \left\langle \phi^h \left(\vec{\theta}^{(1,0)}, \vec{x} \right) \right\rangle &= e^{\frac{i}{h}(\theta_1 x_1 + \theta_2 x_2)} \frac{1}{16} \begin{pmatrix} & e^{-i(\theta_1 \pm \pi + \theta_2)} & + & 2e^{i\theta_2} & + & e^{i(\theta_1 \pm \pi + \theta_2)} \\ + & 2e^{-i(\theta_1 \pm \pi)} & + & 4 & + & 2e^{i(\theta_1 \pm \pi)} \\ + & e^{-i(\theta_1 \pm \pi - \theta_2)} & + & 2e^{-i\theta_2} & + & e^{i(\theta_1 \pm \pi - \theta_2)} \end{pmatrix} \\ &= \frac{1}{4} e^{\frac{i}{H}(2\theta_1 x_1 + 2\theta_2 x_2)} \left(1 + \cos(\theta_1 \pm \pi) + \cos\theta_2 + \cos(\theta_1 \pm \pi)\cos\theta_2\right) \\ &= \frac{1}{4} e^{\frac{i}{H}(2\theta_1 x_1 + 2\theta_2 x_2)} \left(1 - \cos\theta_1 + \cos\theta_2 - \cos\theta_1 \cos\theta_2\right) \\ &= \phi^H \left(2\vec{\theta}, \vec{x}\right) \frac{1}{4} \left(1 - \cos\theta_1\right)\left(1 + \cos\theta_2\right) \end{aligned} \tag{2.35}$$

The same analysis can be performed for the other two high-frequency grid functions, resulting in the Fourier symbol for full weighting of:

$$\tilde{I}_h^H \left(\vec{\theta}\right) = \frac{1}{4} \begin{bmatrix} (1 + \cos\theta_1)(1 + \cos\theta_2) \\ (1 - \cos\theta_1)(1 - \cos\theta_2) \\ (1 - \cos\theta_1)(1 + \cos\theta_2) \\ (1 + \cos\theta_1)(1 - \cos\theta_2) \end{bmatrix}^T \tag{2.36}$$

Thus each grid function is mapped onto its low-frequency harmonic, but with a smaller amplitude. The decrease in amplitude depends on the original frequency, where low-frequency modes are transferred correctly, whereas high-frequency modes are nearly damped out. For most cases full weighting gives better performance than injection, because high-frequency errors are reduced by pre-relaxations and not carried along by restriction. Therefore they do not alias to the coarse grid and thus do not give any spurious corrections.

Interpolation

In this section the influence of the interpolation operator on the coarse grid functions (equation 2.25) is studied for bi-linear interpolation (equation 2.3). This interpolation maps one coarse grid function onto the four grid functions in the grid functionvector:

$$I_H^h \left\langle \vec{A}^H \left(2\vec{\theta}\right) \phi^H \left(2\vec{\theta}, \vec{x}\right) \right\rangle = \tilde{I}_H^h \left(\vec{\theta}\right) \vec{A}^H \left(2\vec{\theta}\right) \vec{\phi}^h \left(\vec{\theta}, \vec{x}\right) \tag{2.37}$$

with $\tilde{I}_H^h \left(\vec{\theta}\right)$ the Fourier symbol for interpolation yet to be calculated.

Four different cases need to be studied: x_1/h is even or odd and x_2/h is even or odd. First consider the result at point (x_1, x_2) of the bi-linear interpolation operator acting on a coarse grid function, where both x_1/h and x_2/h are even (i.e. a coarse grid point).

$$\begin{aligned} I_H^h \left\langle \phi^H \left(2\vec{\theta}, \vec{x}\right) \right\rangle &= e^{\frac{i}{H}(2\theta_1 x_1 + 2\theta_2 x_2)} \\ &= e^{\frac{i}{h}(\theta_1 x_1 + \theta_2 x_2)} \\ &= \phi^h \left(\vec{\theta}^{(0,0)}, \vec{x}\right) \end{aligned} \tag{2.38}$$

Now consider the result at point (x_1, x_2) of the bi-linear interpolation operator acting on a coarse grid function, where both x_1/h and x_2/h are odd.

$$\begin{aligned} I_H^h \left\langle \phi^H \left(2\vec{\theta}, \vec{x}\right) \right\rangle &= e^{\frac{i}{H}(2\theta_1 x_1 + 2\theta_2 x_2)} \frac{1}{4} \begin{pmatrix} & e^{-i\theta_1} e^{i\theta_2} & + & e^{i\theta_1} e^{i\theta_2} \\ + & e^{-i\theta_1} e^{-i\theta_2} & + & e^{i\theta_1} e^{-i\theta_2} \end{pmatrix} \\ &= e^{\frac{i}{h}(\theta_1 x_1 + \theta_2 x_2)} \cos\theta_1 \cos\theta_2 \\ &= \phi^h \left(\vec{\theta}^{(0,0)}, \vec{x}\right) \cos\theta_1 \cos\theta_2 \end{aligned} \tag{2.39}$$

The same analysis can be performed for the other cases (i.e. one odd and one even ratio), resulting in:

$$I_H^h \left\langle \phi^H \left(2\vec{\theta}, \vec{x}\right)\right\rangle = \phi^h \left(\vec{\theta}^{(0,0)}, \vec{x}\right) \begin{cases} 1 & \text{if } x_1/h \text{ and } x_2/h \text{ even} \\ \cos\theta_1 \cos\theta_2 & \text{if } x_1/h \text{ and } x_2/h \text{ odd} \\ \cos\theta_1 & \text{if } x_1/h \text{ odd, } x_2/h \text{ even} \\ \cos\theta_2 & \text{if } x_1/h \text{ even, } x_2/h \text{ odd} \end{cases} \tag{2.40}$$

which can be transformed, with the help of relations (2.29) into the Fourier symbol for bi-linear interpolation:

$$\tilde{I}_H^h \left(\vec{\theta}\right) = \frac{1}{4} \begin{bmatrix} (1+\cos\theta_1)(1+\cos\theta_2) \\ (1-\cos\theta_1)(1-\cos\theta_2) \\ (1-\cos\theta_1)(1+\cos\theta_2) \\ (1+\cos\theta_1)(1-\cos\theta_2) \end{bmatrix} \tag{2.41}$$

which is exactly the transpose of the full weighting operator. Thus a coarse grid function is not only mapped onto its fine grid equivalent, but also on the three high-frequency harmonics.

Conclusion

The influence of the smoothing and fine and coarse grid operators on the grid function vector is problem specific and cannot be treated in this section. But when also their influence is known, the two-grid convergence factor (ρ_{loc}) is given by the largest eigenvalue of the two-level iteration matrix (for $|\theta_1| \leq \pi/2$ and $|\theta_2| \leq \pi/2$):

$$\hat{M}_h^H \left(\vec{\theta}\right) = \left(\hat{S}^h\right)^{\nu_2} \left(I - \tilde{I}_H^h \left(\tilde{L}^H\right)^{-1} \tilde{I}_h^H \hat{L}^h\right) \left(\hat{S}^h\right)^{\nu_1} \tag{2.42}$$

2.2.3 Simplified two-grid analysis

The two-grid analysis may become rather involved, in particular for 3D problems, where there will be three directions and thus 8 different Fourier components. Therefore, it is sometimes useful to analyze the smoothing procedure and the coarse grid correction separately. Instead of a complete two-grid analysis, a simplified two-grid analysis can be performed. The goal is to obtain some insight into the quality of the approximation of the fine grid operator (L^h) by the coarse grid operator (L^H), for very low frequencies. This analysis neglects high frequencies and the coupling of harmonics. For (very) low frequencies, the transfer operators almost act like identity operators and smoothing has nearly no effect. Therefore, the behavior of the two-level iteration matrix (equation 2.42) can be approximated by:

$$I - \left(\tilde{L}^H\right)^{-1} \tilde{L}^h \tag{2.43}$$

This term gives some insight into the quality of the coarse grid correction for very low frequencies, especially for problems with characteristic directions. If a low-frequency $\vec{\theta} = (\theta_1, \theta_2)$ along a characteristic direction with $\theta_2 = c\theta_1$ is considered, equation:

$$\lim_{\theta_1 \to 0} \left(1 - \frac{\tilde{L}^h \left(\vec{\theta}\right)}{\tilde{L}^H \left(2\vec{\theta}\right)}\right) \tag{2.44}$$

should give a very small number, otherwise multi-grid performance will be negatively influenced by a bad coarse grid correction for very low frequencies. See section 5.3 for an example for problems with characteristic directions.

As an alternative to obtaining insight into the quality of the approximation of the fine grid operator (L^h) by the coarse grid operator (L^H), a finite difference approximation can be used. In this method a Taylor series expansion of both the fine and the coarse grid operator are performed.

For good multi-grid performance it is required that the leading terms of the Taylor series expansion for both operators should be equal in each direction. See section 5.3 for an example, where this requirement is not fullfilled.

2.3 Performance

Multi-grid techniques are aimed at efficiently solving systems of algebraic equations. Therefore the question arises how much is gained in terms of efficiency, or how much faster can the discretized problem be solved using multi-grid compared to single-grid relaxations or alternative methods. Two questions arise: How many cycles are required to solve the problem, and, what is the cost of each cycle?

2.3.1 Number of cycles

The required number of cycles depends on the error reduction per cycle and the total error reduction required. For a small number of pre- and post-relaxations the error reduction per cycle follows from the smoothing factor (μ_{loc}, see section 2.2.1) or in general the two-grid convergence factor (ρ_{loc}, see section 2.2.2). The required error reduction depends on the order of the discretization. Generally, the task is not to solve the discretized problem exactly, but to solve the partial differential equation. By definition the exact solution of the discretized problem is still only an approximation to the solution of partial differential equation with an discretization error of $\mathcal{O}(h^m)$ (with m the order of the fine grid operator). Any approximate solution to the system with the same accuracy is as good an approximation to the continuous solution as the exact discrete solution. Therefore it is sufficient to solve the discretized problem up to an accuracy of $\mathcal{O}(h^m)$. Assuming an initial error of $\mathcal{O}(1)$, an estimate of the number of single grid iterations M_s required to reduce the error to the level of the discretization error can be obtained from:

$$\mu^{M_s} = \mathcal{O}(h^m) \tag{2.45}$$

which gives:

$$M_s = \mathcal{O}\left(\frac{m \ln(1/h)}{\ln(1/\mu)}\right) \tag{2.46}$$

Substituting $\mu = 1 - \mathcal{O}\left(n^{-2}\right)$ (for single-grid relaxations), and $h = 1/n$ (with n the number of points in one direction):

$$M_s = \mathcal{O}\left(n^2 \ln(n)\right) \tag{2.47}$$

and with $N = n^d$, where d is the dimension of the problem (N is the total number of points)

$$M_s = \mathcal{O}\left(\frac{N^{\frac{2}{d}}}{d} \ln(N)\right) \tag{2.48}$$

When multi-grid techniques are used μ is independent of the mesh-size and one obtains:

$$\begin{aligned} M_s &= \mathcal{O}(\ln(n)) \\ &= \mathcal{O}\left(\frac{1}{d} \ln(N)\right) \end{aligned} \tag{2.49}$$

Thus for large grids (i.e. larger values of n) multi-grid techniques require far less cycles than single-grid relaxations.

2.3.2 Work

As shown previously, to solve the problem up to the accuracy of the discretization error $\mathcal{O}(\ln(n))$ multi-grid cycles are required. Next the amount of work involved in a single cycle is estimated.

Therefore it is useful to introduce the concept of a Work Unit (WU). One WU is the equivalent of the amount of work of one relaxation on the finest grid. Assuming the fine grid contains N points, one WU is usually $\mathcal{O}(N)$ operations. The required work for a multi-grid cycle will be (neglecting the required work for intergrid routines):

$$\begin{aligned} W_{cycle} &= WU(\nu_1+\nu_2)\left(1+\gamma\left(\frac{h}{H}\right)^d+\gamma^2\left(\frac{h}{H}\right)^{2d}+\gamma^3\left(\frac{h}{H}\right)^{3d}+...\right) \\ &= WU(\nu_1+\nu_2)\sum_{i=0}^{\infty}\left(\gamma^i\left(\frac{h}{H}\right)^{id}\right) \end{aligned} \tag{2.50}$$

with h the fine grid spacing, H the coarse grid spacing γ the cycle parameter and d the dimension of the problem. For $\gamma(H/h)^d < 1$ this series is convergent with a maximum of

$$W_{cycle} \leq WU\frac{\nu_1+\nu_2}{1-\gamma\left(\frac{h}{H}\right)^d} \tag{2.51}$$

When $\gamma(H/h)^d \geq 1$ the infinite series is diverging, but only a limited number of levels ($\mathcal{O}(\ln(N))$) is required to produce a coarse grid with $\mathcal{O}(1)$ points. In this case the work for a multi-level cycle will be $\mathcal{O}(N)$ times that of a single-grid cycle.

In this report only 2D problems are considered with standard coarsening ($H = 2h$), giving:

$$W_{cycle} \leq \frac{4}{3}WU(\nu_1+\nu_2) \tag{2.52}$$

for $\gamma = 1$ and for $\gamma = 2$:

$$W_{cycle} \leq 2WU(\nu_1+\nu_2) \tag{2.53}$$

So the work required to perform one multi-grid cycle is of the same order as the work required for a few relaxations on the finest grid. The total work required to solve the discretized problem up to discretization accuracy is now the required work for one cycle multiplied with the required number of cycles:

$$W_{cycles} = \mathcal{O}\left(\frac{N}{d}\ln(N)\right) \tag{2.54}$$

whereas the total work required for single-grid relaxation is:

$$W_{single-grid} = \mathcal{O}\left(\frac{N^{\frac{2}{d}+1}}{d}\ln(N)\right) \tag{2.55}$$

For large grids (i.e. large N) a huge reduction in computational time will thus be obtained. However, the required work to solve the problem to the order of the discretization error, is not yet independent of the number of points. It still contains a factor $\ln(N)$. Although this factor grows very slowly with increasing N a more elegant approach is possible. [13]

2.4 Full multi-grid

The factor $\ln(N)$ in the required work to solve a discretized problem up to discretization accuracy (equation 2.54) stems from the required number of cycles M_S (see equation 2.49). In this analysis it was assumed that the initial error is $\mathcal{O}(1)$. However, if the converged solution on a coarser grid Ω^H is used as the starting solution on the fine grid, the error of the initial approximation will no longer be $\mathcal{O}(1)$, but be equal to the discretization error on the coarser grid (i.e. $\mathcal{O}(H^m)$ with m the order of the discretization). In that case the number of cycles required to reach a converged solution on the fine grid follows from:

$$\mu^{M_s}H^m < h^m \tag{2.56}$$

giving:

$$M_s > \frac{m \ln(H/h)}{\ln(1/\mu)} = \frac{m \ln(2)}{\ln(1/\mu)} \tag{2.57}$$

which clearly is independent of the mesh-size. So to ensure that a constant number of cycles is sufficient to reach an accuracy of the order of the discretization error, the solution from a coarser grid has to be used as a starting solution. This principle can then be applied recursively leading to the so-called Full Multi-Grid (FMG) algorithm. A FMG algorithm is depicted in figure 2.5. A new parameter M is introduced which gives the number of cycles performed at each level to obtain a converged solution. The amount of work required in a FMG algorithm with M cycles

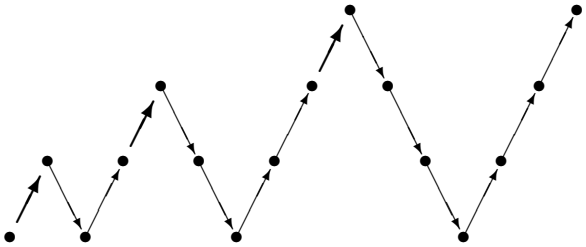


Figure 2.5: Flow diagram for FMG using 4 grids

per level can now be estimated by:

$$\begin{aligned} W_{FMG} &= W_{cycle} M \left(1 + \left(\frac{h}{H}\right)^d + \left(\frac{h}{H}\right)^{2d} + \left(\frac{h}{H}\right)^{3d} + ...\right) \\ &= W_{cycle} M \sum_{i=0}^{\infty} \left(\frac{h}{H}\right)^{id} \\ &\leq \frac{W_{cycle} M}{1 - \left(\frac{h}{H}\right)^d} \end{aligned} \tag{2.58}$$

As only $\mathcal{O}(N)$ operations are needed for one multi-grid cycle and since the problem can be solved in $\mathcal{O}(1)$ multi-grid cycles per level, the total work to solve the problem is $\mathcal{O}(N)$.

A point of special attention is the interpolation of the converged coarse grid solution to serve as an initial estimate for the solution on the finer grid. As the object here is to accurately approach the solution on the fine grid, a higher-order interpolation operators is required. The order of the FMG interpolation needs to be larger than the order of discretization error [15]. Also FMG may not work in cases for which the coarse grid operator does not resemble the fine grid operator accurately, i.e. problems with characteristic directions (see chapter 5).

FMG has the additional advantage of yielding accurate solutions to the problem on coarser grids. This allows for the calculation of the approximate error in the solution (e^h_{app}):

$$e^h_{app} = \left|\tilde{u}^H - I^h_H \tilde{u}^h\right|_2 \tag{2.59}$$

Expanding this relation and neglecting the effect of restriction:

$$\begin{aligned} e^h_{app} &= \left|\tilde{u}^H - \tilde{u}^h\right|_2 \\ &= \left|u^H - v^H - u^h + v^h\right|_2 \\ &= \left|u + \tau^H - v^H - u - \tau^h + v^h\right|_2 \\ &= \left|\tau^H - v^H - \tau^h + v^h\right|_2 \end{aligned} \tag{2.60}$$

with τ^H and τ^h the discretization errors (the difference between the solution of the partial differential equation (u^*) and the discrete solution (u^H and u^h)) and v^H and v^h the difference between

the exact discrete solution and the current approximation ($\tilde{u}^H$ and $\tilde{u}^h$). When using an additional cycle per level, both v^H and v^h should decrease significantly. However, when in that case the approximate error does not change, τ^H and τ^h already dominate the error and thus the accuracy of $\tilde{u}^h$ is equal to the discretization accuracy. In general, when the order of the discretization, m is known and enough cycles per level are used ($v^H \ll \tau^H$ and $v^h \ll \tau^h$), a quantitative estimate of the discretization error can be made: [13]

$$\begin{aligned} e^h_{app} &= \left|\tau^H - v^H - \tau^h + v^h\right|_2 \\ &\approx \left|\tau^H - \tau^h\right|_2 \\ &\approx \left|\left(\frac{H}{h}\right)^m \tau^h - \tau^h\right|_2 \\ \left|\tau^h\right|_2 &\approx \frac{1}{\left(\frac{H}{h}\right)^m - 1} e^h_{app} \end{aligned} \tag{2.61}$$

2.5 Full Approximation scheme

When the equations are non-linear, the step from equation (1.4) to (1.5) is not allowed. In such cases not only the residual, but also the solution itself needs to be represented on the coarse grid. The equations for such schemes are obtained by substituting the error equation (1.2) into equation (1.4):

$$\tilde{r}^h = L^h\left\langle \tilde{u}^h + v^h \right\rangle - L^h\left\langle \tilde{u}^h \right\rangle = f^h - L^h\left\langle \tilde{u}^h \right\rangle \tag{2.62}$$

This leads to a new set of coarse grid equations:

$$\begin{aligned} L^H\left\langle \tilde{u}^H + v^H \right\rangle - L^H\left\langle \tilde{u}^H \right\rangle &= \tilde{r}^H \\ L^H\left\langle I^H_h\left\langle \tilde{u}^h \right\rangle + v^H \right\rangle - L^H\left\langle I^H_h\left\langle \tilde{u}^h \right\rangle \right\rangle &= I^H_h\left\langle \tilde{r}^h \right\rangle \\ L^H\left\langle I^H_h\left\langle \tilde{u}^h \right\rangle + v^H \right\rangle &= I^H_h\left\langle f^h - L^h\left\langle \tilde{u}^h \right\rangle \right\rangle + L^H\left\langle I^H_h\left\langle \tilde{u}^h \right\rangle \right\rangle \end{aligned} \tag{2.63}$$

This equation can be approximated (with $I^H_h\left\langle \tilde{u}^h \right\rangle$ as a first approximation) to obtain v^H and equation (1.7) can be used to update the solution on the fine level. This approach is called the Full Approximation Scheme (FAS). For linear problems, the FAS and the CS give identical results up to machine accuracy. Note that with FAS it is possible to solve non-linear problems with the same efficiency as linear problems.

2.5.1 τ-Extrapolation

A variety of more sophisticated multi-grid techniques is based on the FAS method. One example is τ-Extrapolation, in which a more accurate approximation of the differential equations obtained using a low-order discretization. Consider a discretized problem:

$$L^h\left\langle u^h \right\rangle = f^h \tag{2.64}$$

using the FAS scheme the coarse grid problem becomes:

$$\begin{aligned} L^H\left\langle u^H \right\rangle &= L^H\left\langle I^H_h\left\langle \tilde{u}^h \right\rangle \right\rangle + I^H_h\left\langle f^h - L^h\left\langle \tilde{u}^h \right\rangle \right\rangle \\ &= I^H_h\left\langle f^h \right\rangle + \tau^H_h\left\langle \tilde{u}^h \right\rangle \end{aligned} \tag{2.65}$$

where

$$\tau^H_h\left\langle u^h \right\rangle = L^H\left\langle I^H_h\left\langle u^h \right\rangle \right\rangle - I^H_h\left\langle L^h\left\langle u^h \right\rangle \right\rangle \tag{2.66}$$

$\tau^H_h\left\langle u^h \right\rangle$ is called the (h, H)-relative truncation error. $\tau^H_h\left\langle u^h \right\rangle$ plays a role similar to the truncation error of the continuous solution u with respect to the grids Ω^h and Ω^H. $\tau^H_h\left\langle u^h \right\rangle$ is the required

correction for the coarse grid equation, so that the solution of this coarse grid equation becomes the fine grid solution:

$$\begin{aligned} L^H \langle u^H \rangle &= I_h^H \langle f^h \rangle + \tau_h^H \langle u^h \rangle \\ &= I_h^H \langle f^h \rangle + L^H \langle I_h^H \langle \bar{u}^h \rangle \rangle - I_h^H \langle L^h \langle u^h \rangle \rangle \\ &= I_h^H \langle f^h - L^h \langle u^h \rangle \rangle + L^H \langle I_h^H \langle u^h \rangle \rangle \\ &= L^H \langle I_h^H \langle u^h \rangle \rangle \end{aligned} \tag{2.67}$$

And thus

$$u^H = I_h^H \langle u^h \rangle \tag{2.68}$$

The next step is to realize that $\tau_h^H \langle u^h \rangle$ is related to to $\tau^H \langle u^h \rangle$ (the truncation error) and thus one can also directly use the coarse grid equation to correct the fine grid solution to the exact solution by adding the relative truncation error multiplied with a constant factor:

$$L^H \langle u_\tau^H \rangle = I_h^H \langle f^h \rangle + \frac{2^m}{2^m - 1} \tau_h^H \langle \tilde{u}^h \rangle \tag{2.69}$$

with m the order of the discretization operator. This operator will produce an approximation u_τ^H, which will have a better accuracy than that of u^h. This can be proven with some expansions of the fine and coarse grid operators (with q the order of the second term in the truncation error ($m < q$)):

$$\begin{aligned} L^h \langle u \rangle - L \langle u^* \rangle &= e_1 h^m + \mathcal{O}(h^q) \\ L^H \langle u \rangle - L \langle u^* \rangle &= e_1 2^m h^m + \mathcal{O}(h^q) \\ u^h - u^* &= e_2 h^m + \mathcal{O}(h^q) \end{aligned} \tag{2.70}$$

Substituting these relations into (2.69) and neglecting the effect of the restriction, which is justified because it mainly concerns smooth components, result in:

$$\begin{aligned} L^H \langle u_\tau^H \rangle &= f^h + \frac{2^m}{2^m - 1} \left(L^H \langle u^h \rangle - L^h \langle u^h \rangle \right) \\ &= f + \frac{2^m}{2^m - 1} \left(L^H \langle u^* \rangle - L^h \langle u^* \rangle \right) + \frac{2^m h^m}{2^m - 1} \left(L^H \langle e_2 \rangle - L^h \langle e_2 \rangle \right) + \mathcal{O}(h^q) \\ &= f + \frac{2^m}{2^m - 1} \left(e_1 2^m h^m - e_1 h^p \right) + \mathcal{O}\left(h^{2m}\right) + \mathcal{O}(h^q) \\ &= L \langle u^* \rangle + e_1 2^m h^n + \mathcal{O}\left(h^{2m}\right) + \mathcal{O}(h^q) \\ &= L^H \langle u^* \rangle + \mathcal{O}\left(h^{2m}\right) + \mathcal{O}(h^q) \end{aligned} \tag{2.71}$$

Thus the order of accuracy of u_τ^H is the minimum of $2m$ and q. The only required adjustment for τ-extrapolation is to use equation (2.69) as the coarse grid problem on the second finest level of the last cycle. In all other cases the regular coarse grid problem has to be used. The additional amount of work compared to standard multi-grid is only one additional multiplication per coarse grid point.

Only the u_τ^H is higher-order accurate, u^h itself not, therefore post-relaxations on the finest grid tend to reduce the accuracy of the approximation back to order m. The easiest way to counter act this effect is by not performing any post-smoothing steps on the finest level. A more subtle approach is by also correcting the right-hand side of the fine grid equation after the pre-relaxations have been performed:

$$\tilde{f}^h = f^h + \frac{1}{2^m - 1} I_H^h \langle \tau_h^H \langle u^h \rangle \rangle \tag{2.72}$$

For more detailed description and optimal approaches see [2] and [1].

Chapter 3

Algebraic multi-grid

This chapter discusses different aspects of algebraic multi-grid (AMG). AMG is based on the same principles as geometric multi-grid:

- Simple relaxations
- Identification the slowly converging components
- Accurately representing these problematic components on a restricted set of variables
- Solving the matrix of reduced size by either applying the steps above recursively or by using a direct solver

In contrast to geometrically based multi-grid, AMG does not require a given problem to be defined on a grid; it operates directly on a system of linear, algebraic equations:

$$A\vec{u} = \vec{f} \tag{3.1}$$

As AMG does not depend on physical grids, some components of geometric multi-grid need to be replaced by their AMG equivalent: grids become sets of variables, subgrids become subsets of variables and grid points become single variables. Coarse-grid discretizations used in geometric multi-grid to reduce low-frequency error components now correspond to certain matrix equations of reduced dimension to represent components that are slow to convergence. However, no grid hierarchy needs to be known a priori. In fact the construction of a hierarchy, including the coarsening process and the transfer operators, is part of the AMG algorithm. The only element of the procedure that is determined in advance is the choice of the relaxation scheme. Usually simple relaxation schemes like Gauß-Seidel or Kaczmarz relaxation are used. For historical reasons most of the times throughout this thesis the geometrical multi-grid terms are used even though no actual grid needs to be present.

In AMG the coarse-grid operator is always constructed using the Galerkin condition:

$$A^H = I_h^H A^h I_H^h \tag{3.2}$$

with the restriction operator, defined as the transpose of the interpolation operator, i.e.:

$$I_h^H = \left(I_H^h\right)^T \tag{3.3}$$

The remaining choice of coarse grids and interpolation operators is guided by two principles: Algebraic smoothness and strong influence and dependence. In this chapter a basic description of AMG is given, for a more detailed explanation see [5, 12].

Notation

For ease of notation first some sets of variables are defined:

C	Set of coarse grid points
F	Set of fine grid points
U	Set of yet undecided grid points
N_i	Set of neighboring points $\{j \in \Omega : j \neq i, A_{i,j} \neq 0\}$
S_i	Set of strongly connected neighboring points (see equation 3.5)
S_i^T	Set of points strongly influenced by i (see equation 3.6)
W_i	Set of weakly connected neighboring points ($W_i = N_i \setminus S_i$)
C_i^s	Set of strongly connected neighboring coarse grid points ($C_i^s = C \cup S_i$)
F_i^s	Set of strongly connected neighboring fine grid points ($F_i^s = F \cup S_i$)
Ω	Set of all points

3.1 Influence and Dependence

The first concept used in AMG is that of strong dependence and strong influence. Because of the usual dominance of diagonal entries, the i^{th} equation is associated with the i^th unknown; the purpose of the i^{th} equation is to determine the value of u_i. Of course, it usually takes all of the equations to determine any given variable precisely, but certainly contributions due to some u_j's are more important in the i^{th} equation to determine u_i than others.

When the coefficient $A_{i,j}$, which multiplies u_j in the i^{th} equation, is large relative to the other coefficients in the i^{th} equation, then a small change in the value of u_j has more effect on the value of u_i than a small change in any other variable appearing in the i^{th} equation. If a variable is important for determining the value u_i, it would be a good variable to use in the interpolation of u_i. To determine if a variable u_j is important for the i^{th} equation, the following definition is used:

Definition Given a threshold value $0 < \epsilon_{str} \leq 1$, the variable u_i is *strongly negatively dependent* on the variable u_j if

$$-A_{i,j} \geq \epsilon_{str} \max_{k \neq i} (-A_{i,k}) \tag{3.4}$$

This states that variable u_i has strong negative dependence on variable u_j if the coefficient $A_{i,j}$ is comparable in magnitude to the largest off-diagonal coefficient in the i^{th} equation. All variables j which are important for the i^{th} equation are stored in the set S_i.

$$S_i = \left\{ j \in N_i : -A_{i,j} \geq \epsilon_{str} \max_{k \neq i} (-A_{i,k}) \right\} \tag{3.5}$$

This definition can also be stated from another perspective:

Definition If the variable u_i strongly depends on the variable u_j, then the variable u_j *strongly influences* the variable u_i.

All variables j which are strongly influenced by the i^{th} variable are stored in the set S_i^T:

$$S_i^T = \left\{ j \in N_i : -A_{j,i} \geq \epsilon_{str} \max_{k \neq i} (-A_{j,k}) \right\} \tag{3.6}$$

3.2 Algebraic smoothness

The crux of the AMG algorithm is to determine which components are slow to converge on the fine grid (i.e. what is meant by a smooth error) and how to describe these components on coarse grids. In geometric multi-grid Fourier modes are used to examine which components are slow to converge. In AMG the true location of the grid points is usually unknown, so it is impossible to

examine the Fourier modes of the error. Instead we define smooth errors loosely to be errors that are not reduced effectively by relaxation:

$$S\vec{e} \approx \vec{e} \tag{3.7}$$

Here S is a relaxation operator and $\vec{e}$ the error defined as:

$$\vec{e} = \vec{u} - \vec{u}^* \tag{3.8}$$

with $\vec{u}$ the current approximation and $\vec{u}^*$ the exact solution to the system of equations. Several authors [6, 12] already proved that for symmetric positive definite matrices A, a smooth error implies:

$$D^{-1}A\vec{e} \cdot A\vec{e} \ll A\vec{e} \cdot \vec{e} \tag{3.9}$$

with D the diagonal part of A. In terms of the residual ($\vec{r} = A\vec{e}$) this becomes:

$$D^{-1}\vec{r} \cdot \vec{r} \ll \vec{r} \cdot \vec{e} \tag{3.10}$$

This indicates that, on average, algebraically smooth errors are characterized by (scaled) residuals which are much smaller than the error itself. This can also be seen more directly. For instance, consider Jacobi relaxation at point i. This corresponds to replacing u_i by $\bar{u}_i$, where:

$$\bar{u}_i = \frac{1}{A_{i,i}}\left(f_i - \sum_{j \neq i} A_{i,j}u_j\right) = \frac{1}{A_{i,i}}\left(A_{i,i}u_i + f_i - \sum_{j} A_{i,j}u_j\right) = u_i - \frac{r_i}{A_{i,i}} \tag{3.11}$$

with r_i the residual at point i before relaxation. In terms of the error this becomes:

$$\bar{e}_i = e_i - \frac{r_i}{A_{i,i}} \tag{3.12}$$

with e_i the error before and $\bar{e}_i$ the error after relaxation. For an algebraically smooth error the error before and after relaxation are roughly equal ($e_i \approx \bar{e}_i$), leading to:

$$|r_i| \ll A_{i,i}\,|e_i| \tag{3.13}$$

That is, although the error may still be quite large globally, locally we can approximate e_i as a function of its neighboring error values e_j by evaluating:

$$(r_i =) \sum_j A_{i,j}e_j = A_{i,i}e_i + \sum_{j \in N_i} A_{i,j}e_j = 0 \tag{3.14}$$

3.2.1 Interpretation of algebraically smooth error

In this section a more intuitive interpretation of an algebraically smooth error is given for symmetric positive matrices. These are matrices with positive diagonal and negative off-diagonal terms. Such matrices often result from the discretization of scalar elliptic PDE's. Consider an error that satisfies equation 3.9 and thus is algebraically smooth. With the help of the Schwarz' inequality it can be proven that this error also satisfies: [12]

$$A\vec{e} \cdot \vec{e} \ll D\vec{e} \cdot \vec{e} \tag{3.15}$$

Or equivalently:

$$\sum_{i,j} A_{i,j}e_ie_j \ll \sum_i A_{i,i}e_i^2 \tag{3.16}$$

For the type of matrices considered this can be rewritten into:

$$\frac{1}{2}\sum_{i,j} |A_{i,j}|\,(e_i - e_j)^2 + \sum_i s_i e_i^2 \ll \sum_i A_{i,i}e_i^2 \tag{3.17}$$

with $s_i = \sum_j A_{i,j}$ the i-th row sum of A. This means, that in the most important case of $s_i \approx 0$, on average for each i:

$$\sum_{j \neq i} \frac{|A_{i,j}|}{A_{i,i}} \frac{(e_i - e_j)^2}{e_i^2} \ll 1 \tag{3.18}$$

That is, an algebraically smooth error varies slowly in the direction of large negative connections. Or in other words, relaxation schemes smooth the error along strong negative connections. It can be proven that this conclusion also holds for matrices which are essentially positive-type [3], which are defined by the fact that there exists a constant $c > 0$ such that, for all $\vec{e}$.

$$-\sum_{i,j} A_{i,j} (e_i - e_j)^2 \leq -c \sum_{i,j} A^-_{i,j} (e_i - e_j)^2 \tag{3.19}$$

with $A^-_{i,j}$ the negative off-diagonal parts of A. Essentially positive-type matrices often result from the discretization of higher-order difference approximations to second-order elliptic PDE's or PDE's involving mixed derivatives. Such matrices have the property that for each $A_{i,j} > 0$ $(i \neq j)$ there exists a path of length two from i to j corresponding to relatively large negative connections. I.e. for each $i \neq j$ with $A_{i,j} > 0$ we can find another variable k, such that both $-A_{i,k} > \alpha A_{i,j}$ and $-A_{k,j} > \beta A_{i,j}$, where $\alpha > 1$, $\beta > 1$ and $(\alpha - 1)(\beta - 1) = 1$. Then we can write:

$$A_{i,j} (e_i - e_j)^2 = A_{i,j} \left(\alpha (e_i - e_k)^2 + \beta (e_k - e_j)^2 - \left(\bar{\alpha} e_i + \bar{\beta} e_j - \left(\bar{\alpha} + \bar{\beta} \right) e_k \right)^2 \right) \tag{3.20}$$

where $\bar{\alpha} = \sqrt{(\alpha - 1)}$ and $\bar{\beta} = \sqrt{(\beta - 1)}$. The first two terms will still give a negative contribution to the left hand side of equation 3.19. But these negative contribution are in absolute sense smaller than $-A_{i,k} (e_i - e_k)^2$ for the first term and $-A_{k,j} (e_j - e_k)^2$ for the second term, so always a $c > 0$ can be found to satisfy equation 3.19.

Example essentially positive-type

A simple example of an essentially positive-type matrix results from the higher-order discretization of the 1D Poisson equation. A fourth-order accurate discretization leads to the stencil:

$$-\frac{\partial^2 u}{\partial x^2} = \frac{1}{12h^2} \begin{bmatrix} 1 & -16 & 30 & -16 & 1 \end{bmatrix} \tag{3.21}$$

Ignoring boundary conditions, the matrix becomes (with the number 30 on the diagonal):

$$A = \frac{1}{12h^2} \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & & & & \\ & 1 & -16 & 30 & -16 & 1 & & & & \\ & & 1 & -16 & 30 & -16 & 1 & & & \\ & & & \mathbf{1} & -16 & 30 & -16 & 1 & & \\ & & & & 1 & -16 & 30 & -16 & 1 & \\ & & & & & 1 & -16 & 30 & -16 & 1 \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{3.22}$$

Positive off diagonal weight arise for example for $i = 6$ and $j = i - 2 = 4$, indicated by the bold number in equation 3.22. Equation 3.20 can now be used to prove equation 3.19. Choosing $k = i - 1$ and $\alpha = \beta = 2$ satisfies all required conditions:

$$\begin{gathered} 16 = -A_{i,i-1} > \alpha A_{i,i-2} = 2 \\ 16 = -A_{i-1,i-2} > \beta A_{i,i-2} = 2 \\ (\alpha - 1)(\beta - 1) = 1 \end{gathered}$$

Equation 3.20 now becomes:

$$A_{i,i-2} (e_i - e_{i-2})^2 = A_{i,i-2} \left(2 (e_i - e_{i-1})^2 + 2 (e_{i-1} - e_{i-2})^2 - (e_i + e_{i-2} - 2e_{i-1})^2 \right) \tag{3.23}$$

So that the left-hand side of equation 3.19 can be rewritten into:

$$-\sum_{i,j} A^*_{i,j} \left(e_i - e_j\right)^2 + A_{6,4} \left(e_6 + e_4 - 2e_5\right)^2 \tag{3.24}$$

with:

$$A^* = \frac{1}{12h^2} \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & & & & \\ & 1 & -16 & 0 & -16 & 1 & & & & \\ & & 1 & \mathbf{-14} & 0 & -16 & 1 & & & \\ & & & \mathbf{0} & \mathbf{-14} & 0 & -16 & 1 & & \\ & & & & 1 & -16 & 0 & -16 & 1 & \\ & & & & & 1 & -16 & 0 & -16 & 1 \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{3.25}$$

where the diagonal terms are set to zero because they are always multiplied by 0. The negative contribution of $A_{i,i-2}$ is now eliminated from the left-hand side. Applying this technique for all positive off diagonal terms yields the equation:

$$-\sum_{i,j} A^*_{i,j} \left(e_i - e_j\right)^2 + \sum_i A_{i,i-2} \left(e_i + e_{i-2} - 2e_{i-1}\right)^2 + \sum_i A_{i,i+2} \left(e_i + e_{i+2} - 2e_{i+1}\right)^2 \tag{3.26}$$

with:

$$A^* = \frac{1}{12h^2} \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & & & & \\ & 0 & -12 & 0 & -12 & 0 & & & & \\ & & 0 & -12 & 0 & -12 & 0 & & & \\ & & & 0 & -12 & 0 & -12 & 0 & & \\ & & & & 0 & -12 & 0 & -12 & 0 & \\ & & & & & 0 & -12 & 0 & -12 & 0 \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{3.27}$$

Now all terms will give positive contributions, so that equation 3.19 is easily satisfied with $c = 3/4$.

3.3 Interpolation and restriction

Before considering the splitting between fine and coarse grid points, consider first the interpolation operator. For geometric multi-grid linear interpolation between the coarse grid points is used, but with an unstructured or perhaps non-existent grid, it is not so obvious how to define interpolation. Assume that the grid is already split into coarse (C) and fine grid points (F). The goal is to define interpolation weights $w_{i,j}$, such that the interpolation operator:

$$\left(I^h_H \vec{e}\right)_i = \begin{cases} e_i & \text{if } i \in C \\ \sum_{j \in S_i} w_{i,j} e_j & \text{if } i \in F \end{cases} \tag{3.28}$$

yields a reasonable approximation for any algebraically smooth error which satisfies: (see equation 3.14)

$$A_{i,i} e_i \approx \sum_{i \in N_i} A_{i,j} e_j \tag{3.29}$$

3.3.1 Direct interpolation

In section 3.2.1 it has been shown that for symmetric positive matrices, an algebraically smooth error varies slowly in the direction of strong coupling. That is, the error at point i is essentially determined by a weighted average of the error at its strong neighbors. Consequently:

$$\frac{1}{\sum_{k\in C_i^s} A_{i,k}} \sum_{k\in C_i^s} A_{i,k}e_k \approx \frac{1}{\sum_{k\in N_i} A_{i,k}} \sum_{k\in N_i} A_{i,k}e_k \tag{3.30}$$

is a good approximation when most of the strong negative connections of any F-variable i are contained in C_i^s. This suggests approximating equation 3.29 by:

$$A_{i,i}e_i \approx -\alpha_i \sum_{k\in C_i^s} A_{i,k}e_k \text{ with } \alpha_i = \frac{\sum_{k\in N_i} A_{i,k}}{\sum_{k\in C_i^s} A_{i,k}} \tag{3.31}$$

Or equivalently:

$$w_{i,j} = -\frac{A_{i,j}}{A_{i,i}} \frac{\sum_{k\in N_i} A_{i,k}}{\sum_{k\in C_i^s} A_{i,k}} \text{ with } j \in C_i^s \tag{3.32}$$

Note that:

$$A_{i,i}\left(1 - \sum_{i\in S_i} w_{i,k}\right) = \sum_j A_{i,j} \tag{3.33}$$

Consequently, in the limiting case of zero row sum matrices (i.e. $\sum_j A_{i,j} = 0$), constants are interpolated exactly (i.e. $I_H^h 1^H = 1^h$). For other matrices, however, $I_H^h 1^H$ equals the result of one Jacobi step with the vector $e = 1^h$ as the starting vector. (Here 1^H and 1^h denote a vector with all components having the value of 1).

3.3.2 Standard interpolation

Direct interpolation requires $\emptyset \neq C_i^s \subset C \cap N_i$ to work. So each fine grid point is required to have at least one strong negative connection to a coarse grid point. In some splittings this is not always enforced. Also direct interpolation may give inaccurate results if not enough of the strong negative connections are represented in the interpolation. To obtain a more robust interpolation operator a small modification can be made: for each $i \in F$ its strong F-connections are also included in the interpolation. This is done by eliminating all e_j ($j \in F_i^s$) by means of the corresponding j^{th} equation before approximating equation 3.29.

$$e_j \rightarrow -\frac{\sum_{k\in C_j^s} A_{j,k}e_k}{A_{j,j}} \frac{\sum_{m\in N_j} A_{j,m}}{\sum_{m\in C_j^s} A_{j,m}} \tag{3.34}$$

This leads to the interpolation weights:

$$w_{i,j} = -\frac{\sum_{k\in N_i} A_{i,k}}{A_{i,i}} \frac{\left(A_{i,j} - \sum_{l\in F_i^s} \frac{A_{i,l}A_{l,j}}{A_{l,l}} \frac{\sum_{m\in N_l} A_{l,m}}{\sum_{m\in C_l^s} A_{l,m}}\right)}{\sum_{k\in P_i}\left(A_{i,k} - \sum_{l\in F_i^s} \frac{A_{i,l}A_{l,k}}{A_{l,l}} \frac{\sum_{m\in N_l} A_{l,m}}{\sum_{m\in C_l^s} A_{l,m}}\right)} \text{ with } j \in P_i \tag{3.35}$$

with P_i the union of C_i^s and all C_j^s ($j \in F_i^S$). This interpolation guarantees $0 \leq w_{i,j} \leq 1$ for matrices with positive diagonal terms and negative off-diagonal terms.

3.3.3 Truncation of interpolation

For interpolation, the sets P_i of interpolatory variables may become quite large. Consequently the resulting Galerkin operators will substantially increase for coarser levels. This process without

reasonable truncation, will generally be much too costly. However, interpolation weights corresponding to variables "far away" from variable i, will usually be much smaller than the largest weights. Therefore, before computing the coarser-level Galerkin operator, the full interpolation operator is always truncated by ignoring all interpolatory connections which are smaller than the largest one by a factor of ϵ_{tr} and rescaling the remaining weights so that the total sum remains unchanged.

3.4 Coloring

The last required component in the AMG process is the splitting of the variables between fine and coarse grid variables. The most common concept for this splitting is the so called standard coarsening. This process starts with defining for each point a measure (λ_i) of its potential quality as a C-point. There are several ways to make this initial assessment, but the simplest one is to count the number of other points strongly influenced by i (i.e. $\left|S_i^T\right|$). Once the measure has been made, the point with maximum λ_i is selected to become a C-variable. This point influences several of the other points and should appear in the interpolation formula for each of them. This implies that all variables j, which are strongly dependent on i and have not yet been decided (i.e. $j \in \left(U \cap S_i^T\right)$), become F-variables. This is permissible, because they already have a C-point that strongly influences them. Further more it is logical to look at other points that strongly influence these new F-points as potential C-points. Therefore, for each new F-point j in S_i^T the measure λ_k of each unassigned point k that strongly influences j is incremented by 1. Also points which have a strong influence on point i are not important anymore for point i to become a C-variable (since i is already a C-variable). Therefore the measure of these points is decreased by 1. Now again the point with maximum λ_i is selected to become a C-variables and the process is repeated until all points have been treated.

From this description the definition of λ_i becomes:

$$\lambda_i = \left|S_i^T \cap U\right| + 2\left|S_i^T \cap F\right| \tag{3.36}$$

Initially, variables which are important for many other variables become C-variables. Later, the tendency is to pick C-variables as those variables on which many F-variables depend. An outline of the algorithm is given in algorithm 1. In should be noted that the measure λ_i has to be computed globally only at the beginning of the algorithm. At later stages, it just needs to be updated locally. Therefore it is not necessary to loop over the entire list of unassigned points to search for the point with maximum λ_i to become the next C-variable. In the actual code, the points are stored in a linked list, allowing fast updating, by simply removing the points and inserting them again at the new location (see appendix B). Looping over all points to search for the point with maximum λ_i would have resulted in a coloring algorithm which requires $\mathcal{O}\left(n^2\right)$ operations (with n the number of grid points). Such a slow coloring algorithm would render the use of multi-grid useless.

Before the coloring algorithm starts, variables which have no connection at all (e.g. resulting from Dirichlet boundary points which have not been eliminated from the system) are filtered out and become F-variables. Such variables do not require interpolation.

After the algorithm is terminated all F-variables have (at least) one strong coupling to a C-variable (except for the ones taken out at the beginning). However, there may be a few U-variables left. Such variables do not depend strongly on any of the C-variables (otherwise they would have been assigned as F-variables earlier on). Moreover, no other variables strongly depend on these U-variables (otherwise their measure λ_i would be nonzero). However, each of these U-variables depends on at least one of the F-variables. Therefore all remaining U-variables are declared to become F-variables.

It is useful to observe that the coarsening determined by this method depends on several factors. Among the most influential ones is the order in which the grid points are scanned (and thus also the order in which they are stored) when seeking the next point with maximal λ_i. Because many, if not most, of the grid points will have the same value at the start, any of them could have been selected as the first coarse point. Once the first point is selected, the rest proceeds as described

Algorithm 1 Basic coloring algorithm

$C = \{\}$
$F = \{\}$
get $i \in U$ with λ_i maximal
while $\lambda_i > 0$ **do**
  $C = C \cup \{i\}$
  $U = U \setminus \{i\}$
  for $j \in \left(U \cap S_i^T\right)$ **do**
    $F = F \cup \{j\}$
    $U = U \setminus \{j\}$
    for $l \in (U \cap S_j)$ **do**
      $\lambda_l = \lambda_l + 1$
    end for
  end for
  for $l \in U \cap S_i$ **do**
    $\lambda_l = \lambda_l - 1$
  end for
  get $i \in U$ with λ_i maximal
end while

above. Again, any time there is more than one point with the maximal value, there are many allowable coarsenings, but for the resulting performance of the algorithm this is not crucial.

Chapter 4

Geometric multi-grid applied to the Poisson problem

In this chapter the performance of multi-grid algorithms for the Poisson problem is shown. The Poisson problem is a classical test case for comparison of the performance of numerical methods and serves well for demonstrating the basics of multi-level algorithms and the effect of different choices made in the algorithm. In two dimensions the Poisson problem is given by:

$$\begin{array}{rcll} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} & = & f(x,y) & (x,y) \in \Omega \\ u & = & 0 & (x,y) \in \partial\Omega \end{array} \tag{4.1}$$

Some analytic solutions of the Poisson problem are available and can be used as a comparison for performance evaluation. Full weighting, bi-linear interpolation and a simple coarse grid operator are used as the multi-grid components.

4.1 Discretization

The first step in numerical solutions is to replace the partial differential equations by a system of algebraic equations which approximates the partial differential equation in a finite number of discrete points or volumes. This process is referred to as discretization and can be done in various ways (e.g. finite difference, finite element and finite volume). This approximation of the PDE introduces an error with respect to the solution of the PDE. This error is called the discretization error and depends solely on the discretization. The discretization error for a specific problem can be studied when the exact solution is known, or can be estimated using Taylor series.

The finite difference method uses Taylor series of the unknown function to approximate the derivatives that appear in the partial differential equation. A second-order accurate approximation of the 2D Poisson equation in gridpoint (x_i, y_j) is:

$$\frac{u^h_{i-1,j} - 2u^h_{i,j} + u^h_{i+1,j}}{h_x^2} + \frac{u^h_{i,j-1} - 2u^h_{i,j} + u^h_{i,j+1}}{h_y^2} = f^h_{i,j} \tag{4.2}$$

with h_x and h_y the constant mesh-size in x- and y-direction, respectively, $u^h_{i,j}$ the discrete approximation of the unknown $u(x_i, y_j)$ to be solved for and $f^h_{i,j}$ the discrete representation of the continuous function $f(x_i, y_j)$. For efficiency of notation a stencil notation is often used. The operator acting on u^h, referred to as L^h, is then written as:

$$L^h \left\langle u^h \right\rangle_{i,j} = f^h_{i,j} \tag{4.3}$$

with :

$$L^h = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \tag{4.4}$$

where the gridsize is assumed to be equal in each direction ($h_x = h_y = h$). The coarse grid operator for this problem can be chosen as the same discretization of the equation on the coarse grid:

$$L^H = \frac{1}{H^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \tag{4.5}$$

As an alternative Galerkin coarsening can be used. When full weighting and bi-linear interpolation are chosen as the restriction and interpolation operators, the stencil of the coarse grid operator becomes:

$$L^H = \frac{1}{4H^2} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.6}$$

This operator is also a second-order accurate approximation to the Laplace operator. It gives about the same performance as the direct discretization, however, it has two disadvantages. Firstly, the coarse grid operator becomes different on each grid. For special cases it even needs to be calculated at each level (i.e. for problems with varying coefficients). Secondly, the Galerkin coarse grid operator corresponds to a 9 point stencil (instead of a 5-point one), thus increasing the time to evaluate the operator by a factor of 1.8. When more Galerkin coarsening steps are performed the coarse grid operator will converge towards the following 9-point differential operator:

$$L^H = \frac{1}{3H^2} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.7}$$

4.2 Performance

The performance of multi-grid cycles for the Poisson problem is compared with predictions obtained by LMA and two-grid convergence analysis.

4.2.1 Smoothing analysis

Standard one-point Gauß-Seidel or damped Jacobi relaxation have good smoothing properties for the Poisson problem. This follows from the LMA: (for a more general overview of LMA, see section 2.2.1)

Jacobi relaxation

For point-Jacobi relaxation the previous values are used for the approximation of all new values (i.e. $\bar{u}^h_{i,j} = \tilde{u}^h_{i,j}$) in equation (2.19). With the standard 5-point differential operator defined in equation (4.4) on a uniform grid ($h_x = h_y = h$), the relation between the error before and after relaxation can be expressed as:

$$\hat{v}^h_{i,j} = \tilde{v}^h_{i,j} + \omega \frac{\tilde{v}^h_{i-1,j} + \tilde{v}^h_{i+1,j} - 4\tilde{v}^h_{i,j} + \tilde{v}^h_{i,j-1} + \tilde{v}^h_{i,j+1}}{4} \tag{4.8}$$

Substitution of the Fourier series (equation 2.20 and 2.20) and considering only one Fourier component, yields:

$$\hat{A}\left(\vec{\theta}\right) = \tilde{A}\left(\vec{\theta}\right)\left((1-\omega) + \omega\frac{\left(e^{-i\theta_1} + e^{i\theta_1} + e^{-i\theta_2} + e^{i\theta_2}\right)}{4}\right) \tag{4.9}$$

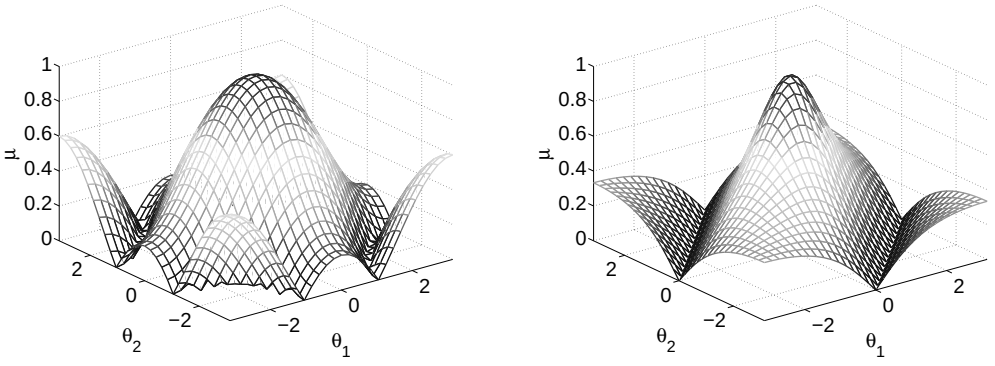
so the error amplification factor (equation 2.21) is:

$$\mu\left(\vec{\theta}\right) = \left|\frac{\hat{A}\left(\vec{\theta}\right)}{\tilde{A}\left(\vec{\theta}\right)}\right| = \left|(1-\omega) + \omega\frac{\left(e^{-i\theta_1} + e^{i\theta_1} + e^{-i\theta_2} + e^{i\theta_2}\right)}{4}\right| \tag{4.10}$$

This can be rewritten as:

$$\mu\left(\vec{\theta}\right) = \left|1 - \omega\left(\sin^2\left(\frac{\theta_1}{2}\right) + \sin^2\left(\frac{\theta_2}{2}\right)\right)\right| \tag{4.11}$$

Equation (4.11) is plotted in figure 4.1(a) for $\omega = 4/5$ (the optimal value for the smoothing factor [6,12]). The smoothing factor for multi-grid, defined in equation (2.23), is 0.6 for $\theta_1 = \pm\theta_2 = \pm\pi$. This implies that a (ν_1, ν_2) cycle has the potential of reducing the error by a factor of $0.6^{\nu_1+\nu_2}$. When no multi-grid techniques are used convergence can be approximated by equation (2.22) and is $1 - \mathcal{O}\left(\omega h^2\right)$.



(a) Jacobi $\omega = \frac{4}{5}$ (b) Gauß-Seidel

Figure 4.1: Poisson problem: LMA analysis for point Jacobi and point Gauß-Seidel relaxation

Gauß-Seidel relaxation

Gauß-Seidel relaxation can be analyzed in the same way. The only difference is that. instead of using only previous values, already updated values are used (i.e. $\bar{u}^h_{i,j} = \hat{u}^h_{i,j}$) in equation (2.19). With the standard 5-point differential operator defined in equation (4.4) on a uniform grid ($h_x = h_y = h$), the relation between the error before and after relaxation can be expressed as:

$$\hat{v}^h_{i,j} = \tilde{v}^h_{i,j} + \frac{\hat{v}^h_{i-1,j} + \hat{v}^h_{i+1,j} - 4\tilde{v}^h_{i,j} + \tilde{v}^h_{i,j-1} + \tilde{v}^h_{i,j+1}}{4} \tag{4.12}$$

Substitution of the Fourier series (equation 2.20 and 2.20) and assuming that each component is mapped onto itself, yields for the error amplification factor (equation 2.21):

$$\mu\left(\vec{\theta}\right) = \left|\frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}\right| \tag{4.13}$$

A graph of the absolute local error amplification factor is presented in figure 4.1(b). The smoothing factor as defined in equation (2.23) is 0.5 for e.g. $\theta_1 = \pi/2$ and $\theta_2 = \cos^{-1}(4/5)$. Here overrelaxation cannot provide better convergence. This implies that a (ν_1, ν_2) cycle has the potential of reducing the error by a factor of $0.5^{\nu_1+\nu_2}$. Hence a cycle with $\nu_1 = 2$ and $\nu_2 = 1$, can potentially reduce the error by an order of magnitude.

4.2.2 Two-grid analysis

In section 2.2.2 the two-grid analysis has been described, along with the Fourier symbols for interpolation and restriction. To complete the two-grid analysis also the Fourier symbol for the operator, the coarse grid operator and the smoothing are required. These symbols are discussed below.

Operator

First the Fourier symbol of the operator is calculated. This can be done by applying the operator (equation 4.4) to the grid functions (equation: 2.24)

$$\begin{aligned} L^h \left\langle \phi^h \left(\vec{\theta}, \vec{x} \right) \right\rangle &= e^{i\theta_1 \frac{x_1}{h}} e^{i\theta_2 \frac{x_2}{h}} \frac{1}{h^2} \left(-e^{i\theta_2} - e^{-i\theta_1} + 4 - e^{i\theta_1} - e^{-i\theta_2} \right) \\ &= \phi^h \left(\vec{\theta}, \vec{x} \right) \frac{2}{h^2} \left(2 - \cos\theta_1 - \cos\theta_2 \right) \end{aligned} \tag{4.14}$$

So the Fourier symbol is:

$$\tilde{L}^h \left(\vec{\theta}^\alpha \right) = \frac{2}{h^2} \left(2 - \cos\theta_1 - \cos\theta_2 \right) \tag{4.15}$$

And the transformation matrix:

$$\hat{L}^h \left(\vec{\theta} \right) = \begin{bmatrix} \tilde{L}^h \left(\vec{\theta}^{(0,0)} \right) & & & \\ & \tilde{L}^h \left(\vec{\theta}^{(1,1)} \right) & & \\ & & \tilde{L}^h \left(\vec{\theta}^{(1,0)} \right) & \\ & & & \tilde{L}^h \left(\vec{\theta}^{(0,1)} \right) \end{bmatrix} \tag{4.16}$$

Coarse grid operator

The Fourier symbol of the coarse grid operator is calculated by applying the coarse grid operator (equation 4.5) to the grid functions (equation: 2.24)

$$\begin{aligned} L^H \left\langle \phi^h \left(2\vec{\theta}, \vec{x} \right) \right\rangle &= e^{2i\theta_1 \frac{x_1}{H}} e^{2i\theta_2 \frac{x_2}{H}} \frac{1}{H^2} \left(-e^{2i\theta_2} - e^{-2i\theta_1} + 4 - e^{2i\theta_1} - e^{-2i\theta_2} \right) \\ &= \phi^H \left(2\vec{\theta}, \vec{x} \right) \frac{2}{H^2} \left(2 - \cos 2\theta_1 - \cos 2\theta_2 \right) \end{aligned} \tag{4.17}$$

giving:

$$\tilde{L}^H \left(2\vec{\theta} \right) = \frac{2}{H^2} \left(2 - \cos 2\theta_1 - \cos 2\theta_2 \right) \tag{4.18}$$

For the inverse of the operator the Fourier symbol is:

$$\tilde{L}^H \left(2\vec{\theta} \right)^{-1} = \frac{H^2}{2 \left(2 - \cos 2\theta_1 - \cos 2\theta_2 \right)} \tag{4.19}$$

Smoothing operator

For lexicographic Gauß-Seidel and Jacobi relaxation the Fourier symbol of the smoothing operator can be represented by the diagonal matrix.

$$\hat{S}^h \left(\vec{\theta} \right) = \begin{bmatrix} \mu \left(\vec{\theta}^{(0,0)} \right) & & & \\ & \mu \left(\vec{\theta}^{(1,1)} \right) & & \\ & & \mu \left(\vec{\theta}^{(1,0)} \right) & \\ & & & \mu \left(\vec{\theta}^{(0,1)} \right) \end{bmatrix} \tag{4.20}$$

with $\mu \left(\vec{\theta} \right)$ defined by 4.11 for Jacobi relaxation and by 4.13 for Gauß-Seidel relaxation.

Results

The predicted two-level convergence follows from the largest eigenvalues of the Fourier symbol of the two-grid iteration operator:

$$M_h^H = \left(\hat{S}^h\right)^{\nu_2} \left(\tilde{I}^h - I_H^h \left(\tilde{L}^H\right)^{-1} \tilde{I}_h^H \hat{L}^h\right) \left(\hat{S}^h\right)^{\nu_1} \tag{4.21}$$

Results are shown in table 4.1 and compared to the smoothing factors. For a small number of smoothing steps (ν) results of both analysis are quite similar. For larger ν the smoothing factors become too optimistic (especially for Gauß-Seidel relaxation), because the coarse grid approximation can no longer reduce low-frequency components as fast as the relaxation removes high-frequency components. Table 4.1 also compares the influence of the restriction operator (full weighting (FW) and injection (INJ) on the convergence factors. The results show that both types are satisfactory restriction operators for the Poisson equation. Injection requires less computational work than FW, so it appears to be the better restriction operator. However, the injection operator has the disadvantage that the spectral norm (i.e. the square root of the largest eigenvalue of MM^T, which gives insight into the error reduction in one iteration step) of the corresponding two-grid operators is not bounded. One thus needs to be careful with this operator, in particular if it is used in FMG or more generally, if only a small number of cycles is used.

	Jacobi			Gauß-Seidel-LEX		
$\nu_1+\nu_2$	$\mu_{loc}^{\nu_1+\nu_2}$	ρ_{loc}FW	ρ_{loc}INJ	$\mu_{loc}^{\nu_1+\nu_2}$	ρ_{loc}FW	ρ_{loc}INJ
1	0.600	0.600	0.600	0.500	0.400	0.447
2	0.360	0.360	0.360	0.250	0.193	0.200
3	0.216	0.216	0.216	0.125	0.119	0.089
4	0.130	0.137	0.130	0.063	0.083	0.042
5	0.078	0.133	0.078	0.031	0.064	0.028

Table 4.1: Poisson problem: Smoothing factors and two-grid convergence factors for point Jacobi and Point Gauß-Seidel relaxation

4.3 Results

In this paragraph results for the two dimensional Poisson equation will be shown. First results for the single-grid relaxation sweeps are presentend. Next the relaxation is used in combination with the Multi Level scheme. Finally results of the Full Multi Grid scheme are shown. Al results are obtained using a right-hand side, for which the analytical solution is known. This allows performance to be evaluated not only by means of residual reduction but also by the reduction of the error in the solution itself.

$$\begin{aligned} f(x,y) &= -8\pi^2 \sin(2\pi x)\sin(2\pi y) \\ u(x,y) &= \sin(2\pi x)\sin(2\pi y) \end{aligned} \tag{4.22}$$

Residuals and errors will be measured using the L_2 or Euclidean norm, defined as:

$$\left|u^h\right|_2 = \sqrt{\frac{1}{N_y N_x}} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \left(u_{i,j}^h u_{i,j}^h\right) \tag{4.23}$$

4.3.1 Relaxation

For the case of single-grid Gauß-Seidel relaxation sweeps on a uniform grid with mesh-size h in both directions the asymptotic error reduction is $1 - \mathcal{O}\left(h^2\right)$ as predicted by the LMA in paragraph 4.2.1. This is illustrated in figure 4.2, where the L_2 norm of the residual ($\left|r^h\right|_2$) and of

the error ($\left|u^h - u^*\right|_2$) are plotted as a function of the number of sweeps for different mesh-sizes. Actually the curves show three different types of behaviors. In the first part the convergence rate is high. Initially all frequencies are present in the residual. Due to the nature of the Gauß-Seidel relaxation the high-frequency components are reduced very effectively, leading to a good initial convergence rate. After a few sweeps, however, the convergence rate reduces to the asymptotic convergence speed, because only the troubling frequency components remain. After many sweeps the residual becomes so small that the machine accuracy limits further reduction of the residual. The error mimics the behavior of the residual. However, it converges to a value which represents the accuracy of discretization scheme and decreases to zero only with decreasing mesh-size. The chosen discretization scheme is $\mathcal{O}\left(h^2\right)$ accurate (see section 4.1) so the lower limit of the error reduces quadratically as the number of grid points increases, which is indeed observed in figure 4.2(b).

4.3.2 Multi-Level

For Multi-Level cycles using Gauß-Seidel relaxation the expected error reduction per cycle is 0.12 for V(2,1) multi-grid cycles. This is illustrated in figure 4.3, where the L_2 norm of the residual and the error are plotted as a function of the number of cycles for different mesh-sizes. In the graphs a constant (independent of the mesh-size) reduction of the residual and the error can be seen up to machine accuracy for the residual and discretization accuracy for the error. In table 4.2 the convergence rates for V- and W-cycles on a mesh-size $h = 1/256$ are compared with the smoothing and two-grid convergence factors for different numbers of relaxation. Good agreement is achieved for small ν. For higher ν the smoothing factors for V-cycles tend to be too optimistic, whereas two-grid convergence factors are still accurate. Also for high ν, convergence for W-cycles is slightly better than for V-cycles, but the increased computational cost of W-cycles (see section 2.3.2) makes V-cycles more efficient overall. Whether post- or pre-relaxation are used turned out to have little effect as long as the total number of relaxation sweeps used on each level remains constant. This is consistent with the results of the two-grid analysis.

	Jacobi				Gauß-Seidel-LEX			
$\nu_1 + \nu_2$	$\mu_{loc}^{\nu_1+\nu_2}$	ρ_{loc}	V-cycle	W-cycle	$\mu_{loc}^{\nu_1+\nu_2}$	ρ_{loc}	V-cycle	W-cycle
1	0.600	0.600	0.593	0.593	0.500	0.400	0.390	0.390
2	0.360	0.360	0.361	0.356	0.250	0.193	0.186	0.191
3	0.216	0.216	0.240	0.213	0.125	0.119	0.127	0.117
4	0.130	0.137	0.184	0.134	0.063	0.083	0.096	0.083

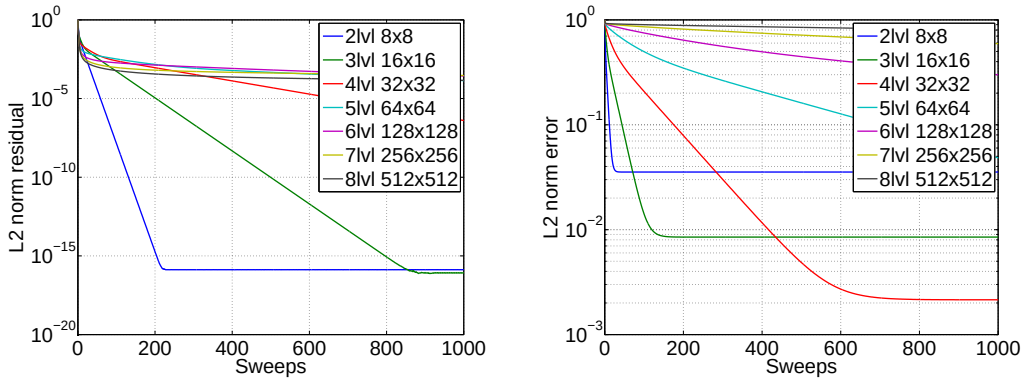
Table 4.2: Poisson problem: Computed theoretical smoothing factors and two-grid convergence factors, compared to V- and W-cycle performance on a target 256x256 grid using point Gauß-Seidel relaxation

4.3.3 Full-Multi-Grid

FMG results for the Poisson equation are shown in figure 4.4. Here the L_2-norm of the residual and the error are plotted as a function of the number of grid points for 1 and 2 V(2,1) cycles and for a W(2,1) cycle per level. As the difference in error between the test cases is small, it becomes clear that one FMG V-cycle is sufficient to obtain solutions with a numerical error that is small compared to the discretization error.

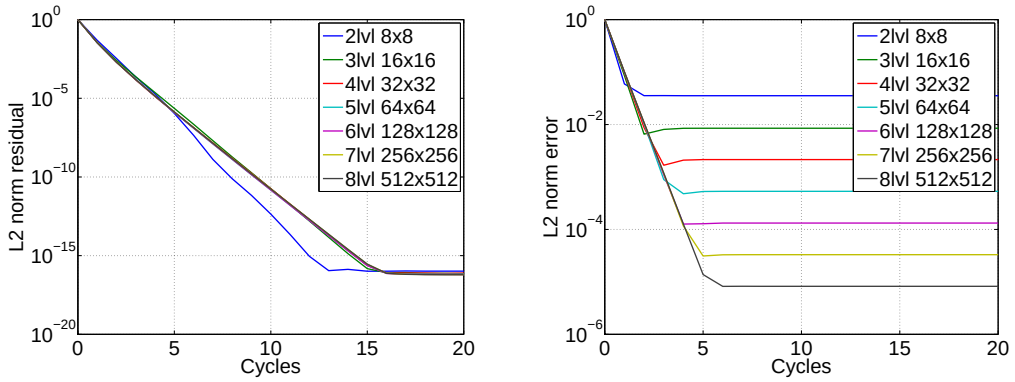
4.3.4 τ-Extrapolation

An easy way to obtain higher-order accurate results facilitated by multi-grid is τ-extrapolation. This technique has been applied in combination with FMG for the Poisson equation. However, for τ-extrapolation it is not useful to analyze the residual, because with τ-extrapolation the cycle corrects to an accurate solution of the PDE and not to an accurate solution of the discretized



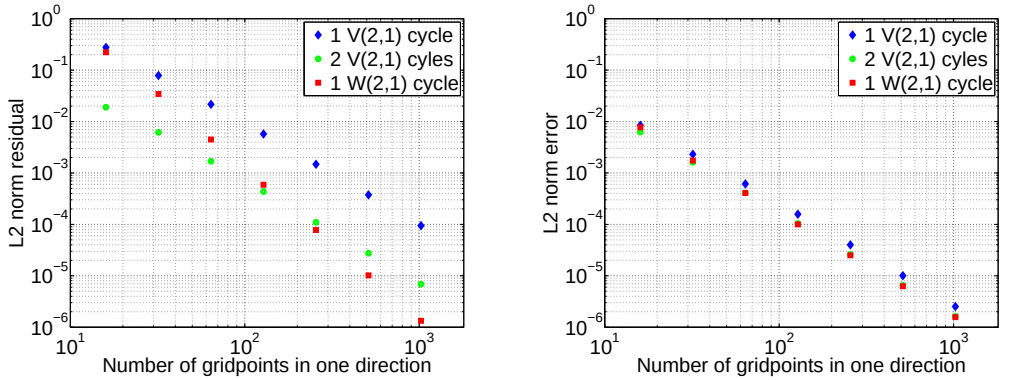
(a) L_2-norm of the residual (b) L_2-norm of the error

Figure 4.2: Poisson problem: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of single grid relaxation sweeps for different grid sizes using point Gauß-Seidel relaxation (the mesh-size for a grid identified by n x n is $h = 1/n$)



(a) L_2-norm of the residual (b) L_2-norm of the error

Figure 4.3: Poisson problem: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of V(2,1) multi-grid cycles for different grid sizes using point Gauß-Seidel relaxation



(a) L_2-norm of the residual (b) L_2-norm of the error

Figure 4.4: Poisson problem: L_2-norm of the residual ($\left|r^h\right|_2$) and error ($\left|u^h - u^*\right|_2$) on different grids for the FMG cycle using point Gauß-Seidel relaxation

equations. Therefore only the errors are plotted in figure 4.5 as a function of the number of grid points for FMG with 1 and 2 V(2,1) cycles and for a W(2,1) cycle per level. From this figure two observations can be made. Firstly, the τ-extrapolation makes the discretization more accurate (compare with figure 4.4(b)). In the present case the τ-extrapolation produces a fourth-order accurate solution (see setion 2.5.1). Halving the mesh-size will decrease the error by a factor of 16. Secondly, either two V(2,1) cycles or one W(2,1)-cycle per mesh are required to reach the desired accuracy. This is what is expected. Assume the initial approximation on a level to have an error small compared to the discretization error of the coarser grid ($O(H^4)$ with $H = 2h$). To obtain an error which is small compared to the discretization error of the current grid ($O(h^4)$), the error needs to be reduced by a factor of $H^4/h^4 = 2^4 = 16$. One cycle reduces the error by a factor of 8, hence 2 cycles are required.

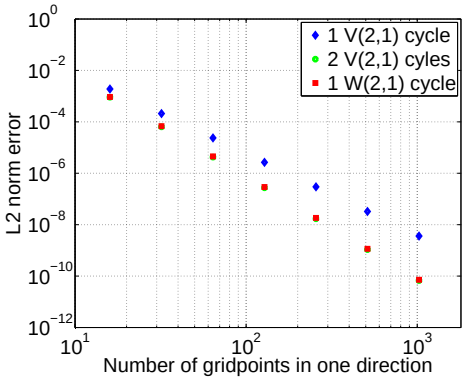


Figure 4.5: Poisson problem: L_2-norm of the error ($\left|u^h - u\right|_2$) on different grids for the FMG cycle using τ-extrapolation and point Gauß-Seidel relaxation

4.4 Anisotropic Poisson equation

The preceding section showed that the isotropic Poisson equation can be solved very efficiently using standard multi-grid techniques. Often this efficiency does not extend automatically to other problems even when they appear very similar. Again a detailed analysis is required to ensure satisfactory performance. For example, consider the anisotropic Poisson equation on a uniform mesh (with mesh-size $h_x = h_y = h$):

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial}{\partial y}\left(\epsilon\left(x,y\right)\frac{\partial u}{\partial y}\right) &= f\left(x,y\right) && (x,y) \in \Omega \\ u &= 0 && (x,y) \in \partial\Omega \end{aligned} \tag{4.24}$$

This problem can be discretized using central differences, yielding the standard second-order stencil for the x-direction. For the y-direction the stencil becomes:

$$\begin{aligned} \left.\frac{\partial}{\partial y}\left(\epsilon\left(x,y\right)\frac{\partial u}{\partial y}\right)\right|_{i,j} &= \frac{-\epsilon_{i,j-\frac{1}{2}}\left.\frac{\partial u}{\partial y}\right|_{i,j-\frac{1}{2}} + \epsilon_{i,j+\frac{1}{2}}\left.\frac{\partial u}{\partial y}\right|_{i,j+\frac{1}{2}}}{h} \\ \epsilon_{i,j-\frac{1}{2}}\left.\frac{\partial u}{\partial y}\right|_{i,j-\frac{1}{2}} &= \epsilon_{i,j-\frac{1}{2}}\frac{-u^h_{i,j-1} + u^h_{i,j}}{h} \\ \epsilon_{i,j+\frac{1}{2}}\left.\frac{\partial u}{\partial y}\right|_{i,j+\frac{1}{2}} &= \epsilon_{i,j+\frac{1}{2}}\frac{-u^h_{i,j} + u^h_{i,j+1}}{h} \\ \left.\frac{\partial}{\partial y}\left(\epsilon\left(x,y\right)\frac{\partial u}{\partial y}\right)\right|_{i,j} &= \frac{\epsilon_{i,j-\frac{1}{2}}u^h_{i,j-1} - \left(\epsilon_{i,j-\frac{1}{2}} + \epsilon_{i,j+\frac{1}{2}}\right)u^h_{i,j} + \epsilon_{i,j+\frac{1}{2}}u^h_{i,j+1}}{h^2} \end{aligned} \tag{4.25}$$

This leads to the following 5-point stencil:

$$L^h_{i,j} = \frac{1}{h^2}\begin{bmatrix} & \epsilon_{i,j+\frac{1}{2}} & \\ 1 & -2-\left(\epsilon_{i,j-\frac{1}{2}}+\epsilon_{i,j+\frac{1}{2}}\right) & 1 \\ & \epsilon_{i,j-\frac{1}{2}} & \end{bmatrix} \tag{4.26}$$

For simplicity it is assumed that ϵ is an analytic function and its values can be calculated in each point. When ϵ is only known in grid points the value in the "half" points can be obtained by using the average of its two neighbors:

$$L^h_{i,j} = \frac{1}{h^2}\begin{bmatrix} & \frac{1}{2}\left(\epsilon^h_{i,j+1}+\epsilon^h_{i,j}\right) & \\ 1 & -2-\frac{1}{2}\left(\epsilon^h_{i,j-1}+2\epsilon^h_{i,j}+\epsilon^h_{i,j+1}\right) & 1 \\ & \frac{1}{2}\left(\epsilon^h_{i,j}+\epsilon^h_{i,j-1}\right) & \end{bmatrix} \tag{4.27}$$

For the case of constant ϵ one obtains:

$$L^h_{i,j} = \frac{1}{h^2}\begin{bmatrix} & \epsilon & \\ 1 & -2-2\epsilon & 1 \\ & \epsilon & \end{bmatrix} \tag{4.28}$$

which is the equivalent of the isotropic Poisson equation, discretized on a non uniform mesh with mesh-sizes $h_x = h$ in x-direction and $h_y = \frac{h}{\sqrt{\epsilon}}$ in y-direction.

As a result of the anisotropy there is a weak connection in the y-direction for $\epsilon \ll 1$. This small difference from the conventional Poisson equation already leads to trouble when the standard multi-grid approach would be applied indiscriminately: multi-grid convergence factors degenerate as ϵ tends to zero or to infinity. This can be seen for example by a LMA analysis using Gauß-Seidel iterations for constant ϵ.

$$\begin{aligned} \hat{v}_{i,j} &= \tfrac{\hat{v}_{i-1,j}+\tilde{v}_{i+1,j}+\epsilon(\hat{v}_{i,j-1}+\tilde{v}_{i,j+1})-h^2 f_{i,j}}{2+2\epsilon} \\ \mu\left(\vec{\theta},\epsilon\right) &= \left|\tfrac{e^{\theta_1 i}+\epsilon e^{\theta_2 i}}{e^{-\theta_1 i}+\epsilon e^{-\theta_2 i}-2-2\epsilon}\right| \\ \mu_{loc} &= \max_{\vec{\theta}\in T^{high}} \mu\left(\vec{\theta},\epsilon\right) = 1 \end{aligned} \tag{4.29}$$

The asymptotic error reduction for small ϵ deteriorates, as can be seen in equation 4.29 and in figure 4.6. For example for $\epsilon = 0$ and $\theta_1 = 0$ the convergence rate becomes $G(0,\theta_2,0) = 1$. This means that components, which are oscillatory in the y-direction and smooth in the x-direction, are not affected by the relaxation at all. In figure 4.6 this effect can be seen from the values along the line $\theta_1 = 0$ for small ϵ.

4.4.1 Semi-coarsening

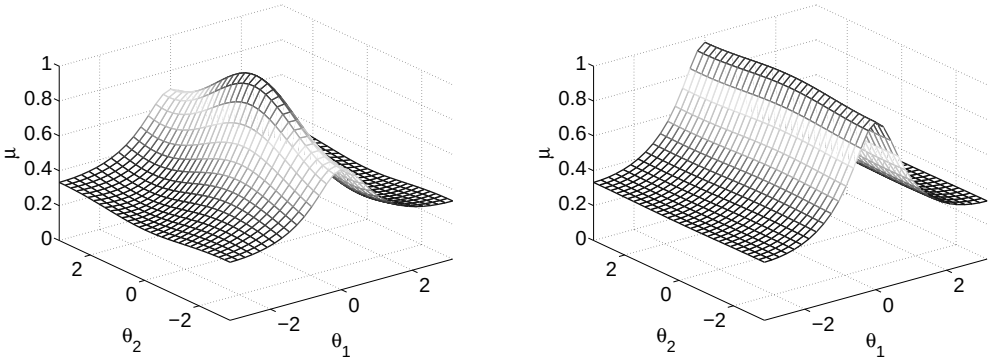
A solution to this problem is to realize that the error after a few relaxations is only smooth in the x-direction. Such errors can be well approximated on a grid that is coarser in the direction of strong coupling only, i.e. a semi-coarsened grid.

To predict the convergence of a cycle with semi-coarsening LMA can still be used, but the range of high frequencies, that the fine grid should reduce, is now changed to either $|\theta_1| \geq \pi/2$ or $|\theta_2| \geq \pi/2$, depending on the direction of coarsening. Smoothing factors for semi-coarsening for the anisotropic Poisson problem vary from $\mu_{loc} = 0.5$ for $\epsilon = 1$ to $\mu_{loc} = \sqrt{5}/5$ for $\epsilon = 0$.

For semi-coarsening with small ϵ the coarse grid operator becomes (by directly discretizing the continuous function on the coarse grid):

$$L^H_{i,j} = \frac{1}{H^2}\begin{bmatrix} & 4\epsilon & \\ 1 & -2-8\epsilon & 1 \\ & 4\epsilon & \end{bmatrix} \tag{4.30}$$

Compared to the fine grid operator (4.28), the anisotropy has decreased. When the semi-coarsening process is continued, the anisotropy will decrease further. When a grid is reached for which the isotropy is restored, full coarsening can be applied to solve the coarse grid problem.



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.01$

Figure 4.6: Anisotropic Poisson problem: LMA analysis for point Gauß-Seidel relaxation

4.4.2 X-Line relaxation

A second possibility to overcome the problems associated with anisotropy is to use line relaxation. Line relaxation implies that in the relaxations, instead of applying single point changes, systems of equations for an entire line of unknowns in the strongly coupled direction are solved simultaneously. The effect in the LMA can be determined easily. The errors before ($\tilde{v}$) and after relaxation ($\hat{v}$) will satisfy:

$$\hat{v}_{i-1,j} - (2 + 2\epsilon)\,\hat{v}_{i,j} + \hat{v}_{i+1,j} = -\epsilon\,(\hat{v}_{i,j-1} + \tilde{v}_{i,j+1}) + h^2 f_{i,j} \tag{4.31}$$

So that:

$$\begin{aligned} \mu\left(\vec{\theta}, \epsilon\right) &= \left| \frac{\epsilon e^{\theta_2 I}}{2\cos(\theta_1) + \epsilon e^{-\theta_2 I} - 2 - 2\epsilon} \right| \\ \mu_{loc} = \max_{\vec{\theta} \in T^{high}} \mu\left(\vec{\theta}, \epsilon\right) &= \begin{cases} \frac{\sqrt{5}}{5} \approx 0.45, & \text{if } \epsilon \leq \frac{2\sqrt{5}}{5-\sqrt{5}} \approx 1.62 \\ \frac{\epsilon}{2+\epsilon}, & \text{if } \epsilon \geq \frac{2\sqrt{5}}{5-\sqrt{5}} \approx 1.62 \end{cases} \end{aligned} \tag{4.32}$$

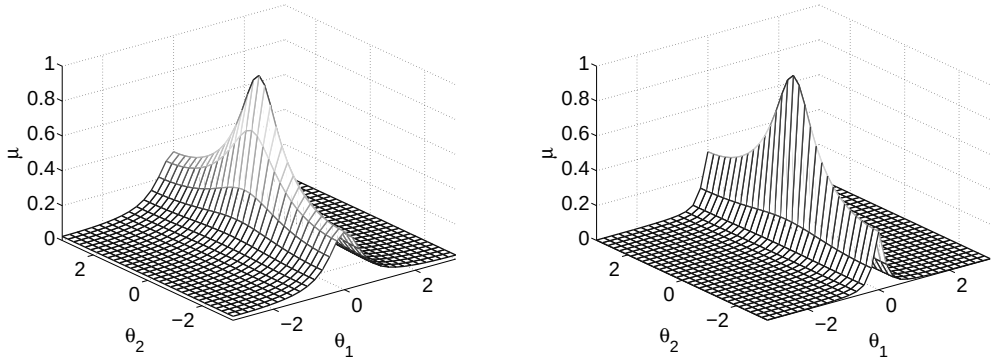
ϵ	0.001	0.01	0.1	1	10	100	100
μ	0.447	0.447	0.447	0.447	0.833	0.980	0.998

Table 4.3: Anisotropic Poisson problem: asymptotic convergence rates for line Gauß-Seidel relaxation for different values of ϵ

One sweep of the line Gauß-Seidel relaxation method for the Poisson equation, means solving a tri-diagonal system for each line of constant y. This can be performed efficiently with Gaussian elimination (Thomas algorithm) or a 1D multi-level solver. In particular a multi-level solver is interesting, as there is no need to solve the system exactly. A single V- or W- cycle is sufficient to obtain the full line relaxation efficiency. Both approaches require the same computational time of $\mathcal{O}(n)$ operations for one sweep. The work required for line relaxation is only a slightly larger than for point relaxations, so the cost of the whole cycle does not change drastically. The LMA analysis predicts good convergence rates for $\epsilon \leq 1$ for x-line relation (equation 4.32, table 4.3 and figure 4.7).

To illustrate the above method consider the numerical results obtained using a random initial condition and a right hand side corresponding to a known analytical solution.

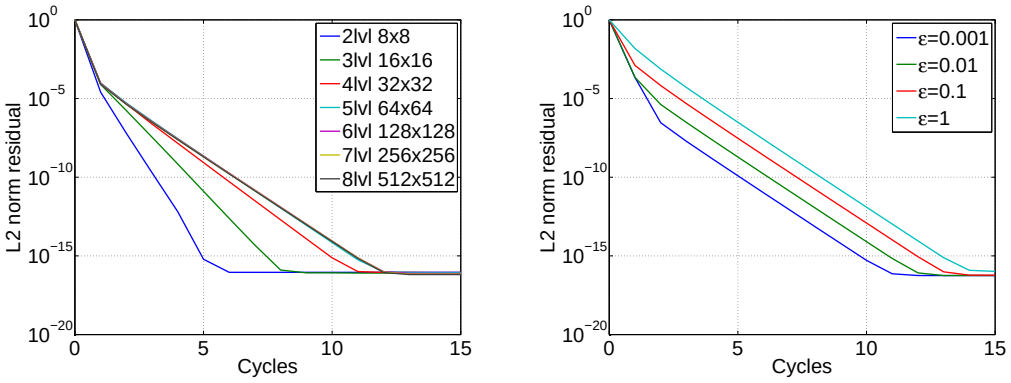
$$\begin{aligned} f(x,y) &= -4\,(1+\epsilon)\,\pi^2 sin(2\pi x)\, sin(2\pi y) \\ u(x,y) &= sin(2\pi x)\, sin(2\pi y) \end{aligned} \tag{4.33}$$



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.01$

Figure 4.7: Anisotropic Poisson problem: LMA analysis for line Gauß-Seidel relaxation

In figure 4.8(a) residuals are plotted for multi-level cycles with $\epsilon = 0.01$ on different mesh-sizes. The grid-independent convergence rates expected from the LMA are easily obtained for all grids. In figure 4.8(b) residuals are plotted for multi-level cycles on a 1024x1024 grid for different values of ϵ. Good convergence rates are obtained for all values of $\epsilon \leq 1$.



(a) Different grids (b) Different ϵ

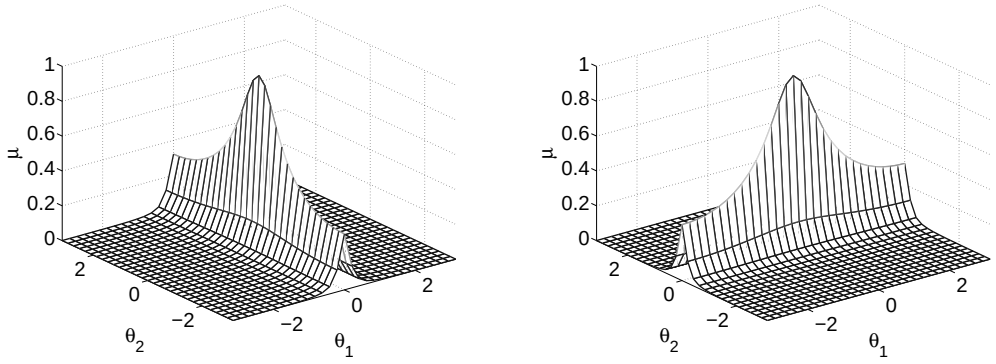
Figure 4.8: Anisotropic Poisson problem: L_2-norm of the residual ($|r^h|_2$) as a function of the number of V(2,1) multi-grid cycles on different grids (with constant $\epsilon = 0.01$) and for different values of ϵ (on a constant 1024x1024 grid) using line Gauß-Seidel relaxation

4.4.3 Alternating line relaxation

The single-line relaxation only performs well when the direction of weak coupling is known. However, often this direction is unknown or changing within the domain (i.e. Poisson's equation on a stretched grid with stretching towards the boundaries). To solve this problem one could use line relaxation in both directions. This process is known as alternating line relaxation. This makes the solver more robust and applicable to a larger range of Poisson-like problems, but it also doubles the computational cost. Alternating line relaxations works well, because line relaxation in the wrong direction does not adversely affect convergence. The LMA analysis is easily obtained by

multiplying the asymptotic convergence rate of x-line and y-line smoothers:

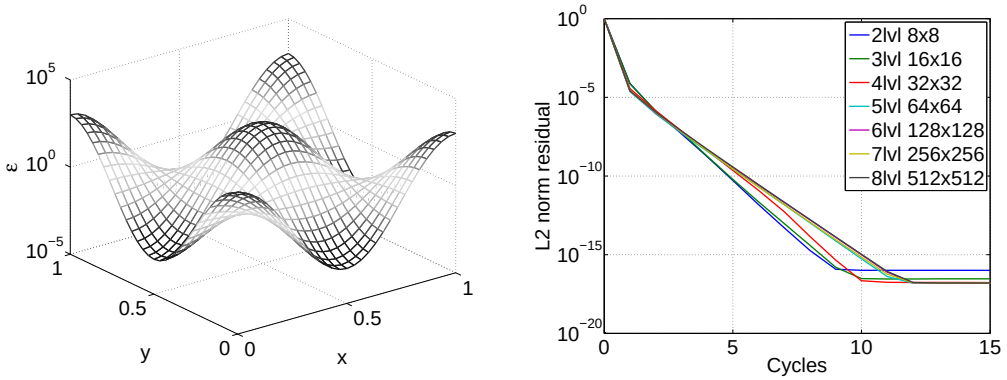
$$\mu\left(\vec{\theta},\epsilon\right) = \left|\frac{\epsilon e^{\theta_2 i}}{2\cos\left(\theta_1\right)+\epsilon e^{-\theta_2 i}-2-2\epsilon}\right|\left|\frac{e^{\theta_1 i}}{2\epsilon\cos\left(\theta_2\right)+e^{-\theta_1 i}-2-2\epsilon}\right|$$
$$\mu_{loc} = \max_{\vec{\theta}\in T_{high}} \mu\left(\vec{\theta},\epsilon\right) = \begin{cases} \frac{\sqrt{5}}{5}\frac{1}{1+2\epsilon}, & \text{if } \epsilon \leq 1 \\ \frac{\sqrt{5}}{5}\frac{\epsilon}{2+\epsilon}, & \text{if } \epsilon \geq 1 \end{cases} \tag{4.34}$$



(a) $\epsilon = 0.01$ (b) $\epsilon = 100$

Figure 4.9: Anisotropic Poisson problem: LMA analysis for alternating line Gauß-Seidel relaxation

The effect of alternating line relaxation on the convergence of a cycle is presented in figure 4.9, showing a graph of $\mu\left(\vec{\theta},\epsilon\right)$ for $\epsilon = 0.01$ and 100



(a) $\epsilon(x,y)$

(b) L_2-norm of the residual ($|r^h|_2$) after several V(2,1) multi-grid cycles on different grids using alternating line Gauß-Seidel relaxation

Figure 4.10: Anisotropic Poisson problem with varying $\epsilon(x,y)$

To illustrate the performance of alternating line relaxation in a cycle, numerical results have been obtained using a random initial condition and varying ϵ. A right-hand side corresponding to a known analytical solution is prescribed (see equation 4.35 and figure 4.10(a)).

$$\begin{aligned} \epsilon(x,y) &= 10^{3cos(2\pi x)cos(2\pi y)} \\ f(x,y) &= -4\pi^2 u(x,y)\left(1+\epsilon(x,y)\left(1+3cos(2\pi x)\,cos(2\pi y)\ln(10)\right)\right) \\ u(x,y) &= sin(2\pi x)\,sin(2\pi y) \end{aligned} \tag{4.35}$$

In figure 4.10(b) residuals are plotted for multi-level cycles on different mesh-sizes. The grid-independent convergence rates expected from the LMA are obtained for all grids. Results are also comparable to those of the single-line smoother in figure 4.8(a) for constant $\epsilon < 1$, but the computational cost is twice that of a single line relaxation. So when the direction of weak coupling is known it is faster to use the one-directional line relaxation.

4.5 Boundary conditions

So far Dirichlet boundary conditions have been imposed at the boundary of the domain. In practice, however, often boundary conditions like Neumann or periodic boundary conditions want to be imposed. In principle the convergence rate of a multi-grid solver is determined by the equations applied in the interior of the domain. But some boundary conditions may lead to locally non smooth errors near the boundary. In that case special measures are necessary to preserve the efficiency of the algorithm.

4.5.1 General treatment of boundary conditions

Consider the general 2D problem:

$$\begin{aligned} L_\Omega u &= f_\Omega(\Omega) \\ L_\Gamma u &= f_\Gamma(\Gamma = \partial\Omega) \end{aligned} \tag{4.36}$$

with a discretization

$$\begin{aligned} L_\Omega^h u &= f_\Omega^h\left(\Omega^h\right) \\ L_\Gamma^h u &= f_\Gamma^h\left(\Gamma^h = \partial\Omega^h\right) \end{aligned} \tag{4.37}$$

The general approach of treating boundary conditions can be summarized as follows:

- Relax both the equations in the interior of the domain and the equation on the boundary on the fine grid. This can be done one at a time, or by grouping some equations together and solving them.
- Transfer the residual from the fine grid to the coarse grids separately for boundary conditions and interior equations. This separation is important since the discretized boundary conditions and the discretized interior equations have different operators, typically with different powers of h.
- Relax both the coarse grid representation of the boundary conditions and the interior equations
- Interpolate the coarse grid correction to the fine grid including its boundary points.

Two examples of this approach for the case of the Poisson problem are explained below.

4.5.2 Neumann boundary conditions

Consider the Poisson equation on the unit square, with a Neumann boundary condition at one of the sides, for instance at the $x = 0$ boundary, and Dirichlet boundary conditions otherwise:

$$\begin{aligned} L_\Omega u = -\Delta u = f_\Omega \qquad & \Omega = (0,1)^2 \\ L_{\Gamma_N} u = u_n = f_{\Gamma_N} \qquad & \Gamma_N = \{(x,y) : x = 0, 0 < y < 1\} \\ L_{\Gamma_D} u = u = f_{\Gamma_D} \qquad & \Gamma_D = \partial\Omega \backslash \Gamma_N \end{aligned} \tag{4.38}$$

where u_n is the derivative of u in the direction of the outward normal. For the discretization of the Neumann boundary condition a central second-order approximation is used:

$$L^h_{\Gamma_N} u^h_{0,j} = \frac{1}{2h}\left(u^h_{-1,j} - u^h_{1,j}\right) \tag{4.39}$$

which is implemented assuming an extended grid with ghost points outside Ω. Because new points are introduced with this discretization, one also needs to add extra equations. It is assumed that the Laplace operator will hold on Γ_N too:

$$\begin{aligned} L^h_\Omega u^h &= f^h_\Omega \qquad \left(\Omega^h \cup \Gamma^h_N\right) \\ L^h_{\Gamma_N} u^h &= f^h_{\Gamma_N} \qquad \left(\Gamma^h_N\right) \\ u^h &= f^h_{\Gamma_D} \qquad \left(\Gamma_D\right) \end{aligned} \tag{4.40}$$

So the discretized Poisson equation is used to provide new approximations for the boundary points, whereas the discretized Neumann boundary condition is used to update the unknowns at the ghost points. In principle, these relaxations can be done one after the other or simultaneously. The latter means that a 2x2 system is solved per pair of grid points. The residuals of the boundary condition and the interior equations are restricted separately. The 1D full weighting operator (2.3) can be used to restrict the boundary conditions. For the restriction of the discretized interior equation at the Neumann boundary points also the residual at the ghost points is required. Equation 4.39 suggests using the residual at the interior points, leading to the so-called modified full weighting operator:

$$I^h_H = \frac{1}{16}\begin{bmatrix} 2 & 2 \\ 4 & 4 \\ 2 & 2 \end{bmatrix} \tag{4.41}$$

It is also possible to eliminate the unknowns at the ghost points. This can be done by substituting the equation for the Neumann boundary points in the discretized Poisson equation at the boundary points. This leads to the following equations for the Neumann boundary:

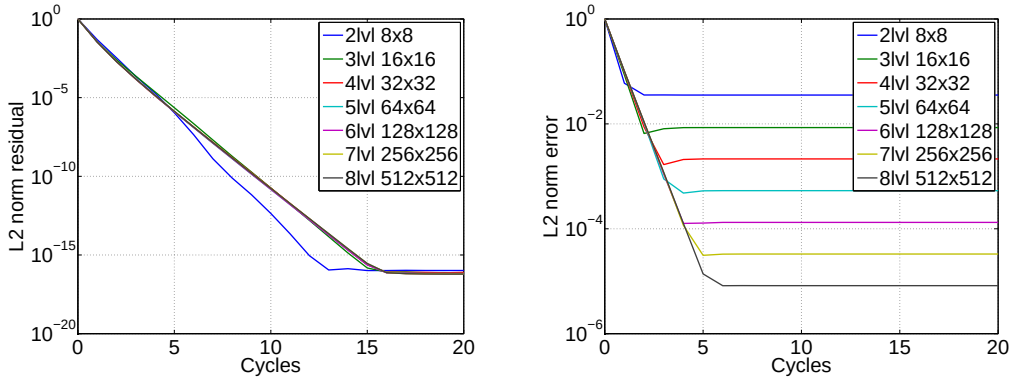
$$\begin{aligned} f^h_{\Gamma_N} &= f^h_\Omega + \frac{2}{h} f^h_{\Gamma_N} \\ L^h &= \frac{1}{h^2}\begin{bmatrix} -1 & \\ 4 & -2 \\ -1 & \end{bmatrix} \end{aligned} \tag{4.42}$$

The relaxation of the eliminated boundary conditions is equivalent to the collective relaxation of both equations located on the boundary in the non-eliminated case described above. The use of the modified full weighting operator (equation 4.41) for the eliminated Neumann boundary conditions is equivalent to the non-eliminated approach (where 1D full weighting is used for the discretized Neumann boundary condition and the modified full weighting for the discretized Poisson equation at boundary points).

To illustrate the method, multi-grid cycles are performed with a right hand side, corrseponding to a known analytical solution:

$$\begin{aligned} f_\Omega(x,y) &= -8\pi^2 \sin(2\pi x)\sin(2\pi y) \\ f_{\Gamma_N}(x,y) &= 2\pi \sin(2\pi y) \\ f_{\Gamma_D}(x,y) &= 0 \\ u(x,y) &= \sin(2\pi x)\sin(2\pi y) \end{aligned} \tag{4.43}$$

Performance is illustrated in figure 4.11, where the L_2 norm of the residual and the error are plotted as a function of the number of V(2,1) cycles for different grid sizes for a Poisson problem with a Neumann boundary condition. Convergence is just as good as for the Poisson problem with only Dirichlet boundary conditions (see figure 4.3).



(a) L_2-norm of the residual

(b) L_2-norm of the error

Figure 4.11: Poisson problem with Neumann boundary: L_2-norm of the residual ($\left|r^h\right|_2$) and error ($\left|u^h - u^*\right|_2$) as a function of the number of V(2,1) multi-grid cycles, for different grid sizes, using point Gauß-Seidel relaxation

4.5.3 Mixed boundary conditions

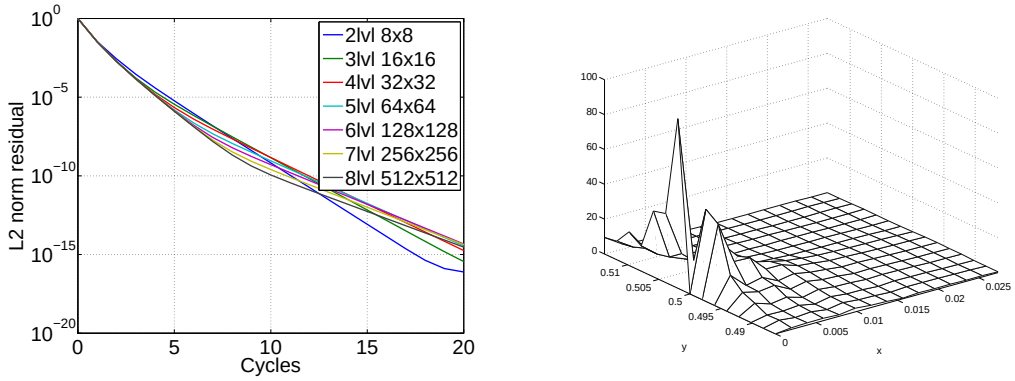
Consider the Poisson equation on the unit square, with a Neumann boundary condition imposed on half of one of the sides and Dirichlet boundary conditions on the other boundaries:

$$\begin{aligned} L_\Omega u = -\Delta u = f_\Omega \qquad & \Omega = (0,1)^2 \\ L_{\Gamma_N} u = u_n = f_{\Gamma_N} \qquad & \Gamma_N = \{(x,y) : x = 0, 0 < y < 0.5\} \\ L_{\Gamma_D} u = u = f_{\Gamma_D} \qquad & \Gamma_D = \partial\Omega \setminus \Gamma_N \end{aligned} \tag{4.44}$$

When the resulting the system is solved in the same way as equation 4.38, some convergence difficulties exist. These difficulties are shown in 4.12(a), where the L_2-norm of the residual is shown for V(2,1) cycles on different grids. Initial convergence is good, but after a few cycles convergence deteriorates and even becomes grid-size dependent. The cause of this deteriorated convergence is shown in figure 4.12(b), where the residual near the discontinuity in the boundary condition is plotted after six V(2,1) cycles on a 512x512 grid. Residuals are low everywhere except around the discontinuity in the boundary condition. Additional work in the neighborhood of the discontinuity is required for optimal performance. The extra work required for these relaxations is relatively small compared tot the total work, so the total required computational work still is $\mathcal{O}(N)$.

First additional point relaxations are performed on lines parallel to the y-axis. The number of lines and relaxations per point have been varied to study their optimal values. Results are shown in figure 4.13(a) for different cases. Two additional relaxations on four lines (including the boundary line) are required to restore performance (blue line), additional work does not further increase convergence.

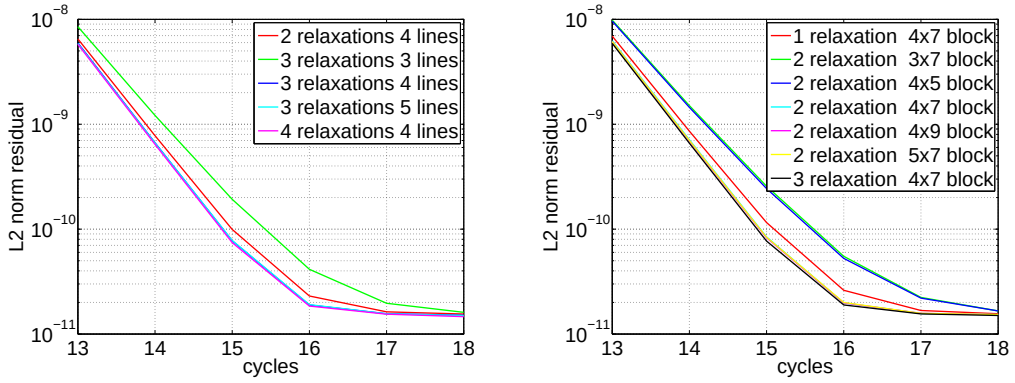
As an alternative, additional point relaxations can be performed on points in a square box around the discontinuity in the boundary condition. The length, width and number of additional relaxations have been varied to study optimal performance. Results are shown in figure 4.13(b) for different cases. Two additional relaxations for the points in a box with depth of four points (including the boundary line) and a width of seven points are required to preserve the performance (cyan line). Additional work does not further increase the convergence. The performance proved to be independent of the starting corner for the additional relaxations. The second method is computationally less expensive than the first method (i.e. 56 versus 4096 additional point relaxations for a 512x512 grid), but has the same performance. Therefore this method is optimal for this problem.



(a) L_2-norm of the residual ($\left|r^h\right|_2$) after several V(2,1) multi-grid cycles, for different grids

(b) Residual (r^h) after six V(2,1) multi-grid cycles on a 512x512 grid

Figure 4.12: Poisson problem with a discontinuity in the boundary condition



(a) Extra lines

(b) Extra block

Figure 4.13: Poisson problem with a discontinuity in the boundary condition: L_2-norm of the residual ($\left|r^h\right|_2$) after several V(2,1) multi-grid cycles on a 512x512 grid using point Gauß-Seidel relaxation for different types additional relaxations

4.6 Conclusions

In this chapter is has been shown that multi-grid methods are able to solve the Poisson problem and Poisson-like problems very efficiently. Convergence rates of 0.12 per cycle are possible and solutions which are accurate up to discretization accuracy are obtained in $\mathcal{O}(N)$ operations. However, often this efficiency does not extend automatically to other problems which appear as very similar. A new detailed analysis is required to detect slowly converging components on the fine grid and for a way to represent and solve these components on the coarse grid. Also boundary conditions do not need to cause any problems for multi-grid methods. Additional local relaxation work in the neighborhood of the boundary is usually sufficient to maintain optimal performance.

Chapter 5

Geometric multi-grid applied to the convection-diffusion equation

In this chapter the convection-diffusion equation in 2 dimensions is considered:

$$-\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial au}{\partial x}+\frac{\partial bu}{\partial y}=f\left(x,y\right)\qquad (x,y)\in\Omega \tag{5.1}$$

on a domain Ω with Dirichlet conditions on inflow boundaries and Neumann conditions on outflow boundaries. This is an important equation in CFD, since the combination of diffusion and convection is often found in nature, e.g. the momentum and energy equations describing fluid flows. a and b are the dimensionless convective speeds in x and y direction, respectively and the parameter $\epsilon\,[m]$ determines the ratio between diffusion and convection. Obviously, the convection-diffusion equation is a singularly pertubated equation: in the limit case $\epsilon\rightarrow 0$ it is no longer elliptic, but hyperbolic. The convection-diffusion equation is a linear model problem for the momentum equations in the Navier-Stokes equations and for the energy equation.

5.1 Discretization

The first complications for the convection-diffusion equation already show in the discretization of the convective terms. In this section three possible discretizations (central differencing, first-order upwind and κ-schemes) are discussed together with possible relaxation techniques. All discretizations are given for the case of constant coefficients (i.e. constant ϵ, a and b):

$$-\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+a\frac{\partial u}{\partial x}+b\frac{\partial u}{\partial y}=f\left(x,y\right) \tag{5.2}$$

In appendix A, the derivation of the discretizations for non constant coefficient cases is discussed.

5.1.1 Central differencing

A second-order accurate approximation on a uniform grid with gridsize h to the convective terms can be obtained using central differencing:

$$a\frac{\partial u}{\partial x}_{central}=\frac{a}{2h}\left[\begin{array}{ccc}-1 & 0 & 1\end{array}\right] \tag{5.3}$$

However, this discretization is only stable for a Péclet-number of:

$$Pe=\frac{h}{\epsilon}\max\left(\left|a\right|,\left|b\right|\right)\leq 2 \tag{5.4}$$

where Pe is the so-called mesh Péclet-number. This can easily be illustrated by solving the one dimensional, constant coefficient, convection-diffusion equation with Dirichlet boundary conditions:

$$\begin{aligned} -\epsilon\frac{\partial^2 u}{\partial x^2} + a\frac{\partial u}{\partial x} &= 0 \qquad x \in (0, L) \\ u(0) &= 0 \\ u(L) &= 1 \end{aligned} \tag{5.5}$$

The analytic solution is:

$$u(x) = \frac{1 - e^{\frac{ax}{\epsilon}}}{1 - e^{\frac{aL}{\epsilon}}} \tag{5.6}$$

For a central discretization of the convective and diffusive terms, the discrete equations become:

$$\begin{gathered} u_o = 0 \\ -\left(\epsilon + \tfrac{ah}{2}\right) u_{i-1} + 2\epsilon u_i - \left(\epsilon - \tfrac{ah}{2}\right) u_{i+1} = 0 \quad i \in [1..n-1] \\ u_n = 1 \end{gathered}$$

with $n+1$ the number of points and $h = L/n$ the gridsize. Substituting the general solution $u_i^h = Cq^i$, into the equation for all interior points yields:

$$-\left(\epsilon + \frac{ah}{2}\right) q^{-1} + 2\epsilon - \left(\epsilon - \frac{ah}{2}\right) q = 0$$

Solving for q gives:

$$\begin{aligned} q_1 &= 1 \\ q_2 &= \frac{2\epsilon + ah}{2\epsilon - ah} \end{aligned} \tag{5.7}$$

So the general solution becomes:

$$u_i^h = C_1 q_1^i + C_2 q_2^i = C_1 + C_2 q_2^i \tag{5.8}$$

The constants C_1 and C_2 can be determined by applying the boundary conditions which gives the discrete solution (see also figure 5.1):

$$u_i^h = \frac{1 - q_2^i}{1 - q_2^n} \tag{5.9}$$

When $q_2 < 0$ (or equivalently $n < \frac{|a|L}{2\epsilon}$), the numerical solution is highly oscillatory. This indicates that on such a grid the truncation error is too large for the discretized operator to mimic the physical behavior of the partial differential equation. Researchers have been able to solve this problem for central differencing by using special techniques at boundaries (i.e. Summation By Parts (SBP) and Simultaneous Approximation Term (SAT)), but these techniques fall beyond the scope of this thesis. [7, 12, 14]

5.1.2 First-order upwind discretization

A simple and stable discretization is the first-order upwind discretization. In this discretization, the first derivatives (convection terms) are approximated by one-sided differences such that only upstream grid points are used in the discretization of the convection terms. Depending on the sign of a, an $\mathcal{O}(h)$ accurate discretized of au_x is given by:

$$\begin{aligned} a\frac{\partial u}{\partial x}_{upwind} &= \begin{matrix} \frac{a}{h} \end{matrix} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \text{ if } a < 0 \\ &\quad \begin{matrix} \frac{a}{h} \end{matrix} \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \text{ if } a \geq 0 \\ &= \frac{1}{h} \begin{bmatrix} -\frac{1}{2}(a + |a|) & |a| & \frac{1}{2}(a - |a|) \end{bmatrix} \end{aligned} \tag{5.10}$$

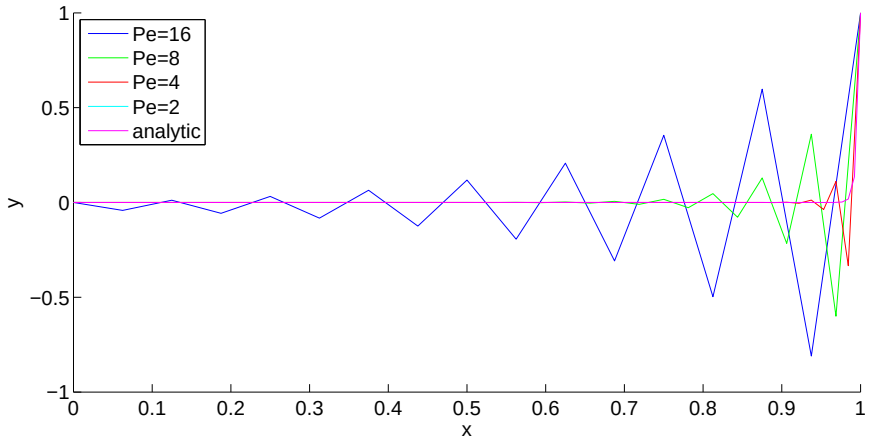


Figure 5.1: Convection-diffusion problem: 1D solutions with Dirichlet boundary conditions for different Péclet-numbers (by varying the number of grid points) using a second order central differencing discretization of the convective term, with $a > 0$

For $a > 0$ the discretized equations for the one-dimensional convection-diffusion equation (equation 5.5) become:

$$\begin{gathered} u_o = 0 \\ -(\epsilon + ah)\, u_{i-1} + (2\epsilon + ah)\, u_i - \epsilon u_{i+1} = 0 \quad i \in [1..n-1] \\ u_n = 1 \end{gathered}$$

with the discrete solution:

$$u_i^h = \frac{1 - q^i}{1 - q^n} \text{ with } q = 1 + \frac{ah}{\epsilon} \tag{5.11}$$

Since for $q > 0$ independent of h, the first-order upwind discretization does not exhibit any spurious oscillating behavior, for any mesh-size. The upwind discretization of the 2D convection-diffusion equation becomes:

$$L^h = \frac{1}{h^2} \begin{bmatrix} & -\epsilon + \frac{h}{2}(|b| - b) & \\ -\epsilon - \frac{h}{2}(|a| + a) & 4\epsilon + h(|a| + |b|) & -\epsilon + \frac{h}{2}(|a| - a) \\ & -\epsilon - \frac{h}{2}(|b| + b) & \end{bmatrix} \tag{5.12}$$

Relaxation for first-order upwind discretization

To illustrate the behavior of relaxations for the first-order upwind discretization, a local mode analysis on the special case $a \geq 0$ and $b \geq 0$ is considered. For this case the stencil of the operator simplifies to:

$$L^h = \frac{1}{h^2} \begin{bmatrix} & -\epsilon & \\ -\epsilon - ah & 4\epsilon + h(a + b) & -\epsilon \\ & -\epsilon - bh & \end{bmatrix} \tag{5.13}$$

It is immediately clear that pointwise Gauß-Seidel relaxation in the direction of the flow (downstream relaxation) will be an exact solver for $\epsilon \to 0$. For small ϵ only a few iterations are needed to solve the problem. This is also reflected in the LMA analysis:

$$\mu(\theta_1, \theta_2) = \left| \frac{-\epsilon e^{\theta_1 i} - \epsilon e^{\theta_2 i}}{-(\epsilon + ah)\, e^{\theta_1 i} + 4\epsilon + ah + bh - (\epsilon + bh)\, e^{\theta_2 i}} \right| \tag{5.14}$$

For $\epsilon = 0\,\mathrm{m}$ the numerator becomes zero and the relaxation is an exact solver. The worst behavior is obtained for $a = b = 0$, when only the diffusion part remains (the Poisson equation) which has an asymptotic convergence factor of 0.5.

Downstream Gauß-Seidel relaxation is a good smoother, but for most cases the direction of the flow is not known in advance or varies throughout the domain. Therefore four-directional Gauß-Seidel relaxation can be used. This involves four Gauß-Seidel iteration sweeps, each starting at a different corner of the domain. This is allowed because Gauß-Seidel relaxation does not diverge when applied in the wrong direction, although it does not converge either. This can be shown by computing the smoothing factor of the relaxation in upstream direction:

$$\mu\left(\theta_1,\theta_2\right)=\left|\frac{-\left(\epsilon+ah\right)e^{\theta_1 i}-\left(\epsilon+bh\right)e^{\theta_2 i}}{-\epsilon e^{\theta_1 i}+4\epsilon+ah+bh-\epsilon e^{\theta_2 i}}\right| \tag{5.15}$$

The worst performance is obtained for the case $\epsilon = 0\,\mathrm{m}$:

$$\mu\left(\theta_1,\theta_2\right)=\left|\frac{ae^{\theta_1 i}+be^{\theta_2 i}}{a+b}\right| \tag{5.16}$$

However, for this case the numerator is always smaller than or equal to the denominator, so the relaxation scheme does indeed not diverge.

Similar results can be obtained for alternating symmetric line Gauß-Seidel relaxation. This relaxation consist of four steps: a forward and a backward line relaxation step in each direction. The advantage of line relaxation over point relaxation is that excellent smoothing factors are also obtained for anisotropic diffusion problems, so problems involving a combination of anisotropy and dominating convection are handled well by these smoothers.

Need for higher-order

A major drawback of the first-order discretization is its low order of accuracy. A Taylor series expansion of the operator, shows that the error in the discretization is $\frac{|a|h}{2}u_{xx}$. The derivative associated with the error of the first-order upwind schemes is the same as the derivative associated with the diffusion terms. Therefore the error is also referred to as artificial viscosity. For small ϵ the artificial viscosity dominates the physical viscosity unless an extremely fine grid is used ($h \to 0$). In realistic flow applications $\epsilon/aL = \mathcal{O}\left(Re^{-1}\right)$, so $ah < \epsilon$ implies that a grid with $h/L < 1/Re$ is required. This accuracy condition is very similar to the mesh requirement for stability of the central scheme. Therefore first-order upwind schemes are not suitable for high Reynolds number flow problems and higher-order upwind-biased schemes have to be used.

5.1.3 κ-schemes

A special class of upwind biased higher-order discretization approximation schemes are the so called κ-schemes proposed by van Leer [8]. For the convective term au_x, the scheme is defined as:

$$a\frac{\partial u}{\partial x}_{\kappa}=\begin{cases}\frac{a}{2h}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}-\frac{a}{4h}\left(1-\kappa\right)\begin{bmatrix}0 & -1 & 3 & -3 & 1\end{bmatrix} & \text{if } a<0\\ \frac{a}{2h}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}-\frac{a}{4h}\left(1-\kappa\right)\begin{bmatrix}-1 & 3 & -3 & 1 & 0\end{bmatrix} & \text{if } a\geq 0\end{cases} \tag{5.17}$$

The stencil is the sum of the central difference scheme and a second-order dissipation term, which can be interpreted as an (upwind) approximation of $\frac{ah^2}{4}\left(\kappa-1\right)u_{xxx}$. For specific values of κ, some well-known schemes are obtained:

- $\kappa = -1$ Fully upwind scheme, which is $\mathcal{O}(h^2)$
- $\kappa = 0$ Fromm's scheme, which is $\mathcal{O}(h^2)$ (always used in this research)
- $\kappa = \frac{1}{3}$ CUI scheme "cubic upwind interpolation", which is $\mathcal{O}(h^3)$
- $\kappa = \frac{1}{2}$ QUICK scheme "quadratic upwind interpolation for convective kinematics", which is $\mathcal{O}(h^2)$
- $\kappa = 1$ the central difference scheme, which is $\mathcal{O}(h^2)$

The κ-schemes can be stable for a wider range of problems than the central discretization schemes. However, they still produce unphysical oscillations near sharp gradients in the solution or discontinuities in boundary condition. In order to suppress such spurious oscillations so-called limiters have been introduced. These limiters are based on the total variation diminishing (TVD) concept.

Limiters

In this section a brief overview of the concept of total variation and limiters is given. For a more detailed explanation see van Leer [8] or Leveque [9]. The concept finds its origin in the scalar conservation law:

$$\frac{\partial u}{\partial t} + \frac{\partial au}{\partial x} = 0 \tag{5.18}$$

The total variation, which is defined as

$$TV = \int \left| \frac{\partial u}{\partial x} \right| dx \tag{5.19}$$

of any physically admissible solution to equation 5.18 does not increase in time:

$$\frac{\partial}{\partial t} TV \leq 0 \tag{5.20}$$

When a numerical scheme for equation 5.18 is considered, it is important that such a scheme also has the non-increasing TV property. Such schemes are called Total Variation Diminishing, or TVD, schemes. To check if a scheme is TVD, first a discretized analog of equation 5.19 must be defined. Therefore consider a 1D grid with coordinates $x_i = ih$ and corresponding solutions u_i. The total variation of the grid function u_i^h is then defined as:

$$TV\left(u\right) = \sum_i \left| u_i - u_{i-1} \right| \tag{5.21}$$

A discrete scheme is now called TVD if for the state at the new time level (u^{n+1}), the total variation is less than or equal to the total variation of the current level (u^n), i.e.

$$TV\left(u^{n+1}\right) \leq TV\left(u^n\right) \tag{5.22}$$

It can be proven that a scheme that obeys equation 5.22 is monotonic [8], where monotonicity is defined as:

$$\min\left(u_{i-1}, u_{i+1}\right) \leq u_i \leq \max\left(u_{i-1}, u_{i+1}\right), \quad \forall i \tag{5.23}$$

Now consider the semi-discretization (i.e. only the spatial part is discretized) of equation 5.18 at grid point i:

$$\frac{\partial u_i}{\partial t} = \sum_k c_k \left(u_{i+k} - u_i\right) \tag{5.24}$$

where the coefficients c_k are a function of the equation to be solved and the discretization. For a scheme to be monotonic, and thus TVD, the coefficients c_k must obey the following condition: [14]

$$c_k \geq 0 \tag{5.25}$$

Consider for example the first-order upwind discretization (equation 5.10). For a constant positive convection speed a, the semi-discretization of equation 5.18 in point i becomes:

$$\frac{\partial u}{\partial t} + a\frac{u_i - u_{i-1}}{h} = 0 \Longrightarrow \frac{\partial u}{\partial t} = a\frac{u_{i-1} - u_i}{h} \tag{5.26}$$

So the coefficients in equation 5.24 become:

$$c_k = \begin{cases} \frac{a}{h} & \text{if } k = -1 \\ 0 & \text{otherwise} \end{cases} \tag{5.27}$$

This clearly satisfies equation 5.25 so the first-order upwind discretization is total variation diminishing.

Now consider the κ-scheme discretization (equation 5.17). For constant positive convection speeds the semi-discretization of equation 5.18 becomes:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{a}{4h}\left((1-\kappa)\,u_{i-2} - (5-3\kappa)\,u_{i-1} + (3-3\kappa)\,u_i + (1+\kappa)\,u_{i+1}\right) &= 0 \\ \frac{\partial u}{\partial t} = \frac{a}{4h}\left((\kappa-1)\,(u_{i-2}-u_i) + (5-3\kappa)\,(u_{i-1}-u_i) + (-\kappa-1)\,(u_{i+1}-u_i)\right) & \end{aligned} \tag{5.28}$$

So the coefficients in equation 5.24 become:

$$c_k = \begin{cases} \frac{a}{4h}\,(\kappa-1) & \text{if } k=-2 \\ \frac{a}{4h}\,(5-3\kappa) & \text{if } k=-1 \\ \frac{a}{4h}\,(-\kappa-1) & \text{if } k=1 \\ 0 & \text{otherwise} \end{cases} \tag{5.29}$$

This leads to the following requirements for a monotonic scheme:

$$\kappa \geq 1 \qquad \kappa \leq \frac{5}{3} \qquad \kappa \leq -1 \tag{5.30}$$

These requirements are clearly conflicting and hence none of the linear κ-schemes is monotonic and thus TVD. This turns out to be a general result, leading to the conclusion that a linear monotonic scheme for the discretization of a hyperbolic system can only be first-order accurate. This is known as Godunov's order barrier theorem. Many scientific papers have already been written about this topic, e.g. [8]. Most often solutions are found by using a nonlinear reconstruction on the interfaces $(i \pm 1/2)$. The partial derivative is then approximated as:

$$a\frac{\partial u}{\partial x} = \frac{a}{h}\left(-u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}\right) \tag{5.31}$$

with

$$\begin{aligned} u_{i-\frac{1}{2}} &= \begin{cases} u_{i-1} + R_{i-1}\left(\frac{1+\kappa}{4}\,(u_i - u_{i-1}) + \frac{1-\kappa}{4}\,(u_{i-1} - u_{i-2})\right), & \text{if } a_{i-\frac{1}{2}} \geq 0 \\ u_i - R_i\left(\frac{1+\kappa}{4}\,(u_i - u_{i-1}) - \frac{1-\kappa}{4}\,(u_{i+1} - u_i)\right), & \text{if } a_{i-\frac{1}{2}} < 0 \end{cases} \\ u_{i+\frac{1}{2}} &= \begin{cases} u_i + R_i\left(\frac{1+\kappa}{4}\,(u_{i+1} - u_i) + \frac{1-\kappa}{4}\,(u_i - u_{i-1})\right), & \text{if } a_{i+\frac{1}{2}} \geq 0 \\ u_{i+1} - R_{i+1}\left(\frac{1+\kappa}{4}\,(u_{i+1} - u_i) - \frac{1-\kappa}{4}\,(u_{i+2} - u_{i+1})\right), & \text{if } a_{i+\frac{1}{2}} < 0 \end{cases} \end{aligned} \tag{5.32}$$

where R_i is a limiter function. In the literature several possibilities for these limiter functions are given. However, describing them is beyond the scope of this thesis. All of them have the property that the resulting discretization is second-order accurate and total variation diminishing. In this research the Van Albada limiter is used (whenever limiters are used), for which R_i can be written as:

$$R_i = \frac{2\,(u_i - u_{i-1})\,(u_{i+1} - u_i) + 2d^2}{(u_i - u_{i-1})^2 + (u_{i+1} - u_i)^2 + 2d^2} \tag{5.33}$$

with d some arbitrary small number to make sure that $R_i = 1$ for constant u.

Smoothing for κ-scheme discretizations

To illustrate the smoothing behavior for the discretization of the convection-diffusion equation based on the κ-schemes, LMA analysis for the special case $a \geq 0$ and $b \geq 0$ is considered. For this

case the discretization (without limiters) becomes:

$$L^h = \frac{1}{4h}\begin{bmatrix} & & 0 & & \\ & & b\,(1+\kappa) - \frac{4\epsilon}{h} & & \\ a\,(1-\kappa) & a\,(3\kappa-5) - \frac{4\epsilon}{h} & a\,(3-3\kappa) + b\,(3-3\kappa) + \frac{16\epsilon}{h} & a\,(1+\kappa) - \frac{4\epsilon}{h} & 0 \\ & & b\,(3\kappa-5) - \frac{4\epsilon}{h} & & \\ & & b\,(1-\kappa) & & \end{bmatrix} \tag{5.34}$$

Straightforward application of smoothers discussed so far (i.e. Jacobi and Gauß-Seidel in both point and line form) is not suitable for the κ-schemes. Smoothing factors for a convection dominated problem (i.e. $\epsilon \to 0$) often turn out to be larger than 1.0. [10, 12]

One possibility is to use smoothers, in which the operator is split into three parts:

$$L^h = L^h_+ + L^h_0 + L^h_- \tag{5.35}$$

where L^h_+ corresponds to the weights of already updated variables, L^h_0 to the weights of all variables to be solved for in the current relaxation step and L^h_- the remaining weights. One relaxation step now consists of solving $\bar{u}^h$ from:

$$L^h_0 \bar{u}^h = f^h - L^h_- u^h - L^h_+ \bar{u}^h \tag{5.36}$$

For relaxations able to smooth the discretization of κ-schemes, the splitting is motivated by the defect-correction approach. This technique is used to obtain high-order accuracy by employing low-order accurate schemes. Consider the problem:

$$L^h u^h = f^h \tag{5.37}$$

where L^h is a higher-order accurate discretization of L. A general defect correction iteration can be performed by solving:

$$\hat{L}^h \bar{u}^h = \hat{f}^h \text{ with } \hat{f}^h = f^h - L^h u^h - \hat{L}^h u^h \tag{5.38}$$

for $\bar{u}^h$, where $\hat{L}^h$ is a lower-order accurate discretization of L.

Now let L^h be a κ-scheme based discrete operator. For lexicographical x-line κ-smoothing the operator L_h is split according to equation 5.36 with L^h_0 based on a lower-accuracy operator (at least in x-direction):

$$L^h_+ = \frac{1}{4h}\begin{bmatrix} & & 0 & & \\ & & 0 & & \\ 0 & 0 & 0 & 0 & 0 \\ & & b\,(3\kappa-5) - \frac{4\epsilon}{h} & & \\ & & b\,(1-\kappa) & & \end{bmatrix} \tag{5.39}$$

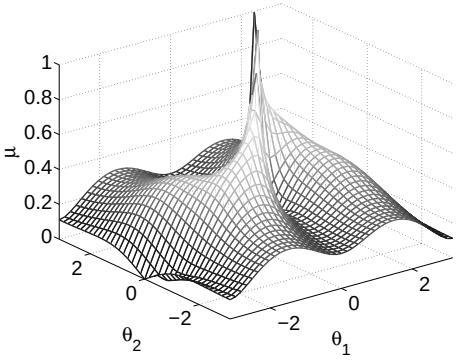
$$L^h_0 = \frac{1}{4h}\begin{bmatrix} & & 0 & & \\ & & 0 & & \\ 0 & -4a - \frac{4\epsilon}{h} & 4a + 4b + \frac{16\epsilon}{h} & -\frac{4\epsilon}{h} & 0 \\ & & 0 & & \\ & & 0 & & \end{bmatrix} \tag{5.40}$$

$$L^h_- = \frac{1}{4h}\begin{bmatrix} & & 0 & & \\ & & b\,(1+\kappa) - \frac{4\epsilon}{h} & & \\ a\,(1-\kappa) & a\,(3\kappa-1) & a\,(-1-3\kappa) + b\,(-1-3\kappa) & a\,(1+\kappa) & 0 \\ & & 0 & & \\ & & 0 & & \end{bmatrix} \tag{5.41}$$

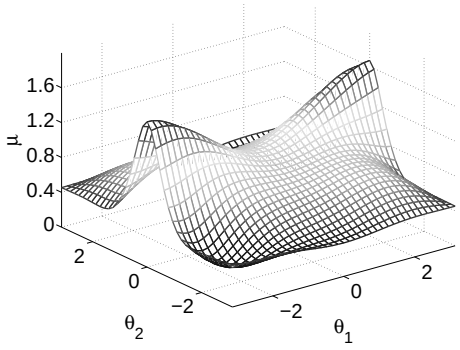
Smoothers based on this splitting are referred to as κ-smoothers. Local mode analysis shows acceptable convergence when applied in the downstream direction (see figure 5.2(a)). Relaxations in upstream direction (figure 5.2(b)) show divergence for certain grid functions. However, downstream relaxation is able to counteract the diverging behavior, making a four directional symmetric

κ-smoother a robust solver for discretizations of the convection-diffusion equation (see figure 5.2(c) and table 5.1).

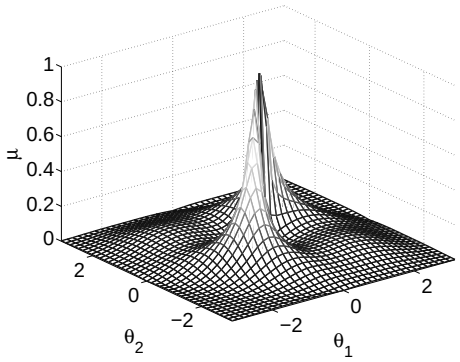
Whenever limiters are used the same smoother can be employed. In the line solver the weights of the variables to be solved for (L_0^h) are those of the first-order upwind discretization and therefore they do not depend on the limiters. Only the right hand side (i.e. L_+^h and L_-^h) depends on the choice of limiters and can easily be calculated using the currently available approximations.



(a) Downstream



(b) Upstream



(c) Four directional symmetric

Figure 5.2: Convection-diffusion problem: LMA analysis for κ-smoothers applied to the κ-scheme discretization of the convective terms ($\kappa = 0$, $a = \cos\frac{\pi}{4}$, $b = \cos\frac{\pi}{4}$, $\epsilon = 1e-5\,\mathrm{m}$ and $h = \frac{1}{256}$

$\epsilon = 0\,\mathrm{m}$					$\epsilon = \infty$
0	$\pi\frac{1}{8}$	$\pi\frac{1}{4}$	$\pi\frac{3}{8}$	$\pi\frac{1}{2}$	
0.079	0.233	0.176	0.233	0.079	0.022

Table 5.1: Convection-diffusion problem: Smoothing factors for four directional symmetric κ-smoother applied to the κ-scheme discretization of the convective terms, in the hyperbolic limit for different angles ($a = \cos\alpha$, $b = \sin\alpha$) and in the elliptic limit

5.2 Smith-Hutton Problem

In this section the results of suitable types of discretizations (first-order upwind and κ-schemes) applied to a convection-diffusion equation with discontinuous boundary conditions are presented. Such a contact discontinuity results in large gradients in the solution, which is similar to problems

occurring for boundary layer solutions. A well known example is the Smith-Hutton problem [11]:

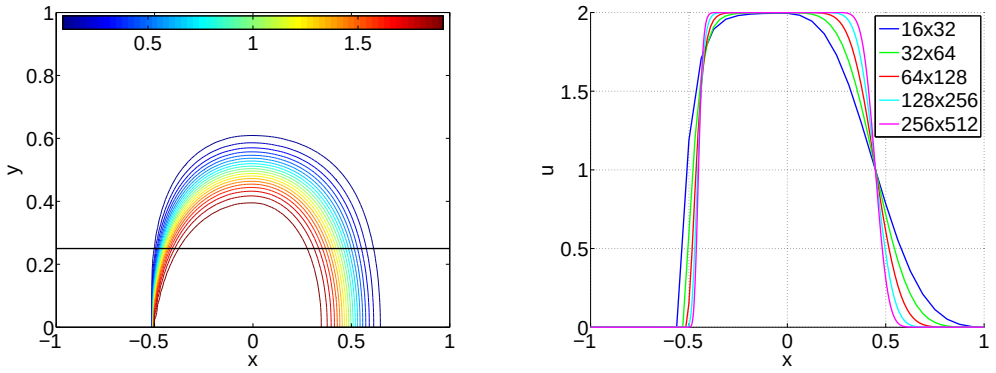
$$\begin{gathered} -\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial au}{\partial x}+\frac{\partial bu}{\partial y}=f\left(x,y\right) \text{ on } \Omega=[-1,1]\,X\,[0,1] \\ a=2y\left(1-x^2\right) \\ b=-2x\left(1-y^2\right) \\ \epsilon=10^{-5}\,\mathrm{m} \end{gathered} \tag{5.42}$$

and boundary conditions

$$\begin{aligned} u&=2, \quad \text{if} \quad -\frac{1}{2}\leq x\leq 0 \text{ and } y=0 \\ \frac{\partial u}{\partial y}&=0, \quad \text{if} \quad 0<x\leq 1 \text{ and } y=0 \\ u&=0, \quad \text{elsewhere} \end{aligned} \tag{5.43}$$

5.2.1 First-order upwind

Obtaining solutions for the Smith-Hutton problem using the first-order upwind discretization of the convection terms is relatively easy. Because of the small viscosity, downstream Gauß-Seidel relaxation itself already acts as an exact solver and only a few multi-grid cycles are necessary to reduce the residual by several orders of magnitude. Solutions are shown in figure 5.3. In the left figure contours of the solution are shown on a grid consisting of 64x128 cells and in the right figure solutions on the line $y = 0.25$ for different grid sizes. From these figures the artificial diffusion is clearly visible. The slopes on the coarse grids are much to smooth. The width of the region where u^h drops from two to zero should be $\mathcal{O}(\epsilon)$ and independent of the grid size, which is certainly not the case. For coarse grids the solutions smear out too much and unphysical solutions are obtained. This is exactly why higher-order discretizations of the convection terms are necessary.



(a) Contour lines on a 64x128 grid

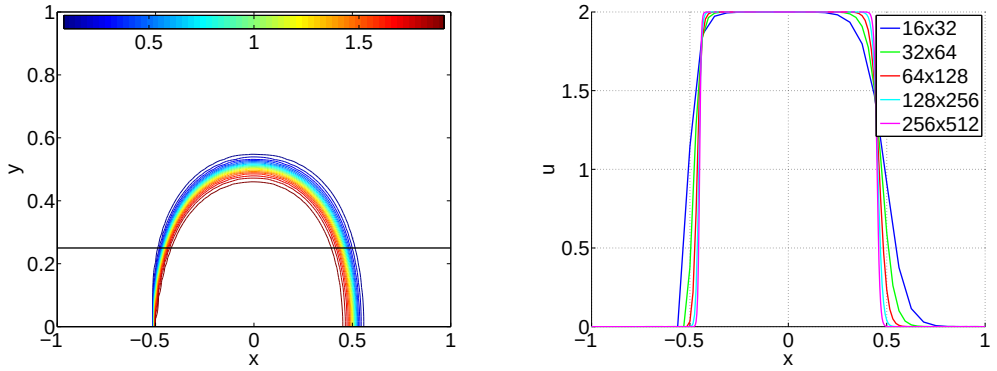
(b) Solutions along $y = 0.25$ for different mesh-sizes

Figure 5.3: Smith-Hutton problem: Solutions for first-order upwind discretization of the convection-diffusion equation

5.2.2 κ-schemes

Solutions for the Smith-Hutton problem for a κ-scheme discretization ($\kappa = 0$, Van Albada limiter) of the convection terms are shown in figure 5.4. Compared to figure 5.3 the reduction in artificial viscosity is immediately clear and solutions are much more accurate even for coarse grids.

The improved accuracy, however, comes at a cost, namely obtaining converged solutions is much more difficult than for the first-order scheme. After a few multi-level cycles the residual consists

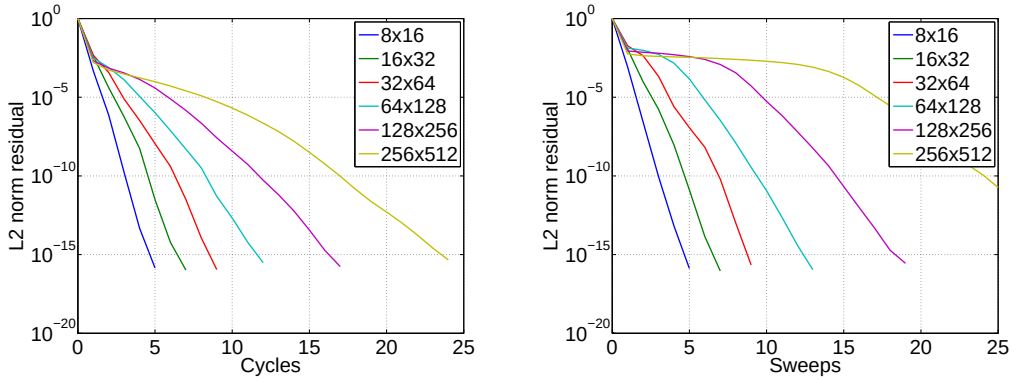


(a) Contourlines 64x128

(b) Solutions at $y = 0.25$ for different mesh-sizes

Figure 5.4: Smith-Hutton problem: Solutions for the κ-scheme discretization of the convection-diffusion equation with Van Albada limiter ($\kappa = 0$)

mainly of some highly oscillating components perpendicular to the flow. These components cannot be represented accurately on coarser grids and thus have to be eliminated by relaxations on the finest level. However, local relaxations fail to reduce these error components efficiently, because the operator has nearly no coupling in this direction (because of the low viscosity). The only way to reduce these residuals is by improving the upstream solution and convecting this solution. For higher-order schemes this process is mesh-size dependent, so it is impossible to obtain grid-independent convergence rates. This is reflected in figure 5.5 where the residuals for the Smith-Hutton problem are plotted as a function of the number of cycles for several grid sizes. In figure 5.5(a) V(2,1) cycles are used, while in figure 5.5(b) only single-grid relaxation is used (3 relaxations per sweep to obtain comparable results). At first the multi-grid cycles converge faster than the single-grid sweeps, but after a few cycles both approaches show equal performance. Theoretically the same problems should occur for the first-order upwind scheme. However, for this discretization the upstream solution is convected with the flow in just one relaxation sweep (for all mesh-sizes) and thus good performance is achieved.



(a) As a function of the number of V(2,1) multi-level cycles

(b) As a function of the number of triple single-grid relaxations

Figure 5.5: Smith-Hutton problem: L_2-norm of the residual ($\left|r^h\right|_2$) for the κ-scheme discretization of the convective terms with Van Albada limiter and $\kappa = 0$

5.3 Coarse grid correction problem

Although multi-grid convergence for the first-order discretization of the Smith-Hutton problem seems convincing, difficulties may arise for certain convection-diffusion problems with very small ϵ. The true difficulty of convection dominated problems is not the smoother, but the coarse grid correction. With standard coarsening and standard coarse grid operators the two-grid convergence factor is limited by 0.5 for first-order schemes. For higher-order schemes this problem is even more severe. The factor of 0.5 is due to the fact that first-order upwind differencing for the coarse grid produces an artificial viscosity on the coarse grid, which is twice as large as on the fine grid.

This factor can already be seen from the simplified two-grid analysis (see section 2.2.3). For the first-order upwind scheme without diffusion:

$$\begin{aligned} 1 - \frac{\tilde{L}^h(\theta)}{\tilde{L}^H(2\theta)} &= 1 - \frac{a\left(1-e^{-i\theta_1}\right)+b\left(1-e^{-i\theta_2}\right)}{a\left(1-e^{-2i\theta_1}\right)+b\left(1-e^{-2i\theta_2}\right)}\frac{H}{h} \\ \lim_{\theta\to 0}\left(1 - \frac{\tilde{L}^h(\theta)}{\tilde{L}^{2h}(\theta)}\right) &= \begin{cases} \frac{H}{4h}, & \text{if} \quad a\theta_1 = -b\theta_2 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5.44}$$

For higher-order schemes the simplified two-grid analysis even gives a maximum convergence rate of 0.75 (or 0.83 for $|a| = |b|$ see equation 5.59).

The coarse grid correction problem can also be shown by a simple analysis. Consider the Taylor series expansion of the first-order upwind discretization of the pure convection equation for $a > 0, b > 0$.

$$L^h \langle u \rangle = a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \frac{h}{2}\left(a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}\right) + \mathcal{O}\left\langle h^2 \right\rangle \tag{5.45}$$

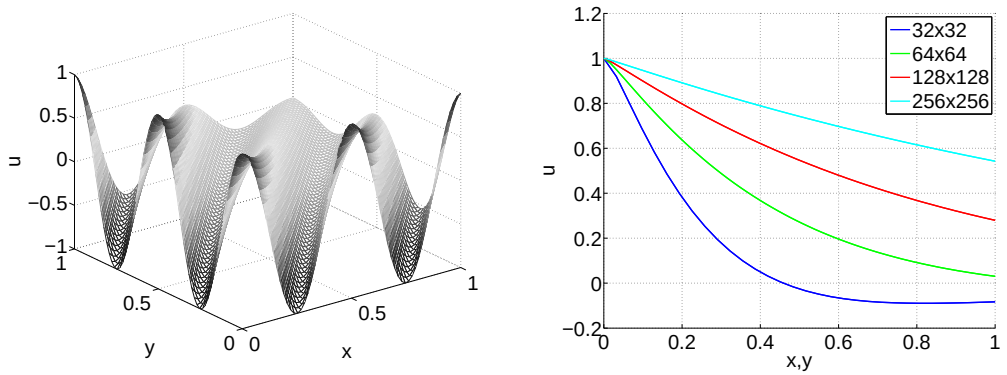
The third term on the right-hand side is the main error associated with the first-order upwind scheme. Introducing a rotated (ξ, η) axis system, with ξ in streamwise direction and η perpendicular to the flow, yields: (for $\epsilon = 0\,\mathrm{m}$)

$$L^h \langle u \rangle = \sqrt{a^2+b^2}\frac{\partial u}{\partial \xi} - \frac{h}{2}\left(\frac{a^3+b^3}{a^2+b^2}\frac{\partial^2 u}{\partial \xi^2} - 2ab\frac{a-b}{a^2+b^2}\frac{\partial^2 u}{\partial \xi \eta} + ab\frac{a+b}{a^2+b^2}\frac{\partial^2 u}{\partial \eta^2} + \mathcal{O}\left\langle h^2 \right\rangle\right) \tag{5.46}$$

Perpendicular to the flow, there is no convection, so the main term in this direction is an error term. For small viscosity ($\epsilon \to 0\,\mathrm{m}$), the solution in the η direction is therefore dominated by an error term. This term depends on the grid size h, so solutions will be different on different grids. This is shown more clearly in figure 5.6, where the results are shown for the case $a = 1$ and $b = 1$ in combination with a sinusoidal inlet condition. The exact numerical solution can be obtained by using Gauß-Seidel relaxation and is shown on a grid with 64x64 cells in figure 5.6(a). In figure 5.6(b) the solution on the line $x = y$ is shown for different grid sizes. From these graphs it becomes immediately clear that the sinusoidal in let condition decays much faster on coarser grids, due to the higher artificial viscosity. This is exactly the coarse grid problem: there exist some low-frequency components (the so-called characteristic Fourier components) which are damped more on coarser grids than on the fine grid. These components are not "correctly" approximated on the coarse grid and therefore they do not give the required correction to the fine grid. A two-level factor of 0.5 caused by the coarse grid correction leads to an even worse multi-grid convergence. The convergence factor of a multi-grid cycle can be estimated from the two-grid factor. If it is assumed that high-frequency components are correctly smoothed, that the low-frequency components remain unchanged under smoothing, and that the two-level factor is determined by the coarse grid approximation of low frequency components, a prediction of the convergence factor for multi-grid cycles with l levels can be given by:

$$\rho_l = 1 - (1-\rho)\left(1 - \rho_{l-1}^{\gamma}\right) \tag{5.47}$$

with ρ the simplified two-grid convergence factor, ρ_l the convergence factor for level l ($\rho_0 = 0$) and γ the grid parameter (i.e. $\gamma = 1$ for V-cycles and 2 for W-cycles).



(a) solution on 64x64 grid

(b) solution on the line $x = y$ for different grids

Figure 5.6: Convection-diffusion problem: Solution using the first-order upwind discretization of the convective terms for $a = \cos\frac{\pi}{4}$, $b = \sin\frac{\pi}{4}$ and $\epsilon = 0\,\mathrm{m}$ with sinusoidal inlet condition at $x = 0$ and at $y = 0$

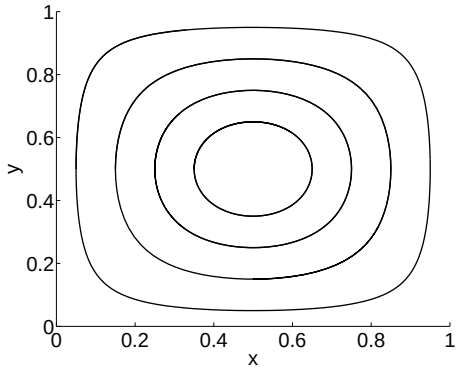
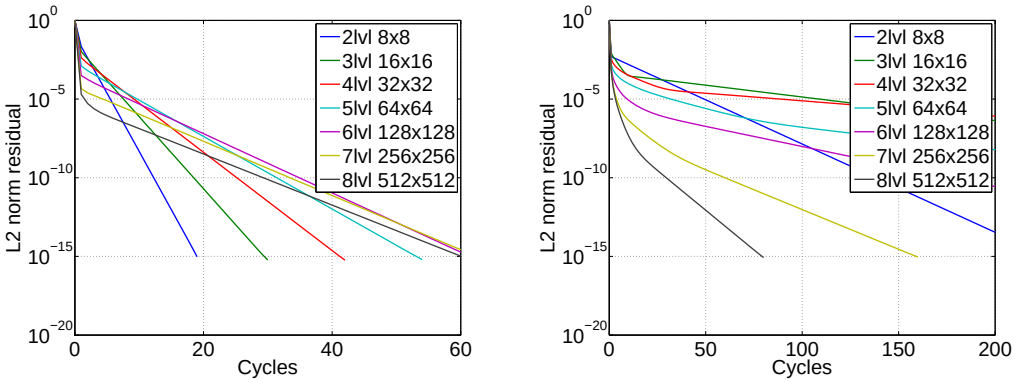


Figure 5.7: Recirculating convection-diffusion problem: Streamlines



(a) First-order upwind discretization of convective terms, using four-directional Gauß-Seidel relaxation

(b) κ-schemes discretization without limiters ($\kappa = 0$) of the convective terms, using the four-directional symmetric κ-smoother

Figure 5.8: Recirculating convection-diffusion problem: Residual ($\left|r^h\right|_2$) as a function of the number of W(2,1) multi-grid cycles

The two-grid factor of 0.5 is not always observed in practice, e.g. in flow simulations for so-called entering flows, i.e. flows that enter the domain through a boundary, see the example discussed above. Characteristic components responsible for the low two-grid factor, that meet upstream Dirichlet boundaries (with zero errors at these boundaries) are automatically reduced by the downstream smoothers. An example for which the 0.5 factor is observed is the convection dominated recirculating flow problem. This is the normal convection-diffusion equation with Dirichlet boundary conditions, but with a special flow pattern:

$$\begin{aligned} \epsilon &= 10^{-5}\,\mathrm{m} \\ a(x,y) &= -sin(\pi x)\,cos(\pi y) \\ b(x,y) &= sin(\pi y)\,cos(\pi x) \end{aligned} \tag{5.48}$$

For this case the characteristics form closed loops and thus never meet upstream Dirichlet boundaries (see figure 5.7). Convergence results of applying W(2,1) cycles are shown in figure 5.8. In the left figure multi-grid results for the first-order upwind discretization of the convection-diffusion equation are shown. When more levels are used the convergence rate deteriorates, because of inaccurate corrections from the coarser grids. In the right figure results are shown for which multi-grid is applied to the higher-order upwind discretization of the convection-diffusion equation. For coarse target meshes (up to 64 by 64 grid cells) the performance is unsatisfactory because of the bad coarse grid correction. When finer target grids are used, the physical diffusion plays a more important role in the discretization and performance becomes somewhat better. For even larger mesh-sizes (i.e. $\epsilon \gg \max(a,b)\,h^2$) the physical diffusion will dominate over the error terms coming from the convective terms and corrections coming from the second finest grid will be accurate. Accurate solutions on the second finest grid are achieved because a W-cycle is used.

5.4 Solution 1: Improved coarse grid operator

Obtaining good coarse-grid corrections when the coarse grid stencil is the direct discretization of the equations on the coarse grid is impossible, because the smooth characteristic components receive only a fraction of the necessary correction from the coarse grid. This fraction is caused by the fact that for characteristic components the first truncation term perpendicular to the flow is different for each grid and dominates the operator (since the main term vanishes). The simplest solution would therefore be to construct an operator whose first truncation term approximates the first truncation term of the fine grid operator [16]. For the first-order upwind scheme this can be done by combining the first-order upwind scheme and the κ-schemes on coarser grids. The improved coarse-grid operator than becomes:

$$L^H_{improved} = \alpha L^H_{upwind} + (1-\alpha)\,L^H_\kappa \tag{5.49}$$

with

$$\alpha = \left(\frac{h}{H}\right) \tag{5.50}$$

For the special case $a > 0$, $b > 0$ and $\kappa = 0$ the stencil becomes:

$$L^H_{improved} = \frac{1}{4h}\begin{bmatrix} & & \\ a(1-\alpha) & -a(5-\alpha)-\frac{4\epsilon}{h} & \begin{matrix} 0 \\ b(1-\alpha)-\frac{4\epsilon}{h} \\ (a+b)(3+\alpha)+\frac{16\epsilon}{h} \\ -b(5-\alpha)-\frac{4\epsilon}{h} \\ b(1-\alpha) \end{matrix} & a(1-\alpha)-\frac{4\epsilon}{h} & 0 \\ & & \end{bmatrix} \tag{5.51}$$

Evidently, since the weights sum up to 1.0, $L^H_{improved}$ is a consistent discretization. The leading truncation term of L^H_{upwind} is $\mathcal{O}(H)$, so the leading truncation term of αL^H_{upwind} is $\mathcal{O}(\alpha H) = \mathcal{O}(h)$.

The leading truncation term of L_κ^H is $\mathcal{O}\left(H^2\right)$. Therefore, the total leading truncation term of $L_{improved}^H$ is equal to $\mathcal{O}\left(h\right) + \mathcal{O}\left(H^2\right)$. This is equal to the leading truncation term of L_{upwind}^h as long as:

$$H^2 < h \tag{5.52}$$

Applying the simplified two-grid analysis (similar to 5.44) to the improved coarse grid operator yields:

$$1 - \frac{\tilde{L}^h(\theta)}{\tilde{L}^H(2\theta)} = 1 - \frac{a\left(1-e^{-i\theta_1}\right)+b\left(1-e^{-i\theta_2}\right)}{\frac{a}{8}\left(e^{-4i\theta_1}-9e^{-2i\theta_1}+7+e^{2i\theta_1}\right)+\frac{b}{8}\left(e^{-4i\theta_2}-9e^{-2i\theta_2}+7+e^{2i\theta_2}\right)}\frac{H}{h}$$
$$\lim_{\theta\to 0}\left(1 - \frac{\tilde{L}^h(\theta)}{\tilde{L}^{2h}(\theta)}\right) = 0 \tag{5.53}$$

A simplified two-grid convergence factor of zero implies that at least the very low frequencies are approximated correctly by the coarse grid operator. The same result can be obtain by performing a Taylor series expansion of the inviscid improved operator for $a > 0$ and $b > 0$:

$$\begin{aligned} L_{improved}^H \langle u \rangle &= a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \alpha\frac{H}{2}\left(a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}\right) + \mathcal{O}\left\langle H^2 \right\rangle \\ &= a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \frac{h}{2}\left(a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}\right) + \mathcal{O}\left\langle H^2 \right\rangle \end{aligned} \tag{5.54}$$

This is exactly the same as for the fine grid operator (equation 5.45). Thus the improved operator overcomes the problems associated with direct coarsening of the first-order upwind operator. Smoothing the improved discretization can be done by applying the κ-smoother discussed in section 5.1.3.

5.4.1 Results

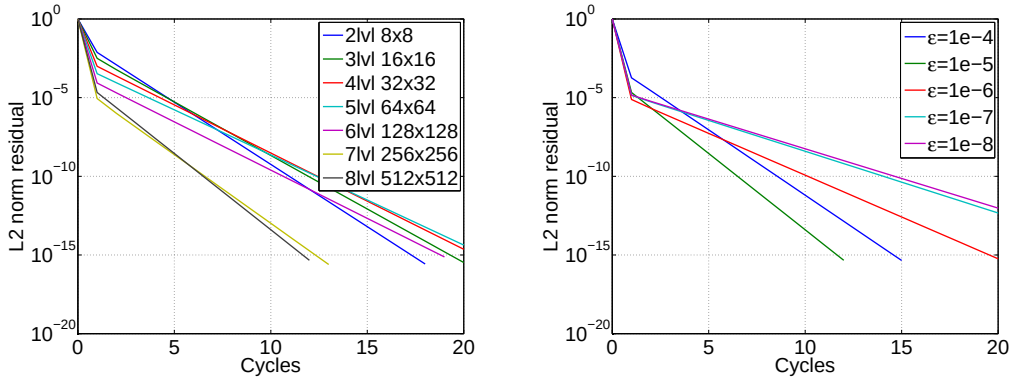
Convergence results of applying W(2,1) cycles with improved coarse grid operators for the recirculating convection-diffusion problem are shown in figure 5.9(a). Convergence is good for all target grid sizes (compare to figure 5.8(a)). For dense target grids (i.e. 256x256 and 512x512) the convergence is even better than for coarser grids. This is due to diffusion terms (ϵ/h^2), that become dominant over the convective terms (a/h and b/h) on coarser grids (or smaller grid spacing). Therefore the two-grid convergence increases towards convergence rates usually observed for Poisson problems (0.13 for V(2,1) cycles). Convergence on coarser grids is somewhat slower, but due to the W-cycles the corrections calculated on coarser grid are good enough to provide fast convergence. This observation is also verified by results obtained for the calculations using different values for ϵ on a target grid of 512x512 (see figure 5.9(b)). For higher diffusion coefficients, the convergence rate increases to that of the Poisson problem.

5.5 Solution 2: Galerkin coarsening

Another method to obtain accurate coarse grid operators is by using Galerkin coarsening (see section 2.1.2). However, Galerkin coarsening introduces some additional problems:

- The operator is not diagonally dominant (nor symmetric positive definite) on coarser grids, so standard relaxation methods (i.e. Gauß-Seidel and Jacobi) may not converge anymore. Smoothers which work for a larger class of problems, i.e. Kaczmarz relaxation (see appendix C), can help to overcome this problem.

- Standard numerical Galerkin coarsening can only be applied to linear operators. Extending this technique to non-linear operators is beyond the scope of this research. Therefore only discretizations of the κ-schemes without limiters are considered in this section.



(a) 4x4 as the coarsest grid

(b) 512x512 as the target grid for different values of ϵ

Figure 5.9: Recirculating convection-diffusion problem: L_2-norm of the residual ($\left|r^h\right|_2$) as a function of the number of W(2,1) multi-grid cycles for the first-order upwind discretization of the convective terms using improved coarse grid operators (with Van Albada limiter and $\kappa = 0$)

- It is possible that the coarse grid operator is not TVD and thus can introduce unwanted oscillations. This can be overcome by using smoothers which are able to efficiently smooth these (high-frequency) oscillations on finer grids, such as Gauß-Seidel relaxation for the first-order discretization and κ-smoothers for higher-order discretizations.

5.5.1 First-order

When full weighting and bi-linear interpolation are used as restriction and interpolation operators, respectively, the coarse grid operator of the standard first-order upwind discretization (5.13) becomes:

$$L^H = \frac{1}{32H}\begin{bmatrix} -3a+b & 2a+6b & a+b \\ -18a+2b & 12a+12b & 6a+2b \\ -3a-3b & 2a-18b & a-3b \end{bmatrix} + \frac{\epsilon}{4H^2}\begin{bmatrix} -1 & -2 & -1 \\ -2 & 4 & -2 \\ -1 & -2 & -1 \end{bmatrix} \tag{5.55}$$

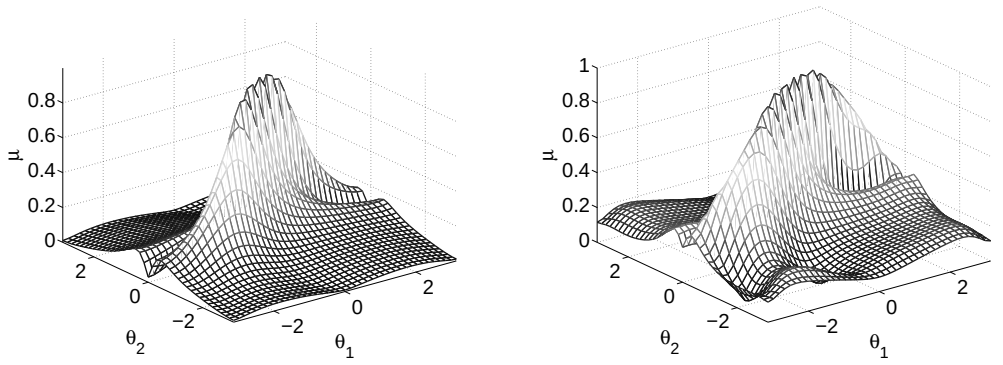
Simplified two-grid analysis (similar to 5.44) to the Galerkin coarse grid operator, yields a convergence factor of zero. This implies that at least the very low frequency components are correctly approximated by the coarse grid operator. The same result can be obtain by performing a Taylor series expansion of the improved inviscid coarse grid operator for $a > 0$ and $b > 0$:

$$\begin{aligned} L^H \langle u \rangle &= a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \frac{H}{4}\left(a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}\right) + \mathcal{O}\left\langle H^2 \right\rangle \\ &= a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} - \frac{h}{2}\left(a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2}\right) + \mathcal{O}\left\langle H^2 \right\rangle \end{aligned} \tag{5.56}$$

This is exactly the same as for the fine grid operator (equation 5.45). Thus also Galerkin coarsening is able to solve the problems associated with direct coarsening of the first-order upwind operator.

Smoothing for Galerkin coarsening of first-order upwind schemes

Local mode analysis for symmetric Kaczmarz line relaxation applied to the Galerkin coarsening of the first-order upwind discretization of the convection-diffusion equation is shown in figure 5.5.1, for $a = \cos\pi/8$, $b = \sin\pi/8$ and $\epsilon = 0\,\mathrm{m}$. Due to the Galerkin coarsening, the operators, and thus also the LMA analysis vary between different grid levels. For coarser grids the performance of Kaczmarz relaxation deteriorates (see table 5.2 for smoothing factors for more cases). Therefore simple V-cycles will not be sufficient to obtain mesh-independent convergence rates.



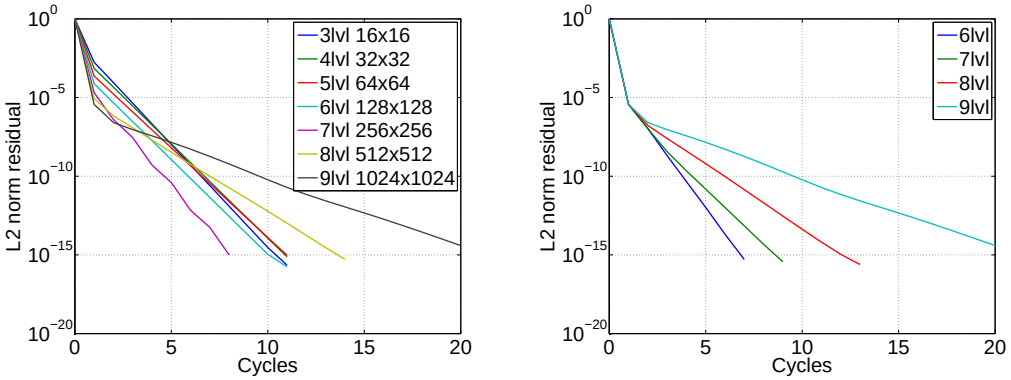
(a) Grid level 2 (equation 5.55)

(b) Grid level 3

Figure 5.10: Convection-diffusion equation: LMA analysis for symmetric line Kaczmarz relaxation applied to the Galerkin coarsened, first-order upwind discretization of the convective terms in the hyperbolic limit ($a = \cos\frac{\pi}{8}$, $b = \sin\frac{\pi}{8}$ and $\epsilon = 0\,\mathrm{m}$)

Grid	$\epsilon = 0\,\mathrm{m}$					$\epsilon = \infty$
level	$\alpha = 0$	$\alpha = \pi\frac{1}{8}$	$\alpha = \pi\frac{1}{4}$	$\alpha = \pi\frac{3}{8}$	$\alpha = \pi\frac{1}{2}$	
1	0	0.497	0.314	0.497	0	0.410
2	0.407	0.548	0.287	0.548	0.407	0.262
3	0.575	0.821	0.617	0.821	0.575	0.209
4	0.621	0.950	0.871	0.950	0.621	0.194
5	0.633	0.987	0.966	0.987	0.633	0.191

Table 5.2: Convection-diffusion equation: Smoothing factors for symmetric line Kaczmarz relaxation applied to the Galerkin coarsened first-order upwind discretization discretization of the convective terms in the hyperbolic limit for different flow angles ($a = \cos\alpha$, $b = \sin\alpha$) and different grid levels and in the elliptic limit for different grid levels



(a) 4x4 as the coarsest grid

(b) 1024x1024 as the target grid

Figure 5.11: Recirculating convection-diffusion problem: L_2-norm of the residual ($\left|r^h\right|_2$) as a function of the number of W(2,1) multi-grid cycles for the first-order upwind discretization of the convective terms using Galerkin coarsening

Results

Results by applying W(2,1)-cycles to the first-order upwind discretization of the recirculating convection-diffusion problem with Galerkin coarsening are shown in figure 5.11. Convergence is satisfactory when not too many grid levels are used (compare to figure 5.8(a)). This is because the performance of the Kaczmarz relaxation deteriorates on coarser grids. Either using a smaller number of grids or an even higher grid parameter (γ) will restore performance. This is shown in figure 5.11(b) where residuals are plotted for different number of levels with a 1024x1024 as the target grid. Convergence increases when a smaller number of levels are used, but this increase comes at the cost of a larger coarsest grid problem. This problem is solved exactly by a direct solver, which requires roughly $\mathcal{O}\left(n^4\right)$ operations (with n the number of grid points in one direction). So there is always a trade-off between the number of levels used and the required work to solve the problem on the coarsest grid.

5.5.2 Higher-order

Higher-order discretizations of the convection-diffusion problem require an even more sophisticated approach. When full weighting and bi-linear interpolation are used as restriction and interpolation operators, the coarse grid operator of the κ-scheme discretization of the convective terms in the convection-diffusion equation (equation 5.34) becomes: (for $\kappa = 0$, $a > 0$ and $b > 0$)

$$L^H = \frac{1}{128H}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ a & -11a+7b & 3a+42b & 7a+7b & 0 \\ 6a & -66a+3b & 18a+18b & 42a+3b & 0 \\ a & -11a-11b & 3a-66b & 7a-11b & 0 \\ 0 & b & 6b & b & 0 \end{bmatrix} + \frac{\epsilon}{4H^2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & -2 & 4 & -2 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.57}$$

Simplified two-grid analysis of the (inviscid) Galerkin coarse grid operator now yields a convergence factor of zero, except for the case $a = b$, for which a factor of 0.5 is obtained. This can also be seen by considering at a Taylor series expansion and rotating the result to the (ξ, η) axis system, with ξ in streamwise direction and η perpendicular to the flow. Now considering only the terms involving derivatives in the η direction yields:

$$-\epsilon\frac{\partial^2 u}{\partial \eta^2} - \frac{H^2}{48}ab\frac{a^2-b^2}{\left(a^2+b^2\right)^{\frac{3}{2}}}\frac{\partial^3 u}{\partial \eta^3} + \left(\frac{H^3}{32}ab\frac{a^3+b^3}{\left(a^2+b^2\right)^2} - \frac{H^2\epsilon}{12}\frac{a^4+3a^2b^2+b^4}{\left(a^2+b^2\right)^2}\right)\frac{\partial^4 u}{\partial \eta^4} + \ldots \tag{5.58}$$

while on the fine grid it yields:

$$-\epsilon\frac{\partial^2 u}{\partial \eta^2} - \frac{h^2}{12}ab\frac{a^2-b^2}{\left(a^2+b^2\right)^{\frac{3}{2}}}\frac{\partial^3 u}{\partial \eta^3} + \left(\frac{h^3}{8}ab\frac{a^3+b^3}{\left(a^2+b^2\right)^2} - \frac{h^2\epsilon}{12}\frac{a^4+b^4}{\left(a^2+b^2\right)^2}\right)\frac{\partial^4 u}{\partial \eta^4} + \ldots \tag{5.59}$$

For small ϵ,the second terms will dominate the flow. These terms are equal (for $H = 2h$), so the coarse grid discretization satisfies the requirements. But for the case $a = b$ the second term will be zero so the third term dominates. However, the third term for both Taylor series expansions differs by a factor of 2, so for these types of flow the convergence is limited by a factor of 0.5. Using a restriction operator with a higher high-frequency order will solve this problem [16]. An example of such a restriction is:

$$I_h^H = \frac{1}{16}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{5.60}$$

Using this restriction operator for the Galerkin coarsening yields:

$$L^H = \frac{1}{1024H}\begin{bmatrix} 0 & 3b & 10b & 3b & 0 \\ 3a & -108a+84b & 18a+280b & 84a+84b & 3a \\ 10a & -360a+18b & 60a+60b & 280a+18b & 10a \\ 3a & -108a-108b & 18a-360b & 84a-108b & 3a \\ 0 & 3b & 10b & 3b & 0 \end{bmatrix} + \frac{\epsilon}{128H^2}\begin{bmatrix} 0 & -3 & -10 & -3 & 0 \\ -3 & -24 & -10 & -24 & -3 \\ -10 & -10 & 200 & -10 & -10 \\ -3 & -24 & -10 & -24 & -3 \\ 0 & -3 & -10 & -3 & 0 \end{bmatrix} \quad (5.61)$$

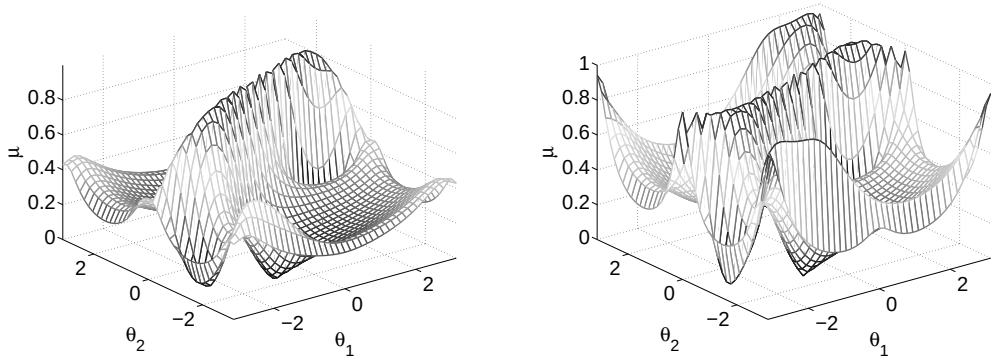
Again applying a Taylor series expansion in the (ξ, η) coordinate system yields: (for terms involving derivatives in the η-direction only)

$$-\epsilon\frac{\partial^2 u}{\partial\eta^2} - \frac{H^2}{48}ab\frac{a^2-b^2}{(a^2+b^2)^{\frac{3}{2}}}\frac{\partial^3 u}{\partial\eta^3} + \left(\frac{H^3}{64}ab\frac{a^3+b^3}{(a^2+b^2)^2} - \frac{H^2\epsilon}{24}\frac{5a^4+9a^2b^2+5b^4}{(a^2+b^2)^2}\right)\frac{\partial^4 u}{\partial\eta^4} + \dots \quad (5.62)$$

For this coarse grid operator also the third term is equal to that of the fine grid operator, so it is an accurate representation of the fine grid problem. This can also be seen by the simplified two-grid convergence, which gives a factor of zero for all a and b.

Smoothing for Galerkin coarsening of κ-schemes

Smoothing of Galerkin coarsened κ-schemes can be performed by applying Kaczmarz line relaxation. LMA results are shown in figure 5.5.2 and table 5.3 for different levels and flow directions. Kaczmarz relaxation is not a good smoother for these types of discretization. High-frequency error components cannot be removed efficiently, especially when dealing with coarser grids.



(a) Grid level 2 (equation 5.61)

(b) Grid level 3

Figure 5.12: Convection-diffusion problem: LMA analysis for symmetric line Kaczmarz relaxation applied to the Galerkin coarsened κ-scheme discretization of the convective terms in the hyperbolic limit ($a = \cos\frac{\pi}{8}$,$b = \sin\frac{\pi}{8}$ and $\epsilon = 0\,\mathrm{m}$)

Results

Results of applying W(2,1)-cycles to the κ-scheme based discretization of the recirculating convection-diffusion problem with Galerkin coarsening are shown in figure 5.13. When only a few levels are employed convergence is reasonable, but when more levels are used convergence deteriorates. When

Grid level	$\epsilon = 0\,\mathrm{m}$ $\alpha = 0$	$\alpha = \pi\frac{1}{8}$	$\alpha = \pi\frac{1}{4}$	$\alpha = \pi\frac{3}{8}$	$\alpha = \pi\frac{1}{2}$	$\epsilon = \infty$
1	0	0.862	0.649	0.862	0	0.410
2	0.761	0.974	0.900	0.974	0.761	0.193
3	0.888	0.998	0.992	0.998	0.888	0.466
4	0.908	1.000	1.000	1.000	0.908	0.535
5	0.913	1.000	1.000	1.000	0.913	0.552

Table 5.3: Convection-diffusion problem: Smoothing factors for symmetric line Kaczmarz relaxation applied to the Galerkin coarsened κ-scheme discretization of the convective terms in the hyperbolic limit for different angles ($a = \cos\alpha$, $b = \sin\alpha$) and different grid levels and in the elliptic limit for different grid levels

more than 5 levels are used even divergence is observed (not plotted in the figures). Geometric multi-grid fails for these cases because the coarse grid operators, that accurately represent the flow problem on the target grid, cannot be smoothed with the smoothing operators considered in this thesis.

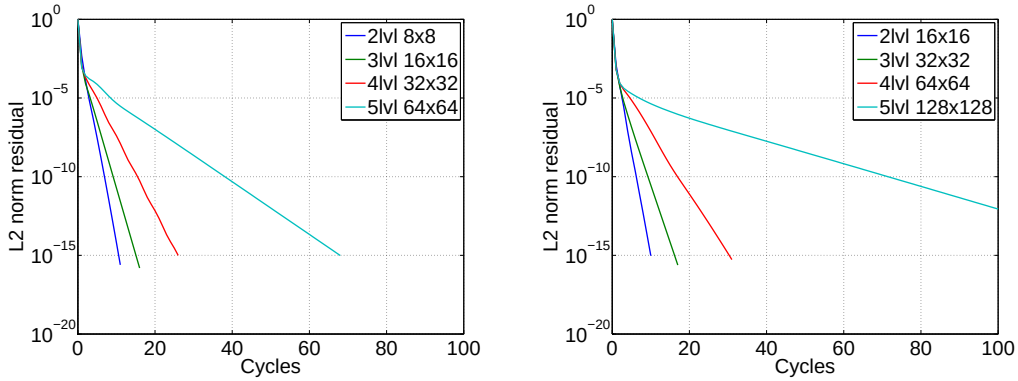
5.6 Conclusion

In this chapter it has been shown that geometric multi-grid can be used in certain cases for the convection-diffusion equation. The first and most simple case is for problems in which diffusion is dominant over the convection. Then coarse grid operators originating from the direct discretization of the partial differential equation on the coarse grid can accurately describe the fine grid problem and excellent multi-grid performance is achieved.

For convection dominated flow standard coarse grid operators lead to difficulties. Smoothers for discretizations of this kind of flows are not always capable of reducing error components which are smooth in the direction of the flow, but oscillatory perpendicular to the flow. Standard coarse grid operators are not able to accurately represent these components on the coarse grid, since they show different behavior in the direction perpendicular to the flow than the fine grid operator. For the first-order discretization a higher artificial viscosity is observed, while for the higher-order discretizations higher-order terms can have different factors on different grids. The solution therefore is to construct coarse grid operators which represent the fine grid operator more accurately. This can be achieved either by using a combination of the conventional discretization and a higher-order discretization, or by using Galerkin coarsening.

When the first-order upwind discretization on the target grid is used, a combination of the first-order upwind and a κ-scheme based discretization can be used on coarser grids. This combination can lead to the same artificial viscosity as observed on the fine grid. However, now a smoother is required which is able to smooth both the high and low-order discretizations, such as the κ-smoother. Good grid-independent convergence is observed for all values of ϵ when using W(2,1)-cycles.

The other possibility is to use Galerkin based coarsening. In this case operators on each grid are calculated based on the fine grid operator and the restriction and interpolation operators. The order of restriction and interpolation needs to be high enough to accurately represent the problematic components. Operators on each grid become different, so more generally applicable smoothers are required. Kaczmarz smoothers are guaranteed to have some smoothing capabilities, but are usually quite slow to converge. For convection dominated flow these smoothers are especially slow when more coarser grids are used. Therefore a higher cycle parameter (γ) is required to achieve good performance. For the first-order discretizations good convergence is observed with W(2,1)-cycles if not too many grid levels are used (up to 7 levels). For higher-order discretization smoothing on the coarse grid becomes more difficult and good convergence is only observed when using up to four grid levels.



(a) 4x4 as the coarsest grid

(b) 8x8 as the coarsest grid

(c) 16x16 as the coarsest grid

(d) 32x32 as the coarsest grid

Figure 5.13: Recirculating convection-diffusion problem: L_2-norm of the residuals ($\left|r^h\right|_2$) as a function of the number of W(2,1) multi-grid cycles for the κ-scheme discretization of the convective terms using Galerkin coarsening

Chapter 6

AMG applied to the Poisson equation

In this chapter different variants of Poisson-like equations with Dirichlet boundary conditions (equation 4.1) will be solved using AMG. Geometric multi-grid techniques, with correct smoothers, show better performance for most Poisson types of equations due to the expensive start-up phase of AMG. However, these types of problems serve as a good example to demonstrate the main principles of the AMG method.

In all examples discussed below Gauß-Seidel relaxation is performed, for which first all coarse grid points and then all fine grid points are relaxed. This relaxation is also referred to as CF-Gauß-Seidel relaxation.

6.1 5-point discretization of the Poisson problem

In this section AMG is used to solve the standard Poisson equation:

$$\begin{array}{rcll} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} & = & f(x,y) & (x,y) \in \Omega \\ u & = & 0 & (x,y) \in \partial\Omega \end{array} \tag{6.1}$$

with a standard 5-point discretization:

$$L^h = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \tag{6.2}$$

Eliminating the Dirichlet boundaries, leads to the systems of equations:

$$A^h u^h = f^h \tag{6.3}$$

with:

$$A = \frac{1}{h^2} \begin{bmatrix} D & I & 0 & \dots & 0 \\ I & D & I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & I & D & I \\ 0 & \dots & 0 & I & D \end{bmatrix} \tag{6.4}$$

,D the matrix corresponding to a single row of points:

$$D = \begin{bmatrix} -4 & 1 & 0 & \dots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -4 & 1 \\ 0 & \dots & 0 & 1 & -4 \end{bmatrix} \tag{6.5}$$

and I the identity matrix.

6.1.1 Grid hierarchy

The first step in the AMG process is to determine the coarse grids. Therefore the coloring algorithm described in section 3.4 is applied to the system of equations resulting from the 5-point discretization of the Poisson equation (equation 6.3). For this example the choice of ϵ_{str} is immaterial, since all points are strongly connected to and dependent on all of their four neighbors for any value of ϵ_{str}. Initially the value of λ (see equation 3.36) is 4 at interior points, 3 at boundary points and 2 at corner points (see figure 6.1(a)). The maximum value $\lambda = 4$ is found at several locations and one of them, i.e. the most bottom left point, is selected as a starting point (a different starting point can lead to a different grid). This point becomes a coarse grid point (indicated by a solid circle in figure 6.1(b)). Now each point that is strongly influenced by this point and has not yet been decided as a coarse or fine grid point (i.e. $S_i^T \cap U$), is turned into a fine grid point (indicated by an open circle). Before a new point can be chosen as a coarse grid point, the values of λ need to be reevaluated (see figure 6.1(b)). However, there is no need to reevaluate λ at each point, since only local changes are applied. Again the point with maximum value of λ is turned into a coarse grid point as can be seen in figure 6.1(c). This process is repeated until all points are defined as either fine or coarse. The final result is visible in figure 6.1(i). It is noteworthy that the obtained coarse grid is the same as the red-black coarsening variant used in geometric multi-grid. Also it should be noted that at several occasions in the process there is no unique point with the maximum value of λ. Different coloring can be created when choosing different points as the new coarse grid points.

After all points have been decided as either coarse or fine grid points, the interpolation operator for each fine grid point i has to be calculated. For the coloring depicted in figure 6.1 there is no difference between direct (see section 3.3.1) and standard interpolation (see section 3.3.2), since there are no fine grid points that have strong influence on gridpoint i (i.e. $F_i^s = \emptyset$). The weights for points not close to boundaries become:

$$w_{i,j} = -\frac{A_{i,j}}{A_{i,i}} \frac{\sum_{k \in N_i} A_{i,k}}{\sum_{k \in C_i^s} A_{i,k}} = \frac{A_{i,j}}{A_{i,i}} \text{ with } j \in C_i^s \tag{6.6}$$

Or in stencil notation:

$$I_H^h = \frac{1}{4} \begin{bmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{bmatrix} \tag{6.7}$$

Thus a fine grid point is interpolated as the average of its four neighbors. The last step in the coarsening process is the calculation of the coarse grid matrix using the Galerkin principle. For interior points the stencil becomes:

$$L^H = \frac{1}{8h^2} \begin{bmatrix} & & -1 & & \\ & -2 & & -2 & \\ -1 & & 12 & & -1 \\ & -2 & & -2 & \\ & & -1 & & \end{bmatrix} \tag{6.8}$$

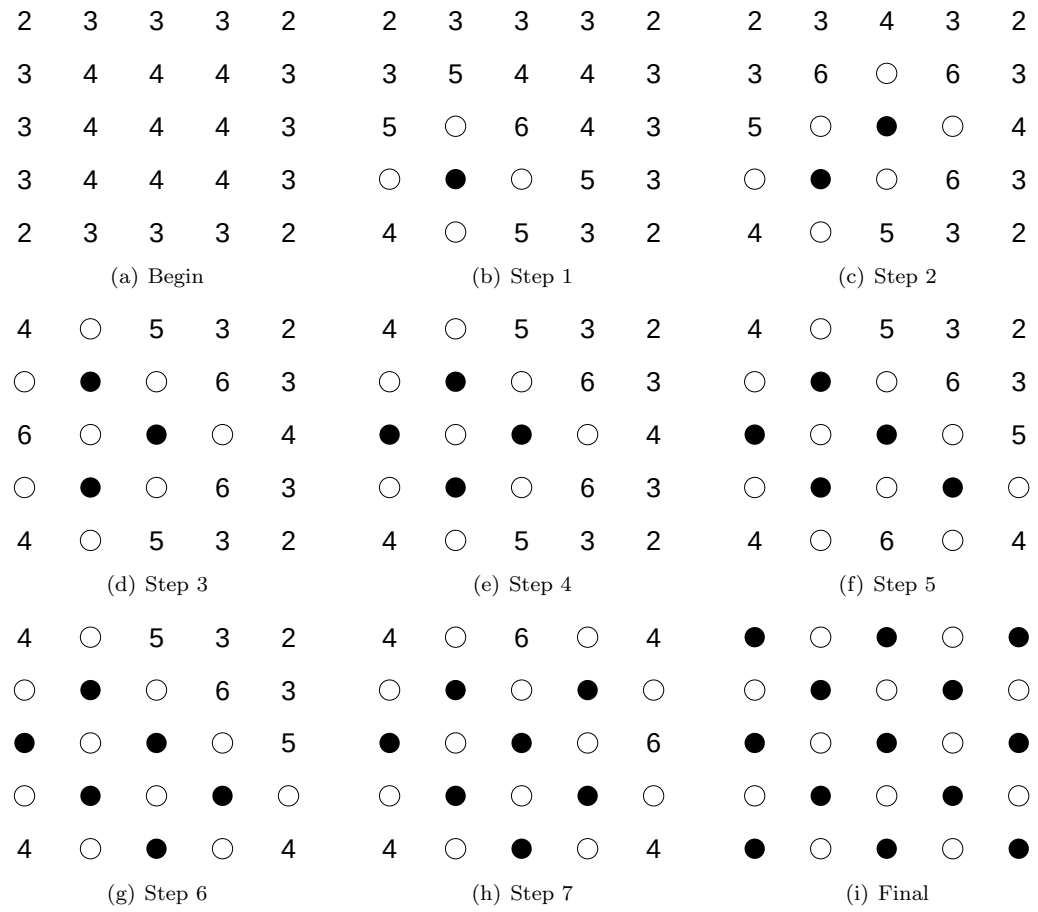


Figure 6.1: Poisson problem: AMG coloring for the 5-point discretization, solid circle=coarse grid points, open circle=fine grid points, number=λ-value of undecided points

Using Taylor series expansion it can be shown that this stencil is indeed a second-order accurate discretization of the Poisson equation:

$$L^H \langle u \rangle = -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \mathcal{O}\left(h^2\right) \tag{6.9}$$

After the coarse grid matrix has been calculated additional coarsening steps can be performed. In figure 6.2 the first two-levels of coarsening are shown for the 5-point Laplacian on a 33x33 grid. The first coarsening step corresponds to red-black coarsening just as explained above. On the second level operators correspond to 9-point operators and a different coarsening to the third level is observed, because each point is now strongly influenced by and dependant on its eight neighbors (at least for $\epsilon_{str} < 0.5$). The second coarsening step corresponds to the standard geometrical multi-grid coarsening, but now at an angle of 45 degrees.

6.1.2 Results

To illustrate the performance of the algorithm V(1,1) AMG cycles are applied to the 5-point discretization of the Poisson problem, with a right hand side corresponding to a known analytical solution:

$$\begin{aligned} f(x,y) &= -8\pi^2 \sin(2\pi x)\sin(2\pi y) \\ u(x,y) &= \sin(2\pi x)\sin(2\pi y) \end{aligned} \tag{6.10}$$

Convergence results are shown in figure 6.3. For each target grid size coarsening has been performed until only maximal 20 grid points remain. On the coarsest level the system of equations is solved directly by Gaussian elimination. Extremely fast convergence is observed when only a few levels are used. This is because on the first few levels, the coarsening algorithm produces regular grids. These regular grids have the special property that CF-Gauß-Seidel relaxation has superb smoothing properties. When using more grid levels, the coloring algorithm produces grids which are not regular anymore, leading to a slightly lower convergence rate.

6.2 9-point discretization of the Poisson problem

A disadvantage of using AMG to solve the Poisson problem using a 5-point Laplacian is in the first coarsening process, the number of unknowns is only halved (compared to one fourth for geometric multi-grid). Even worse, the number of nonzero matrix indices is only multiplied with a factor of 0.9 (from 5 point stencils to 9 points stencils), compared to 0.25 for geometric multi-grid. One possible modification is not to use the 5-point Laplacian, but the 9-point Laplacian:

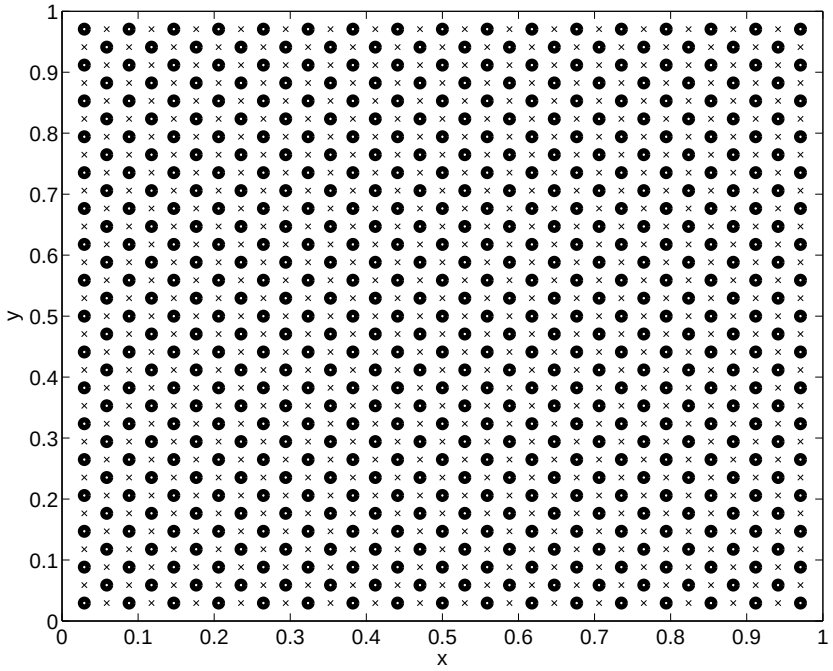
$$\frac{1}{3h^2}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{6.11}$$

For this operator the system of equations with eliminated Dirichlet boundaries becomes:

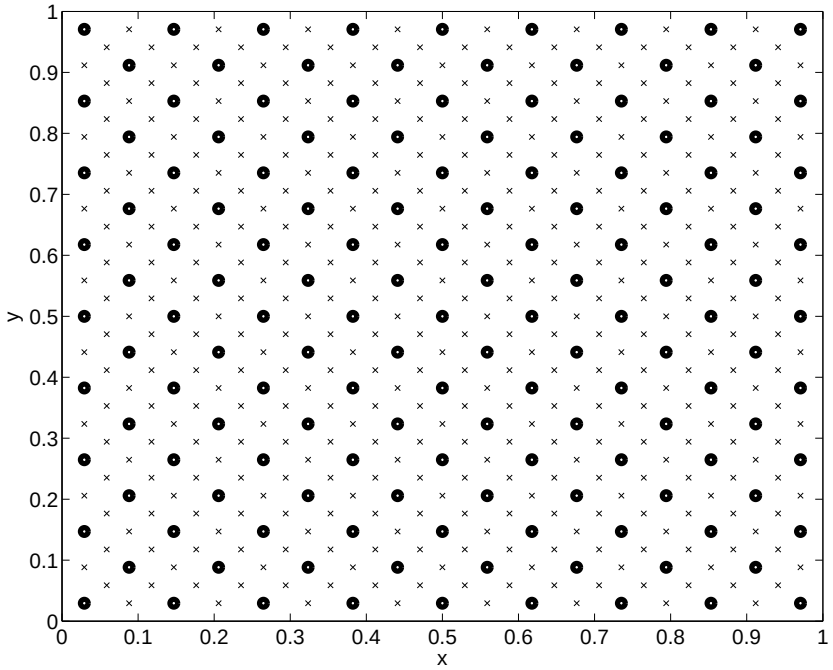
$$A^h u^h = f^h \tag{6.12}$$

with

$$A = \frac{1}{3h^2}\begin{bmatrix} D & O & 0 & \dots & 0 \\ O & D & O & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & O & D & O \\ 0 & \dots & 0 & O & D \end{bmatrix} \tag{6.13}$$

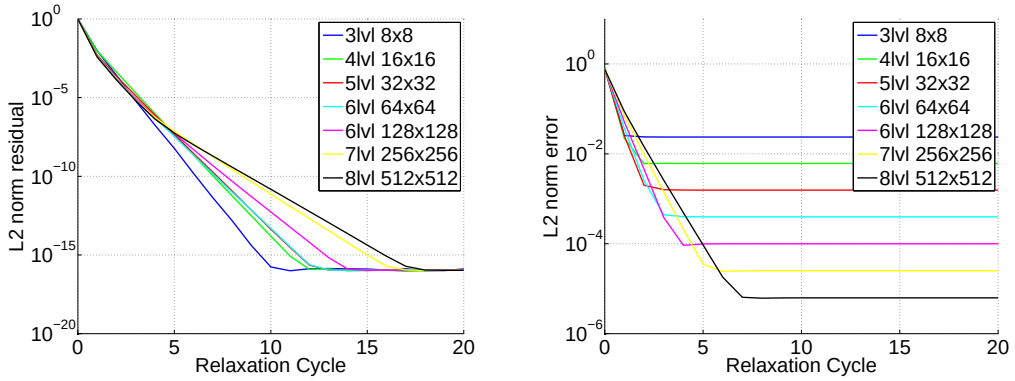


(a) Level 1



(b) Level 2

Figure 6.2: Poisson problem: AMG, two-levels of coloring for the 5-point discretization on a 34x34 grid, solid circle=coarse grid points, cross=fine grid points



(a) Residual (b) Error

Figure 6.3: Poisson problem: AMG, L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of V(1,1) AMG cycles for different grid sizes, using CF-Gauß-Seidel relaxation on the 5-point discretization

D the matrix corresponding to a single row of points:

$$D = \begin{bmatrix} -8 & 1 & 0 & \dots & 0 \\ 1 & -8 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -8 & 1 \\ 0 & \dots & 0 & 1 & -8 \end{bmatrix} \tag{6.14}$$

and

$$O = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 1 & 1 \\ 0 & \dots & 0 & 1 & 1 \end{bmatrix} \tag{6.15}$$

6.2.1 Coloring

The first steps of the coloring process together with two possible final colorings for this discretization of the Poisson problem are presented in figure 6.4. After the second step there are three points with the maximum value of λ. Anyone of them could be used as the next fine grid point. When either the top or the bottom point is selected as the next point, the final coloring will be as illustrated in 6.4(d). When the center point is selected as the next point, the final coloring results is as shown in 6.4(e). Both colorings are valid as a coarse grid, but may give different convergence behavior. With this operator, even after the first coarsening step, the number of unknowns has been reduced by a factor of 4 (when large target grids are used). Actually for the first coloring step, the coarse grid produced is exactly the standard full coarsening that one would use in a geometric multi-grid approach. This is also illustrated in figure 6.5 where the first two-levels of coloring for the 9-point Laplacian on a target 32x32 grid are shown. Here both coloring steps produce the same coarsening as used in geometric multi-grid.

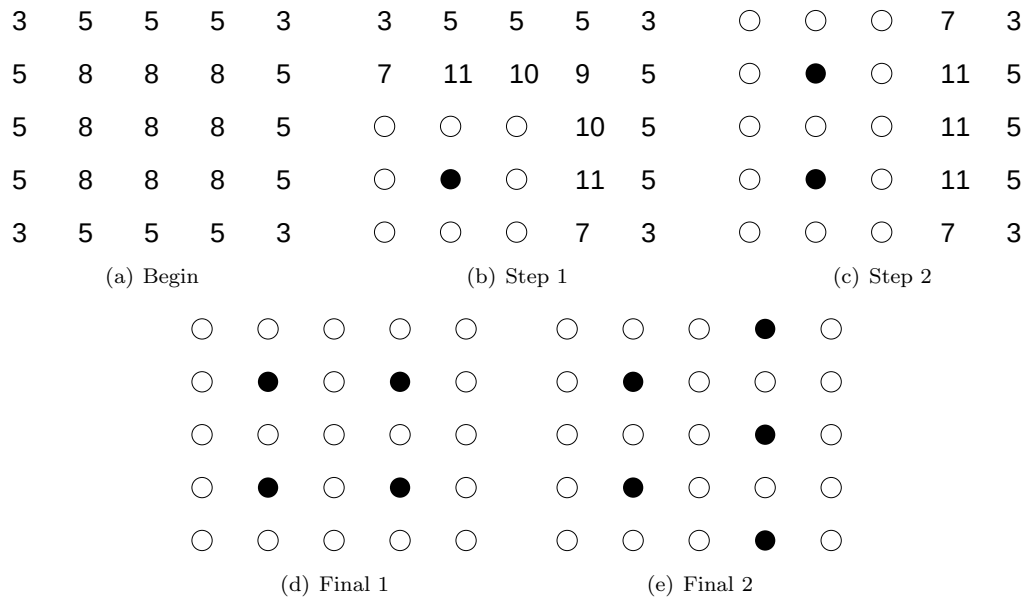


Figure 6.4: Poisson problem: AMG coloring for the 9-point discretization, solid circle=coarse grid points, open circle=fine grid points, number=λ-value of undecided points

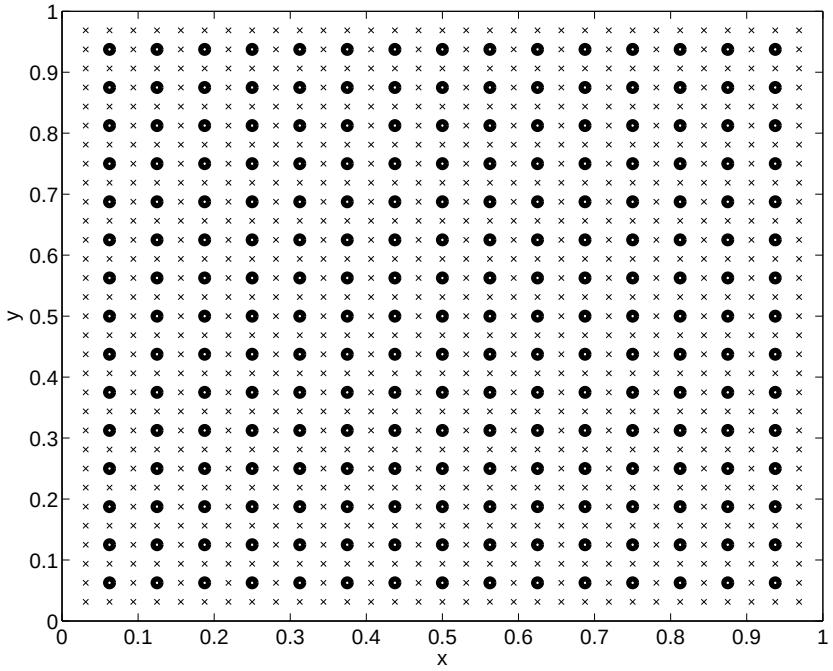
6.2.2 Results

AMG cycles have been applied to the 9-point discretization of the Poisson problem with the right hand side as defined in equation 6.10. Convergence results are shown in figure 6.6. Good convergence is observed for all target grid sizes.

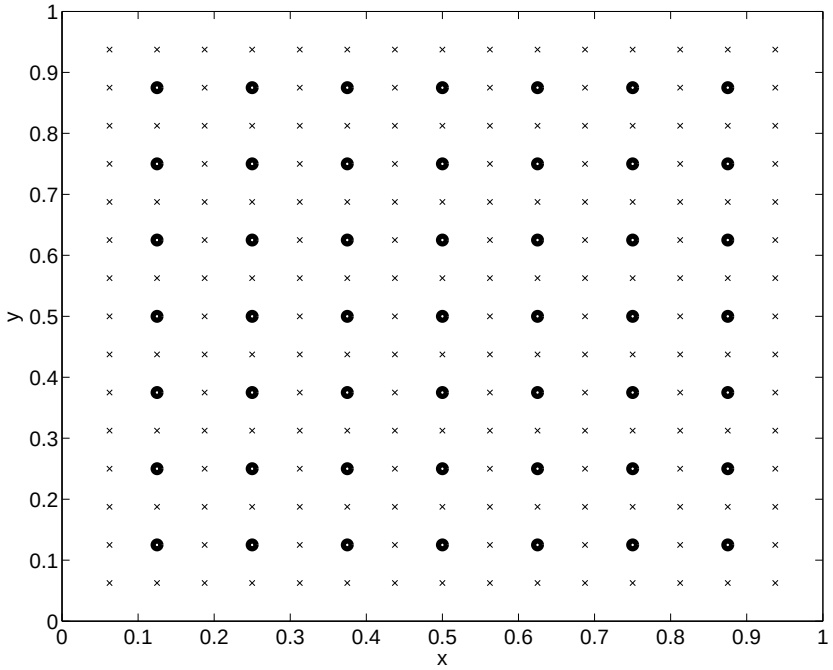
Information about the different grids for a target 256x256 grid are shown in table 6.1 for both the 5- and the 9-point discretizations. The reduction of the number of unknowns by only a factor of 2 for the first coarsening process of the 5-point Laplacian is clearly visible as well as the reduction in the number of non zero matrix indices by a factor of 0.9. Further coarsening steps do reduce the number of unknowns by a factor of 4. Coloring for the 9-point Laplacian immediately reduces the number of unknowns by a factor 4. A good estimate of the total work required for one multi-grid cycle can be obtained by multiplying the number of non zero matrix indices on each grid with the number of relaxations performed on that grid level and summing over all grid levels. For V(1,1) cycles this is $90 \cdot 10^3$ for the 5-point stencil and $112 \cdot 10^3$ for the 9-point stencil, so multi-grid cycles for the 5-point discretization are somewhat faster. Therefore the 5-point discretization is used throughout the rest of this thesis.

Level	5-Point		9-Point	
	unknowns	Nonzero indices	unknowns	Nonzero indices
7	65536	326656	65536	586756
6	32768	292866	16384	380870
5	8192	184082	4096	126366
4	2239	77455	672	18536
3	824	12888	103	2563
2	67	1473	15	185
1	14	156		

Table 6.1: Poisson problem: AMG grid information for the 5- and 9-point discretization on a initial 256x256 grid

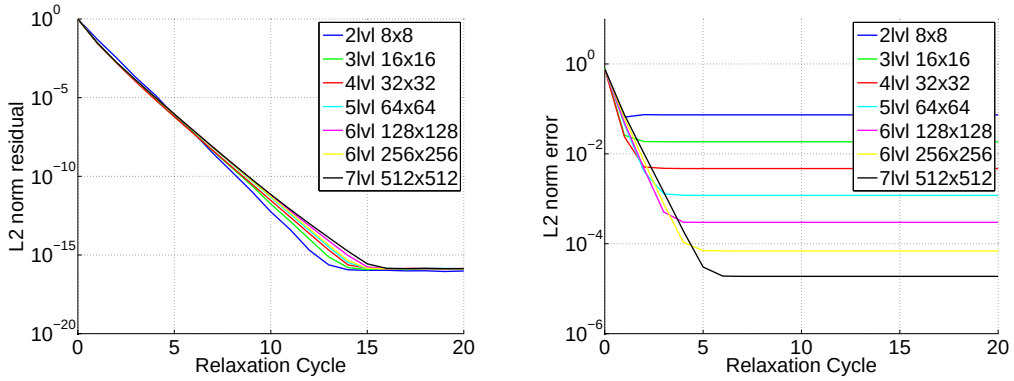


(a) Level 1



(b) Level 2

Figure 6.5: Poisson problem: AMG, two-levels of coloring for the 9-point discretization on a 32x32 grid, solid circle=coarse grid points, cross=fine grid points



(a) Residual (b) Error

Figure 6.6: Poisson problem: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of V(1,1) AMG cycles for different grid sizes, using CF-Gauß-Seidel relaxation on the 9-point discretization

6.3 Anisotropic Poisson equation

In this section AMG is used to solve the anisotropic Poisson equation:

$$\begin{array}{rcll} \frac{\partial^2 u}{\partial x^2} + \frac{\partial}{\partial y}\left(\epsilon\left(x,y\right)\frac{\partial u}{\partial y}\right) & = & f\left(x,y\right) & \left(x,y\right) \in \Omega \\ u & = & 0 & \left(x,y\right) \in \partial\Omega \end{array} \tag{6.16}$$

with a standard 5-point discretization:

$$L^h_{i,j} = \frac{1}{h^2}\begin{bmatrix} & \epsilon\left(x, y-\frac{h}{2}\right) & \\ 1 & -2-\left(\epsilon\left(x, y-\frac{h}{2}\right)+\epsilon\left(x, y+\frac{h}{2}\right)\right) & 1 \\ & \epsilon\left(x, y+\frac{h}{2}\right) & \end{bmatrix} \tag{6.17}$$

Eliminating the Dirichlet boundary conditions, leads to the systems of equations:

$$A^h u^h = f^h \tag{6.18}$$

with:

$$A = \frac{1}{h^2}\begin{bmatrix} D & I & 0 & \dots & 0 \\ I & D & I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & I & D & I \\ 0 & \dots & 0 & I & D \end{bmatrix} \tag{6.19}$$

Here D corresponds to the matrix of a single row of points:

$$D = \begin{bmatrix} -2-\epsilon^- -\epsilon^+ & \epsilon^+ & 0 & \dots & 0 \\ \epsilon^- & -2-\epsilon^- -\epsilon^+ & \epsilon^+ & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \epsilon^- & -2-\epsilon^- -\epsilon^+ & \epsilon^+ \\ 0 & \dots & 0 & \epsilon^- & -2-\epsilon^- -\epsilon^+ \end{bmatrix}$$

$$\epsilon^- = \epsilon\left(x, y - \frac{h}{2}\right)$$

$$\epsilon^+ = \epsilon\left(x, y + \frac{h}{2}\right) \tag{6.20}$$

and I the identity operator.

6.3.1 Coloring

In geometric multi-grid semi-coarsening in the direction of strong coupling could be used to handle anisotropic Poisson problems. This is the same technique as used by AMG solvers. The first two steps of the coloring process along with two possible final colorings for a constant $\epsilon = 0.01$ are shown in figure 6.7. There is not always just one point with the maximum value of λ, so just as for the 9-point Laplacian multiple allowable coarsenings exist.

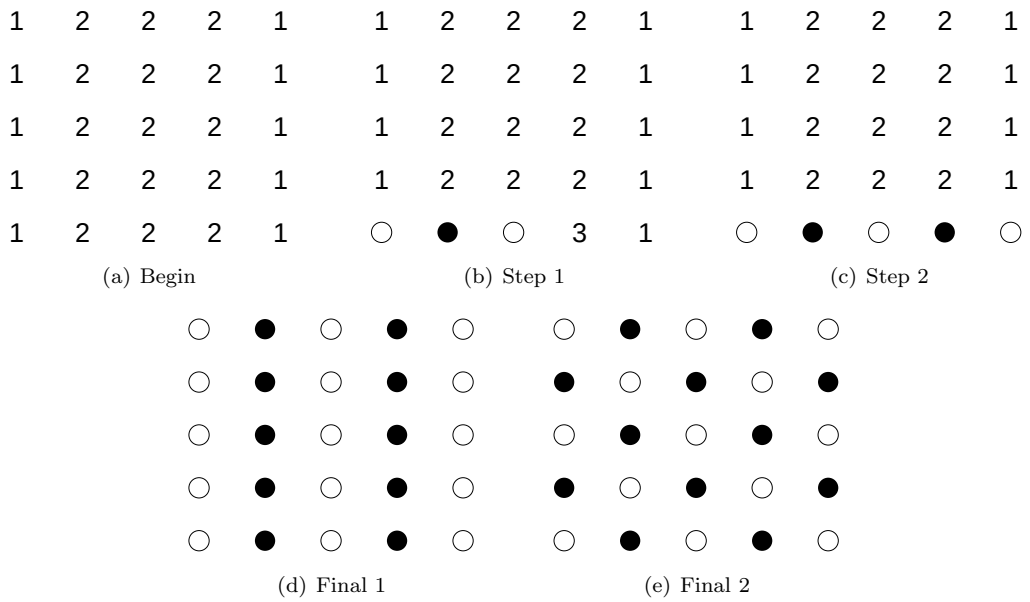
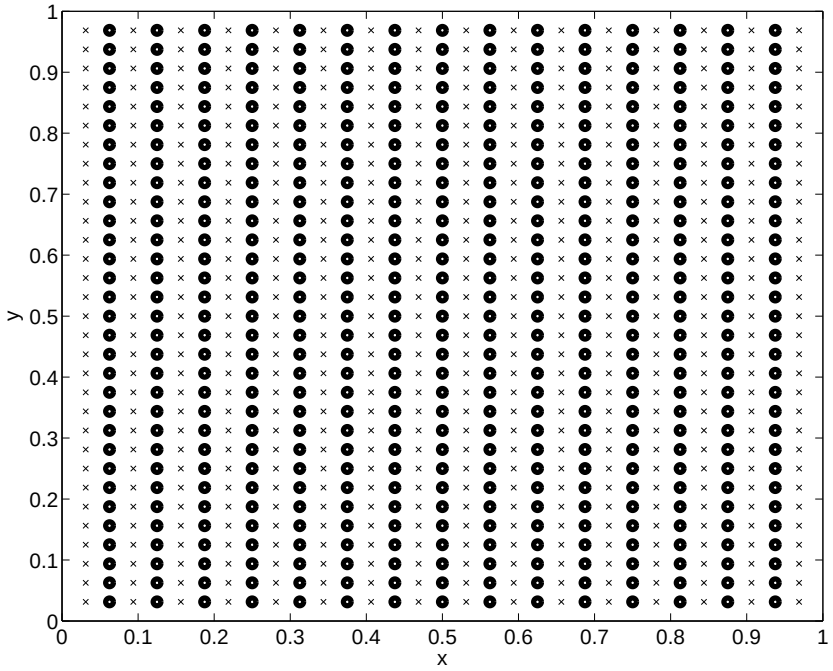


Figure 6.7: Anisotropic Poisson equation: AMG coloring for the 5-point discretization with $\epsilon = 10^{-2}$, solid circle=coarse grid points, open circle=fine grid points, number=λ-value of undecided points

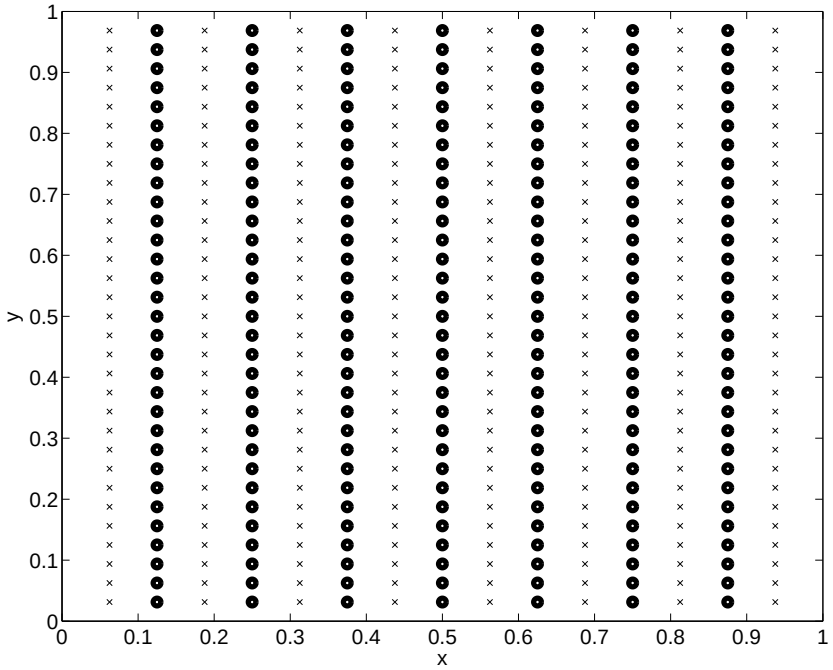
The first two-levels of coarsening of the anisotropic Poisson problem with $\epsilon = 0.01$ on a target 31x31 grid are shown in figure 6.8. The second step in the coarsening is just as the first step a semi-coarsening step. Semi-coarsening decreases the anisotropy of the operator and is continued until isotropy is regained.

In figure 6.9 the second coloring step is shown for the anisotropic Poisson problem on a target 64x64 grid with varying ϵ:

$$\epsilon\,(x, y) = 10^{3cos(2\pi x)cos(2\pi y)} \tag{6.21}$$



(a) Level 1



(b) Level 2

Figure 6.8: Anisotropic Poisson equation: AMG two-levels of coloring for the 5-point discretization on a target 31x31 grid with $\epsilon = 10^{-2}$, solid circle=coarse grid points, cross=fine grid points

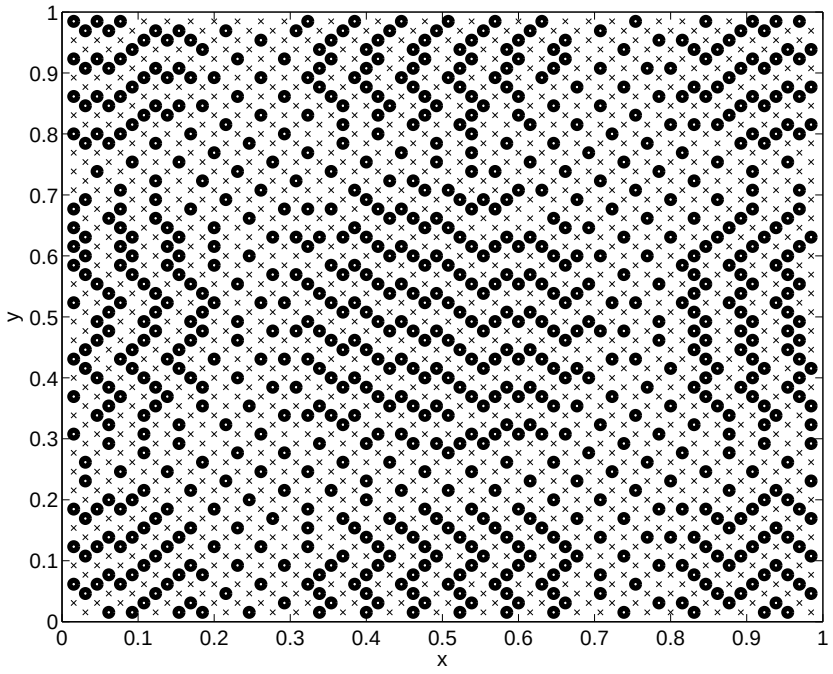


Figure 6.9: Anisotropic Poisson equation: AMG coloring on the second level for the 5-point discretization on a target 64x64 grid with varying ϵ, solid circle=coarse grid points, cross=fine grid points

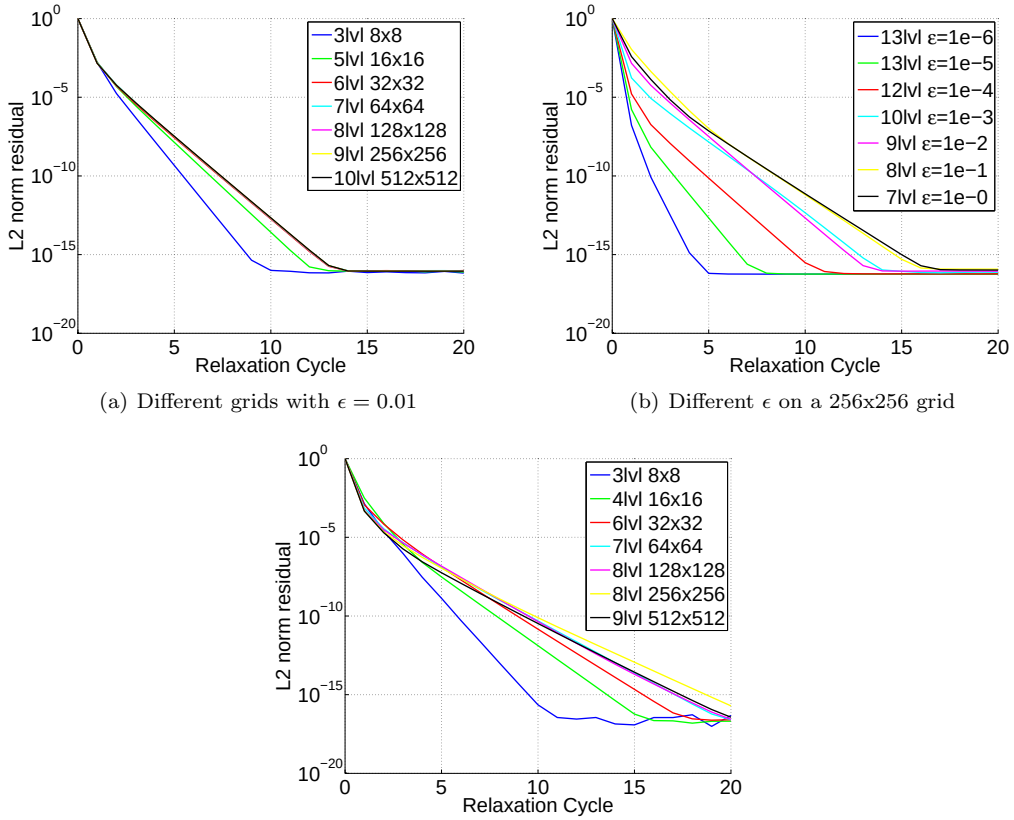
In zones where $\epsilon \gg 1$ (i.e. in the corners and in the center) coupling in the y-direction is strong and the grid is coarsened in y-direction only. In zones where $\epsilon \ll 1$ coupling in the x-direction is strong so the grid is only coarsened in the x-direction.

6.3.2 Results

Convergence results by applying V(1,1) AMG cycles to the anisotropic Poisson problem are shown in figure 6.10. Good grid-independent convergence is observed in figure 6.10(a), where the L_2-norm of the residual is plotted as a function of the number of V(1,1) cycles for different target grids with constant $\epsilon = 0.01$. In figure 6.10(b) residuals are plotted as a function of the number of V(1,1) cycles on a target 256x256 grid for different values of ϵ. For low values of ϵ convergence is better, but more grids are required to obtain a coarsest grid with less than 20 points. This is because the lower ϵ is, the longer semi-coarsening is used (which only reduces the number of unknowns by a factor of 2 per coarsening instead of a factor of 4). Convergence for high values of ϵ is identical to convergence for low values of ϵ. In figure 6.10(c) the L_2-norm of the residual is plotted for varying ϵ for different target grids. Also for this case good convergence is observed for all grids.

6.4 Conclusion

In this chapter it has been shown that algebraic multi-grid techniques are able to solve Poisson and Poisson-like problems very efficiently. In all cases convergence rates of roughly 0.1 are obtained for all problems considered. As far as convergence rate is concerned AMG techniques work just as good as geometric multi-grid techniques. AMG, however, has the advantage that it is able to



(a) Different grids with $\epsilon = 0.01$

(b) Different ϵ on a 256x256 grid

(c) Different grids with varying ϵ

Figure 6.10: Anisotropic Poisson equation: L_2-norm of the residual ($\left|r^h\right|_2$) as a function of the number of V(1,1) AMG cycles for the 5-point discretization on different grids (with constant $\epsilon = 0.01$), for different values of ϵ (on a constant 256x256 grid) and for varying ϵ on different grids

adapt itself to different problems. Therefore no adjustments to the solver are required to solve other partial differential equations. This flexibility, however, comes at the disadvantage of a higher computational cost. This increased cost comes from two factors. Firstly for AMG an initialization process is required where the different grids and operators are determined. Secondly for AMG the reduction in the number of unknowns from one grid level to the next varies for different partial differential equations and discretizations, while the reduction in the number of unknowns for geometric multi-grid is always a factor of four. Therefore it is advantageous to use geometric multi-grid techniques whenever possible and use AMG as a black box solver for a wider class of problems.

Chapter 7

AMG applied to the convection-diffusion equation

In this chapter the 2D convection-diffusion equation with different flow characteristics (equation 7.1) will be solved using AMG.

$$-\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial au}{\partial x}+\frac{\partial bu}{\partial y}=f\left(x,y\right) \tag{7.1}$$

In all examples discussed below four-directional relaxation is performed. This relaxation consists of four relaxation sweeps, each starting at a different corner of the domain. Therefore it is necessary to store a list of all points sorted by position.

7.1 Theoretical overview first-order schemes

In this section it is shown how AMG operates on the first-order upwind discretization of the 2D convection-diffusion equation with small diffusion terms ($\epsilon \ll 1$) and constant convection speeds (a and b) on a domain Ω with Dirichlet boundary conditions on inflow boundaries and Neumann boundary conditions on outflow boundaries.

$$\begin{array}{rcll} -\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+a\frac{\partial u}{\partial x}+b\frac{\partial u}{\partial y} & = & f\left(x,y\right) & \left(x,y\right)\in\Omega \\ u & = & u_0 & \left(x,y\right)\in\partial\Omega_{\text{inflow}} \\ \frac{\partial u}{\partial n} & = & 0 & \left(x,y\right)\in\partial\Omega_{\text{outflow}} \end{array} \tag{7.2}$$

The equation at interior points is discretized as:

$$L^h=\frac{1}{h^2}\begin{bmatrix} & -\epsilon+\frac{h}{2}\left(|b|-b\right) & \\ -\epsilon-\frac{h}{2}\left(|a|+a\right) & 4\epsilon+h\left(|a|+|b|\right) & -\epsilon+\frac{h}{2}\left(|a|-a\right) \\ & -\epsilon-\frac{h}{2}\left(|b|+b\right) & \end{bmatrix} \tag{7.3}$$

Boundary conditions are eliminated using the strategy described in section 4.5. Further more positive a and b are assumed, so that the stencil for the discretization becomes:

$$L^h=\frac{1}{h^2}\begin{bmatrix} & -\epsilon & \\ -\epsilon-ah & 4\epsilon+h\left(a+b\right) & -\epsilon \\ & -\epsilon-bh & \end{bmatrix} \tag{7.4}$$

This leads to the system of equations:

$$A^h u^h=f^h \tag{7.5}$$

with:

$$A = \frac{1}{h^2}\begin{bmatrix} D & B & 0 & \dots & 0 \\ O & D & B & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & O & D & B \\ 0 & \dots & 0 & O+B & D \end{bmatrix} \tag{7.6}$$

$$D = \begin{bmatrix} 4\epsilon + h(a+b) & -\epsilon & 0 & \dots & 0 \\ -\epsilon + bh & 4\epsilon + h(a+b) & -\epsilon & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\epsilon + bh & 4\epsilon + h(a+b) & -\epsilon \\ 0 & \dots & 0 & -2\epsilon + bh & 4\epsilon + h(a+b) \end{bmatrix} \tag{7.7}$$

$$O = (-\epsilon - ah)\, I \tag{7.8}$$

$$B = -\epsilon I \tag{7.9}$$

Based on the values of a, b, h and ϵ four different categories of flow can be defined for positive a and b. These different regions are shown in figure 7.1 and described in table 7.1. In the last column of the table the condition is written in a non-dimensional form using the transformation:

$$\begin{aligned} a &= c\cos\alpha \\ b &= c\sin\alpha \\ \epsilon &= \epsilon_* hc \end{aligned} \tag{7.10}$$

Here α is the angle of the flow with respect to the positive x-axis, c the velocity magnitude and ϵ_* the non-dimensional diffusion coefficient. In each region different colorings may be obtained. Therefore the coarsening steps, including the coloring and construction of interpolation and coarse grid operators, are described separately for each region.

Region	Description	Condition	Non-dimensional condition
I	a dominant	$(ah+\epsilon)\,\epsilon_{str} > bh+\epsilon$	$(\cos\alpha+\epsilon_*)\,\epsilon_{str} > \sin\alpha+\epsilon_*$
II	b dominant	$(bh+\epsilon)\,\epsilon_{str} > bh+\epsilon$	$(\sin\alpha+\epsilon_*)\,\epsilon_{str} > \cos\alpha+\epsilon_*$
III	ϵ dominant	$(\max(a,b)\,h+\epsilon)\,\epsilon_{str} < \epsilon$	$(\max(\cos\alpha,\sin\alpha)+\epsilon_*)\,\epsilon_{str} < \epsilon_*$
IV	a and b dominant	Otherwise	

Table 7.1: Convection-diffusion equation: The four different regions in the (α, ϵ_*)-plane for the first-order upwind discretization of the convective terms, with $\epsilon_{str} = 0.2$

7.1.1 Region I and II

The coarsening of region I and II is in principal the same. The only difference is that they are eachothers image with respect to the line $x = y$. Therefore only region I is described in full detail, which is defined by:

$$(ah+\epsilon)\,\epsilon_{str} > bh+\epsilon \tag{7.11}$$

Or equivalently:

$$(\cos\alpha+\epsilon_*)\,\epsilon_{str} > \sin\alpha+\epsilon_* \tag{7.12}$$

The first step in the AMG process is to determine the coarse grids. Therefore the coloring algorithm described in section 3.4 is applied to the system of equations resulting from the first-order upwind discretization of the convection-diffusion equation (7.5). In figure 7.2(a) initial values of λ are plotted. In region I interior points are only influenced by a single upstream point. On

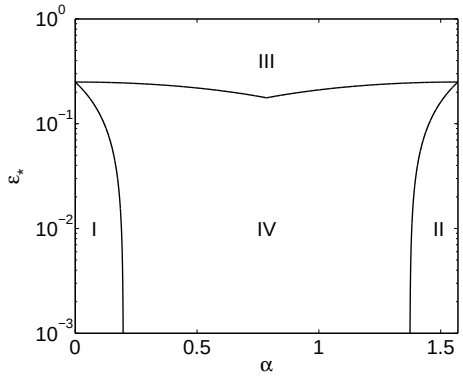


Figure 7.1: Convection-diffusion equation: The four different regions in the (α, ϵ_*)-plane for the first-order upwind discretization of the convective terms, with $\epsilon_{str} = 0.2$

inflow boundary points influence all neighbors and points on outflow boundaries do not influence other points at all.

The first few steps of the coloring process are the same as for the Poisson problem, however, after step 5 (figure 7.2(f)) most points have the same value of λ and anyone of them could be chosen as the next coarse grid point. When points with lower x-coordinates are chosen as the next coarse grid points the final coloring will be as depicted in figure 7.2(g), however, when points with high values of the x-coordinate are preferred the final coloring will be as shown in figure 7.2(h). In both cases points remain with a λ value of zero. These points are chosen as fine grid points.

Two possible coarsenings exist for the interior part of the domain: the red-black coarsening (figure 7.2(g)) which is the same as the first coloring for the isotropic Poisson equation (see section 6.1) and the semi-coarsening (figure 7.2(h)) which is the same as the first coloring for the anisotropic Poisson equation (see section 6.2). Interpolation for both colorings is simple, since points depend on just one upstream point:

$$I_H^h = \begin{bmatrix} & 0 & \\ 1 & 1 & 0 \\ & 0 & \end{bmatrix} \tag{7.13}$$

7.1.2 Red black coarsening

Using the Galerkin principle, the coarse grid operator on a red-black coarsened grid can be calculated as:

$$L^H = \frac{\epsilon}{2h^2}\begin{bmatrix} & 0 & & \\ & -1 & & -1 & \\ -1 & & 6 & & -1 \\ & -1 & & -1 & \\ & & 0 & & \end{bmatrix} + \frac{1}{2h}\begin{bmatrix} & & 0 & & \\ & 0 & & & 0 \\ -a & & a+2b & & \\ & -b & & & -b \\ & & 0 & & \end{bmatrix} \tag{7.14}$$

To check if this discretization is indeed a good approximation to the fine grid operator, a Taylor series expansion can be used, which yields:

$$L^H = a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + \text{Second order derivatives} \tag{7.15}$$

The convective terms are correctly approximated by the coarse grid operator, however, for good convergence also the diffusion in cross-stream direction has to be correctly approximated. To check this, the Taylor series expansion is rotated to the (ξ, η)-axis system, with ξ in streamwise

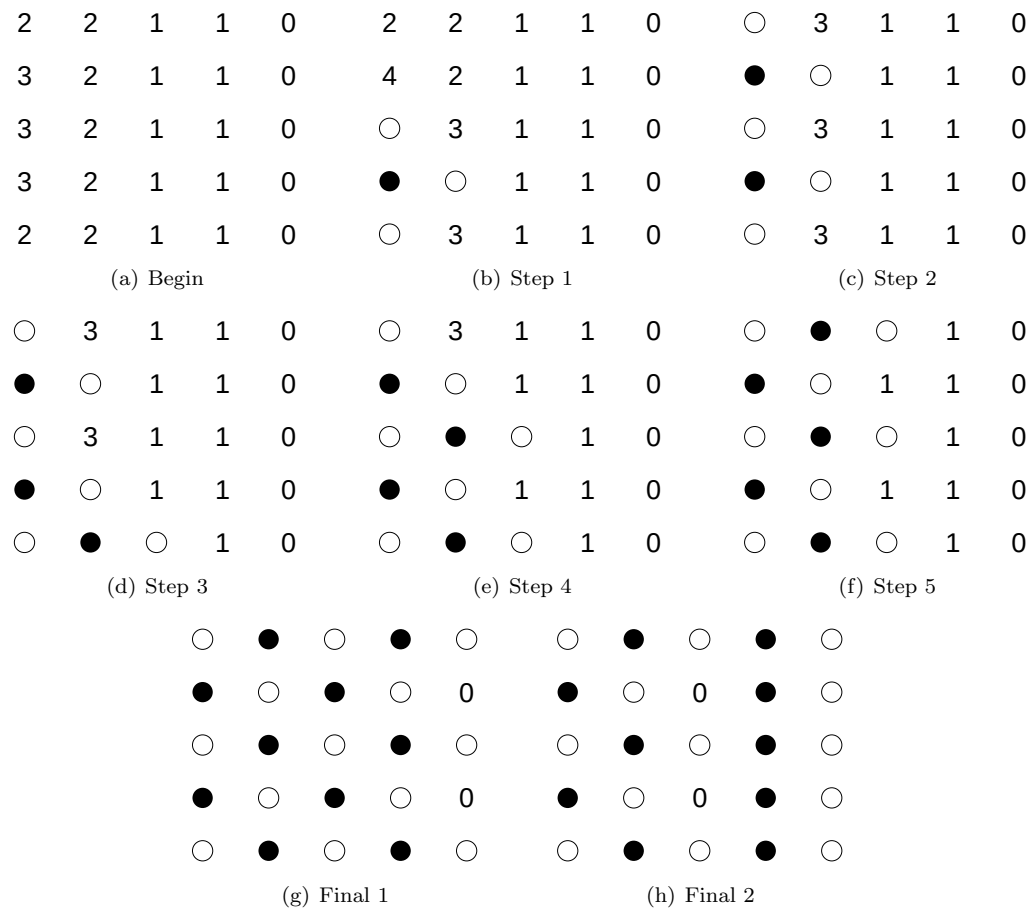


Figure 7.2: Convection-diffusion equation: Coloring for the first-order upwind discretization of the convective terms with $a = 1$ and $b = 0$, solid circle=coarse grid points, open circle=fine grid points, number=λ-value of undecided points

direction and η perpendicular to the flow. Now considering at terms involving derivatives in the η-direction only, yields:

$$-\frac{1}{a^2+b^2}\left(\frac{bh}{2}\left(a+b\right)^2+\epsilon\left(a^2+3b^2\right)\right)\frac{\partial^2 u}{\partial \eta^2}+\mathcal{O}\left(h^2\right) \tag{7.16}$$

While for the fine grid operator it yields:

$$-\frac{1}{a^2+b^2}\left(\frac{abh}{2}\left(a+b\right)+\epsilon\left(a^2+b^2\right)\right)\frac{\partial^2 u}{\partial \eta^2}+\mathcal{O}\left(h^2\right) \tag{7.17}$$

So operator 7.14 is a good coarse grid discretization as long as:

$$\frac{bh}{2}\left(a+b\right)^2+\epsilon\left(a^2+3b^2\right)\approx\frac{h}{2}\left(a+b\right)ab+\epsilon\left(a^2+b^2\right) \tag{7.18}$$

A convergence factor similar to the simplified two-grid convergence factor described in section 2.2.3 can be calculated as:

$$\begin{aligned}\rho=1-\frac{\text{Old diffusion}}{\text{New diffusion}} &= 1-\frac{abh\left(a+b\right)+2\epsilon\left(a^2+b^2\right)}{bh\left(a+b\right)^2+2\epsilon\left(a^2+3b^2\right)}\\ &= \frac{b^2h\left(a+b\right)+4\epsilon b^2}{bh\left(a+b\right)^2+2\epsilon\left(a^2+3b^2\right)}\end{aligned} \tag{7.19}$$

Or in terms of non-dimensional parameters:

$$\rho=\frac{\sin^2\alpha\left(\cos\alpha+\sin\alpha\right)+4\epsilon_*\sin^2\alpha}{\sin\alpha\left(\cos\alpha+\sin\alpha\right)^2+2\epsilon_*\left(1+2\cos^2\alpha\right)} \tag{7.20}$$

A contour plot of this convergence factor is shown in figure 7.3(a). Estimated two-grid convergence is quite good for all allowable α and ϵ_*. Worst performance is obtained for $\epsilon_*=0$ and $\alpha=\arctan\left(\epsilon_{str}\right)$:

$$\rho=\frac{\sin^2\alpha\left(\cos\alpha+\sin\alpha\right)}{\sin\alpha\left(\cos\alpha+\sin\alpha\right)^2}=\frac{\epsilon_{str}}{1+\epsilon_{str}} \tag{7.21}$$

7.1.3 Semi-coarsening

Using the Galerkin principle, the coarse grid operator on a semi-coarsened grid can be calculated:

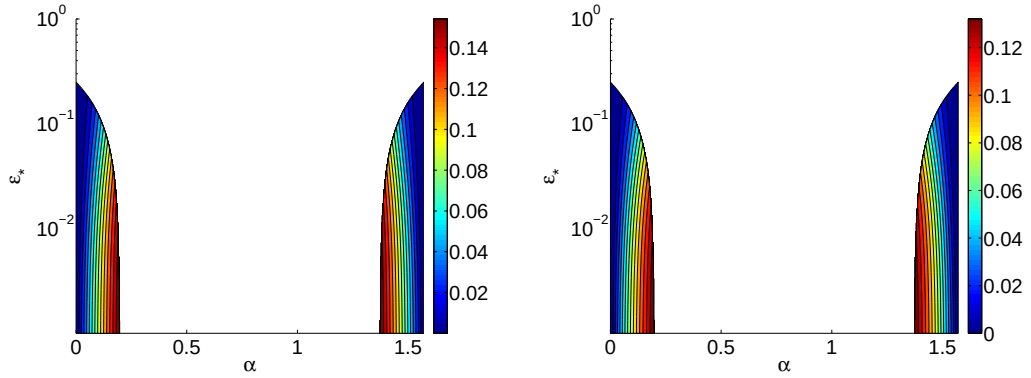
$$L^H=\frac{\epsilon}{2h^2}\begin{bmatrix}0&0&0\\0&-2&0\\-1&6&-1\\0&-2&0\\0&0&0\end{bmatrix}+\frac{1}{2h}\begin{bmatrix}0&0&0\\0&0&0\\-a&a+2b&0\\0&-2b&0\\0&0&0\end{bmatrix} \tag{7.22}$$

To check if this discretization is indeed a good approximation to the fine grid operator a Taylor series expansion can again be used:

$$L^H=a\frac{\partial u}{\partial x}+b\frac{\partial u}{\partial y}+\text{Second order derivatives} \tag{7.23}$$

Like in the previous paragraph special care has to be taken for terms involving derivatives in η-direction only. Rotating the Taylor series expansion to the (ξ,η)-axis and considering terms involving derivatives in the η-direction only, yields:

$$-\frac{1}{a^2+b^2}\left(\frac{h}{2}\left(a+2b\right)ab+\epsilon\left(a^2+2b^2\right)\right)\frac{\partial^2 u}{\partial \eta^2}+\mathcal{O}\left(h^2\right) \tag{7.24}$$



(a) Red-Black coarsening

(b) Semi coarsening

Figure 7.3: Convection-diffusion equation: Two-grid convergence factor in region I and II for the first-order upwind discretization of the convective terms as a function of α and ϵ_* for $\epsilon_{str} = 0.2$

For the fine grid operator the equivalent term is given by equation 7.17, hence operator 7.22 is a good coarse grid discretization as long as:

$$\frac{abh}{2}(a+2b) + \epsilon\left(a^2+2b^2\right) \approx \frac{abh}{2}(a+b) + \epsilon\left(a^2+b^2\right) \tag{7.25}$$

A convergence factor similar to the simplified two-grid convergence factor now yields:

$$\begin{aligned} \rho = 1 - \frac{\text{Old diffusion}}{\text{New diffusion}} &= 1 - \frac{h(a+b)ab + 2\epsilon\left(a^2+b^2\right)}{h(a+2b)ab + 2\epsilon\left(a^2+2b^2\right)} \\ &= \frac{hab^2 + 2\epsilon b^2}{h(a+2b)ab + 2\epsilon\left(a^2+2b^2\right)} \end{aligned} \tag{7.26}$$

Or equivalently:

$$\rho = \frac{\cos\alpha \sin^2\alpha + 2\epsilon_* \sin^2\alpha}{(\cos\alpha + 2\sin\alpha)\cos\alpha\sin\alpha + 2\epsilon_*(1+\cos^2\alpha)} \tag{7.27}$$

A contourplot of the two-grid convergence factor is shown in figure 7.3(b). Estimated convergence is satisfactory for all possible combinations of α and ϵ_* in region I. The largest error is obtained for $\epsilon_* = 0$ and $\alpha = \arctan\epsilon_{str}$:

$$\rho = \frac{\cos\alpha\sin^2\alpha}{(\cos\alpha + 2\sin\alpha)\cos\alpha\sin\alpha} = \frac{\epsilon_{str}}{1+2\epsilon_{str}} \tag{7.28}$$

7.1.4 Region III

In region III diffusion is dominant over convection, so all points are strongly influenced by their four neighboring points. This is exactly the same situation as for the standard 5-point discretization of the isotropic Poisson problem. Therefore the coloring is performed in the same manor as depicted is figure 6.1. Interpolation and restriction, however, become different from that used for the Poisson problem:

$$I_H^h = \frac{1}{ah+bh+4\epsilon}\begin{bmatrix} & \epsilon & \\ ah+\epsilon & ah+bh+4\epsilon & \epsilon \\ & bh+\epsilon & \end{bmatrix} \tag{7.29}$$

Using the Galerkin principle, the coarse grid operator becomes:

$$L^H = \frac{1}{2h\left(ah+bh+4\epsilon\right)}\left(\frac{\epsilon^2}{h}\begin{bmatrix} 0 & -1 & 0 \\ -2 & -2 & \\ -1 & 12 & -1 \\ -2 & -2 & \\ 0 & -1 & 0 \end{bmatrix} + \right.$$
$$\left.\begin{bmatrix} 0 & 0 & 0 \\ -2a\epsilon & 0 & \\ -a\left(ah+2\epsilon\right) & \left(a+b\right)\left(ah+bh+6\epsilon\right) & 0 \\ -2a\epsilon-2b\epsilon-2abh & -2b\epsilon & \\ 0 & -b\left(bh+2\epsilon\right) & 0 \end{bmatrix}\right) \tag{7.30}$$

As in previous sections a Taylor series expandion can be used, which results in the following dissipation term in cross stream direction:

$$-\frac{1}{a^2+b^2}\left(\frac{abh}{2}\left(a+b\right)+\epsilon\left(a^2+b^2\right)\right)\frac{\partial^2 u}{\partial \eta^2}+\mathcal{O}\left(h^2\right) \tag{7.31}$$

This is exactly the same as for the fine grid operator, so operator 7.30 is a good coarse grid representation of operator 7.4.

7.1.5 Region IV

In the last region not just one single variable is dominant, but both a and b are important for the discretization. Therefore points are strongly influenced by both of their downstream points. The coloring process is shown in figure 7.1.5. The final coloring is just the red-black coloring as also observed in the other regions. Interpolation, however, is in this case performed by using both upstream points:

$$I_H^h = \begin{bmatrix} & 0 & \\ a_* & 1 & 0 \\ & b_* & \end{bmatrix} \tag{7.32}$$

with

$$a_* = \frac{ah+\epsilon}{\left(a+b\right)h+2\epsilon} \tag{7.33}$$
$$b_* = \frac{bh+\epsilon}{\left(a+b\right)h+2\epsilon} \tag{7.34}$$

Using the Galerkin principle, the coarse grid operator becomes:

$$L^H = \frac{\epsilon}{2h^2}\begin{bmatrix} & -b_* & \\ -1+4a_*b_* & & -1 \\ -a_* & 2+4\left(a_*^2+b_*^2\right) & -a_* \\ -1 & & -1+4a_*b_* \\ & -b_* & \end{bmatrix}$$
$$+\frac{1}{2h}\begin{bmatrix} & 0 & \\ -ab_*+a_*b_*\left(a+b\right) & & 0 \\ -aa_* & \left(1+a_*^2+b_*^2\right)\left(a+b\right)-aa_*-bb_* & 0 \\ -ab_*-ba_* & & -ba_*+a_*b_*\left(a+b\right) \\ & -bb_* & \end{bmatrix} \tag{7.35}$$

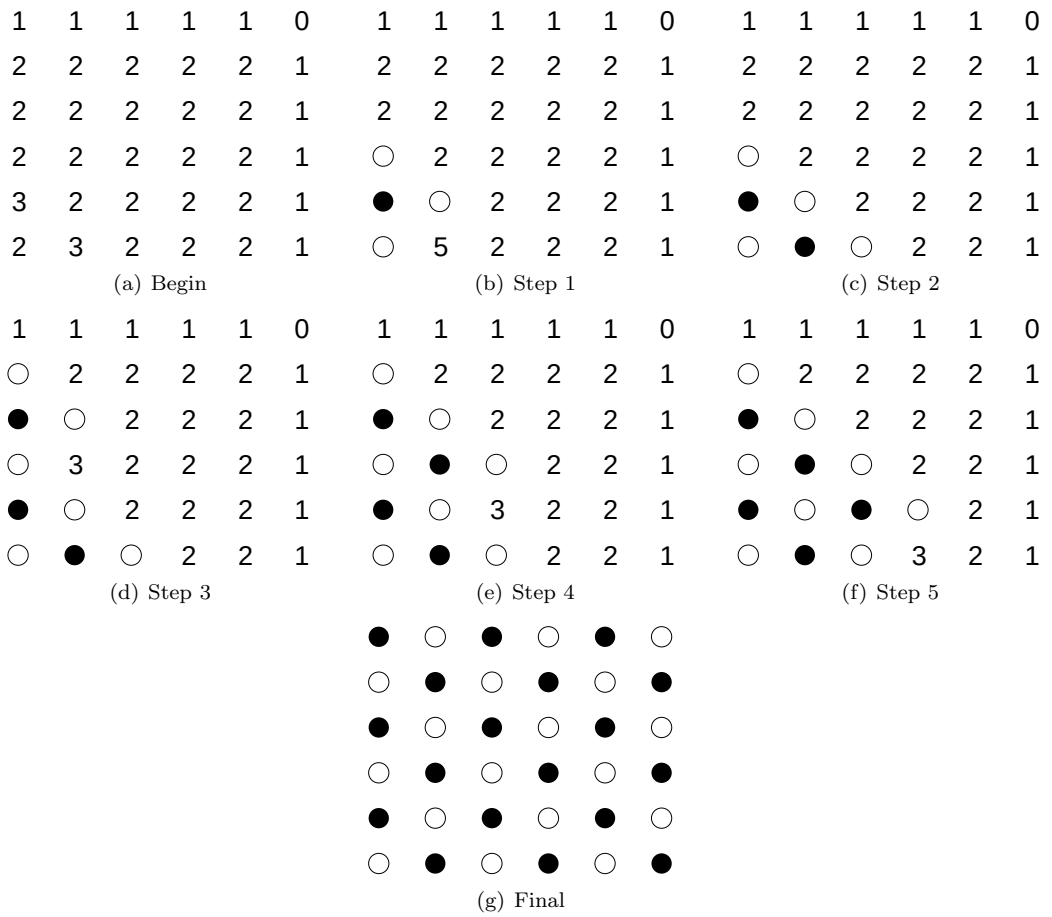


Figure 7.4: Convection-diffusion equation: Coloring for the first-order upwind discretization of the convective terms with $a = \frac{\sqrt{2}}{2}$ and $b = \frac{\sqrt{2}}{2}$, solid circle=coarse grid points, open circle=fine grid points, number=λ-value of undecided points

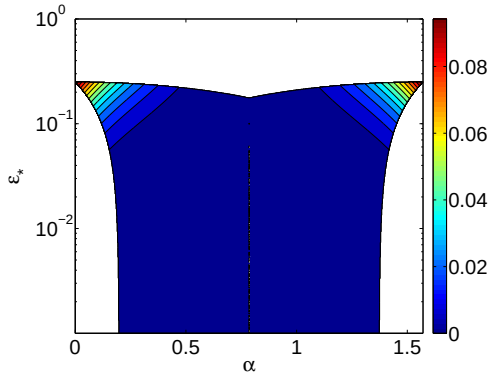


Figure 7.5: Convection-diffusion equation: Two-grid convergence factor in region IV for the first-order upwind discretization of the convective terms as a function of α and ϵ_* for $\epsilon_{str} = 0.2$

In this case a Taylor series expansion lead to the following diffusion term in cross stream direction:

$$-\frac{1}{a^2+b^2}\left(\frac{h}{2}(a+b)\,ab+\epsilon\left(a^2+b^2\right)+\left(\frac{h}{2}(a+b)+2\epsilon\right)\frac{\epsilon^2(a-b)^2}{(ah+bh+2\epsilon)^2}\right)\frac{\partial^2 u}{\partial\eta^2}+\mathcal{O}\left(h^2\right) \quad (7.36)$$

Comparison with the fine grid operator (equation 7.17), shows that operator 7.35 is a good coarse grid discretization as long as:

$$\frac{h}{2}(a+b)\,ab+\epsilon\left(a^2+b^2\right)+\left(\frac{h}{2}(a+b)+2\epsilon\right)\frac{\epsilon^2(a-b)^2}{(ah+bh+2\epsilon)^2}\approx\frac{h}{2}(a+b)\,ab+\epsilon\left(a^2+b^2\right) \quad (7.37)$$

A convergence factor similar to the simplified two-grid convergence factor described in section 2.2.3 can be calculated as:

$$\begin{aligned}\rho = 1-\frac{\text{Old diffusion}}{\text{New diffusion}} &= 1-\frac{h(a+b)\,ab+2\epsilon\left(a^2+b^2\right)}{h(a+b)\,ab+2\epsilon\left(a^2+b^2\right)+(ah+bh+4\epsilon)\frac{e^2(a-b)^2}{(ah+bh+2\epsilon)^2}}\\ &= \frac{(ah+bh+4\epsilon)\frac{e^2(a-b)^2}{(ah+bh+2\epsilon)^2}}{h(a+b)\,ab+2\epsilon\left(a^2+b^2\right)+(ah+bh+4\epsilon)\frac{e^2(a-b)^2}{(ah+bh+2\epsilon)^2}}\end{aligned} \quad (7.38)$$

Or equivalently:

$$\rho=\frac{(\cos\alpha+\sin\alpha+4\epsilon_*)\frac{\epsilon_*^2(\cos\alpha-\sin\alpha)^2}{(\cos\alpha+\sin\alpha+2\epsilon_*)^2}}{(\cos\alpha+\sin\alpha)\cos\alpha\sin\alpha+2\epsilon_*+(\cos\alpha+\sin\alpha+4\epsilon_*)\frac{\epsilon_*^2(\cos\alpha-\sin\alpha)^2}{(\cos\alpha+\sin\alpha+2\epsilon_*)^2}} \quad (7.39)$$

A contourplot of this equation is shown in figure 7.5. The convergence factors are quite small for all possible combinations of α and ϵ_* in region IV. Worst performance is when $\alpha = 0$ or $\pi/2$ and $\epsilon_* = \frac{\epsilon_{str}}{1-\epsilon_{str}}$:

$$\rho=\frac{(1+4\epsilon_*)\,\epsilon_*}{2+9\epsilon_*+12\epsilon_*^2}=\frac{(3\epsilon_{str}+1)\,\epsilon_{str}}{5\epsilon_{str}^2+5\epsilon_{str}+2}=0.1 \text{ for } \epsilon_{str}=0.2 \quad (7.40)$$

7.1.6 Conclusion

In the preceding sections it has been shown that the current AMG method is able to produce accurate coarse grid discretizations for the first-order upwind discretization of the convection-diffusion

equation. For small ϵ the resulting discretizations for the convective terms on coarser grids are also upwind. Four-directional Gauß-Seidel relaxation is a good smoother for these discretizations. The simplified two-grid convergence factors, however, are not zero. Therefore according to equation 5.47, at least W-cycles are required for convergence rates independent of the number of levels.

7.2 First-order discretization, recirculation test case

In this section AMG is applied to the testcase of recirculation. This is the conventional convection-diffusion equation on the unit square domain with Dirichlet boundary conditions:

$$\begin{aligned} -\epsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{\partial a(x,y)u}{\partial x}+\frac{\partial b(x,y)u}{\partial y} &= f(x,y) \qquad (x,y)\in\Omega \\ u &= u_0 \qquad (x,y)\in\partial\Omega \end{aligned} \tag{7.41}$$

with:

$$\begin{aligned} \epsilon &= 10^{-5}\ \tfrac{\mathrm{m}^2}{\mathrm{s}} \\ a(x,y) &= -sin(\pi x)\,cos(\pi y)\ \tfrac{\mathrm{m}}{\mathrm{s}} \\ b(x,y) &= sin(\pi y)\,cos(\pi x)\ \tfrac{\mathrm{m}}{\mathrm{s}} \\ f(x,y) &= 0\ \tfrac{1}{\mathrm{s}} \end{aligned} \tag{7.42}$$

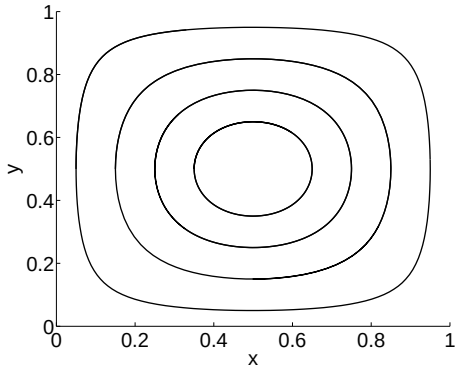
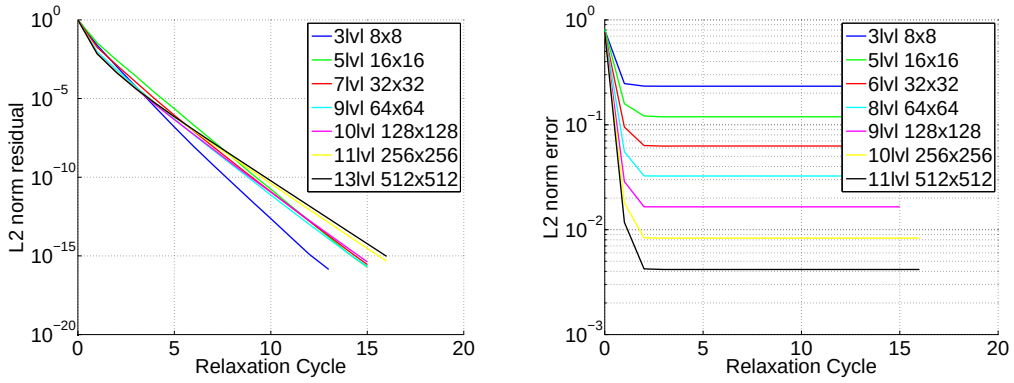


Figure 7.6: Convection-diffusion equation: Streamlines for the recirculation test case

For this case characteristics form closed loops and thus never meet upstream Dirichlet boundaries (see figure 7.6). The first two-levels of coloring for an initial 32x32 grid are shown in figure 7.8. The first step of coarsening is the red-black coarsening, which is a good strategy in all regions. In the second step the algorithm coarsens the grid only in the direction of the flow. This can be seen at the corners of the domain, where the flow is diagonal. More details about the different grid sizes can be found in table 7.2, where information about the different grids is provided for an initial 256x256 grid. An estimate of the total work required for a cycle can be obtained by multiplying the nonzero matrix entries with the number of relaxations performed on that level and summing these values over all levels. For an initial 256x256 grid one W(1,1)-cycle requires approximately as many operations as 56 relaxation sweeps on the finest level. Convergence results of applying W(1,1) AMG cycles to the first-order upwind discretization of this problem are shown in figure 7.7. In all cases coarsening is performed until at most 20 points remain on the coarsest grid. Good performance is achieved for all target grid sizes considered.

Level	Unknowns	Nonzero entries	Percentage	Times relaxed
10	65536	326656	7.61e-3	2
9	32207	264351	2.55e-02	4
8	15568	186168	7.68e-02	8
7	7245	135811	2.59e-01	16
6	3179	92351	9.14e-01	32
5	1322	56490	3.23e+0	64
4	498	26750	1.08e+01	128
3	163	8327	3.13e+01	256
2	44	1260	6.51e+01	512
1	8	62	9.69e+01	

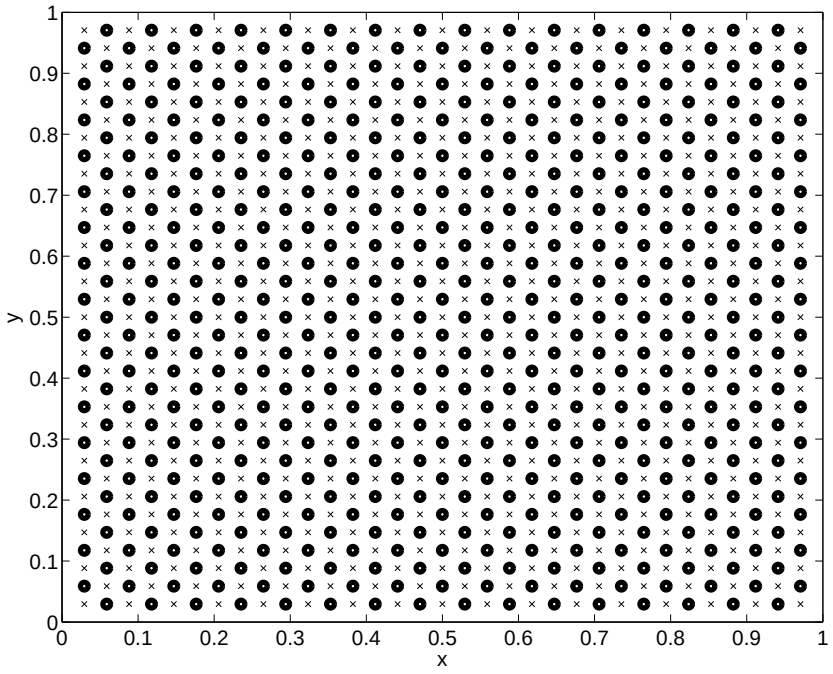
Table 7.2: Convection-diffusion equation: AMG grid information for the recirculating test case with first-order upwind discretization of the convective terms on a target 256x256 grid



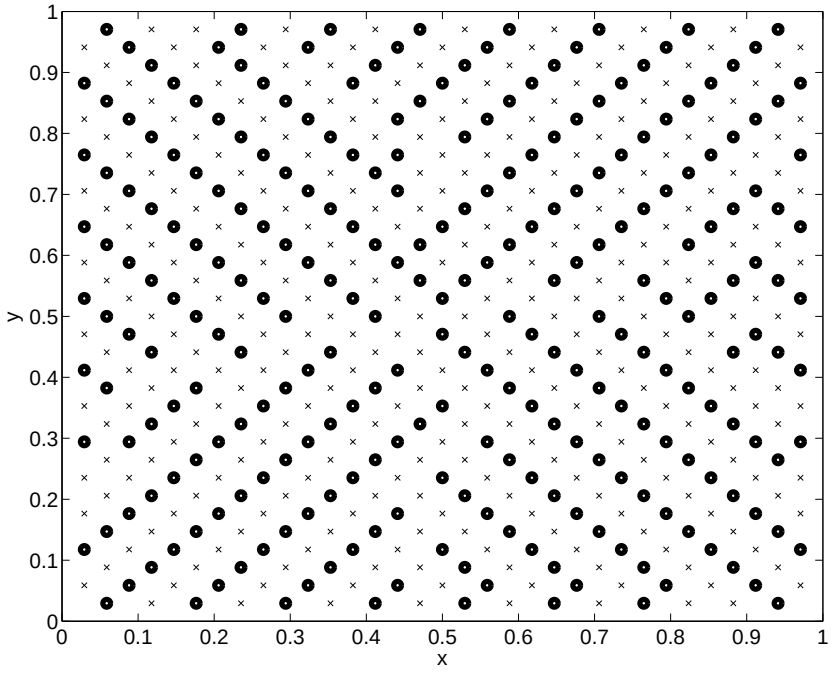
(a) L_2-norm of the residual

(b) L_2-norm of the error

Figure 7.7: Convection-diffusion equation: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of W(1,1) AMG cycles for the test case of recirculation for different gridsizes, using symmetric Gauß-Seidel relaxation with a first-order upwind discretization of the convective terms



(a) Level 1



(b) Level 2

Figure 7.8: Convection-diffusion equation: Two-levels of coloring for the the recirculation test case with first-order upwind discretization of the convective terms and $\epsilon_{str} = 0.2$, solid circle=coarse grid points, cross=fine grid points

7.3 Higher-order discretization, recirculating test case

In this section some preliminary results of AMG applied to higher-order upwind biased discretizations of the convection-diffusion equation without limiters are shown. These results are provided without a good theoretical overview of the effect of algebraic coarsening on the solution of the problem. Further research is required for better performance over a greater range of test cases.

Two important changes have been made to the AMG algorithm with respect to the one used so far:

- Truncation of interpolation operators is omitted (i.e. $\epsilon_{tr} = 0$)
- Points with λ value of 0 after the coloring algorithm has been performed are turned into coarse grid points instead of fine grid points.

Both changes considerably increase the computational effort required for one cycle. The first one, because the size of the coarse grid operators increases significantly as the size of the interpolation operators increases. The second one, simply because the number of coarse grid points increases. These two effects become more clear from table 7.3, where information is given about the different grid levels for a target 256x256 grid. The first few coarsening steps reduce the number of unknowns by a factor of approximately 2 only, while the number of Nonzero matrix entries is only reduced by a factor of 1.3-1.5. An estimate of the total work required for a cycle can be obtained by multiplying the number of nonzero entries with the number of relaxations on that level and over summing all levels. For an initial 256x256 grid a W(1,1)-cycle requires approximately as many operations as 91 relaxation sweeps on the finest level.

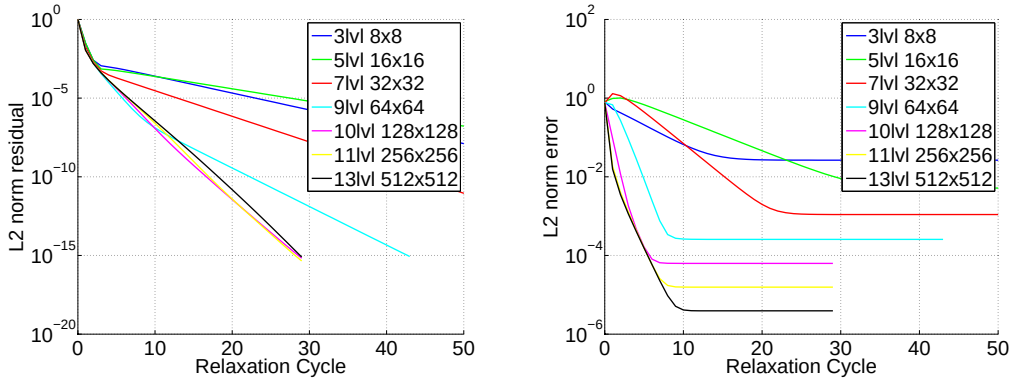
The convergence histories for applying W(1,1) cycles for different target grids is presented in figure 7.9. For large target grid sizes (from 128 cells on) convergence is acceptable, but for small target grids performance is unacceptable. This is because ϵ_* on dense target grids is higher and the problem tends to behave more like Poisson type of problems. This is also illustrated in figure 7.10 where convergence histories of the residual and error are plotted for a constant 256x256 grid for different ϵ. For $h\epsilon > 1$ convergence is similar to that found for the Poisson problem (see figure 6.3), while for smaller ϵ convergence deteriorates.

Level	Unknowns	Nonzero entries	Percentage	Times relaxed
11	65536	586752	1.37e-02	2
10	33461	432435	3.86e-02	4
9	17326	330018	1.10e-01	8
8	9103	239167	2.89e-01	16
7	4219	153955	8.65e-01	32
6	1865	102103	2.94e+00	64
5	822	65812	9.74e+00	128
4	353	41377	3.32e+01	256
3	143	18993	9.29e+01	512
2	54	2916	1.00e+02	1024
1	19	361	1.00e+02	

Table 7.3: Convection-diffusion equation: AMG grid information for the test case of recirculation with κ-scheme discretization of the convective terms on a 256x256 grid ($\kappa = 0$)

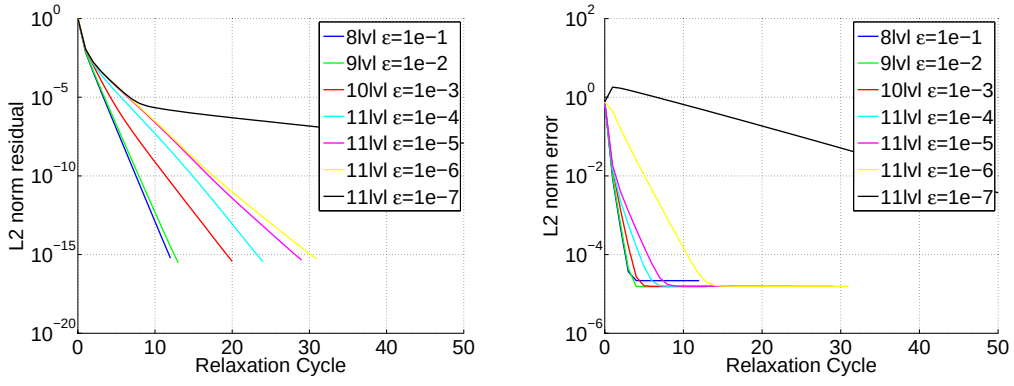
7.4 Conclusion

In this chapter is has been shown that algebraic multi-grid techniques are able to efficiently solve first-order upwind discretizations of the convection-diffusion equation. In all cases convergence rates of about 0.1 are obtained for the problems considered. For higher-order upwind biased



(a) L_2-norm of the residual (b) L_2-norm of the error

Figure 7.9: Convection-diffusion equation: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of W(1,1) AMG cycles for the test case of recirculation for different gridsizes, using symmetric Kaczmarz relaxation on the κ-scheme discretization of convective terms



(a) L_2-norm of the residual (b) L_2-norm of the error

Figure 7.10: Convection-diffusion equation: L_2-norm of the residual ($\left|r^h\right|_2$) and the error ($\left|u^h - u^*\right|_2$) as a function of the number of W(1,1) AMG cycles for the test case of recirculation for different values of ϵ on a target 256x256 grid, using symmetric Kaczmarz relaxation on the κ-scheme discretization of convective terms

discretizations, however, the method does not yet give good convergence for problems with extremely low diffusion. Also the limiters required for smooth solutions with the higher upwind biased discretizations have not yet been included. These limiters could on the one hand increase performance of the algorithm since they reduce (non-physical) oscillations in the flow, while on the other hand they decrease the performance since the problem becomes nonlinear. Further research of AMG methods for this kind of discretizations is required.

Chapter 8

Conclusions and Recommendations

Multi-level techniques have the potential of substantially enhancing the performance of iterative methods used to obtain solutions of the convection diffusion equation end for partial differential equations in general. Standard Multi-Level schemes already are capable of solving the Poisson-like problems very efficiently. However, this efficiency does not automatically extend to other partial differential equations that appear very similar. A new detailed analysis is required to identify slowly converging components on the fine grid and for a way to represent and solve these components on a coarser grid.

This conclusion is even more true for convection-dominated flows. Smoothers for this kind of flow problems are not capable of reducing error components that are smooth in the direction of the flow, but oscillatory in the direction perpendicular to the flow. Standard coarse grid operators are not able to accurately represent these components on the coarse grid, because they show different behavior in the direction perpendicular to the flow than the fine grid operator. Therefore more accurate discretizations are required on coarser levels. In a geometrical multi-grid setting this can be achieved by employing higher-order discretizations or by using Galerkin coarsening. Both approaches require a more sophisticated smoothing procedure on coarser grids. The developed multi-level method shows good, mesh-size independent convergence for first-order upwind based discretizations of the convection-diffusion equation, if not too many grid levels (up to 8) are employed. For higher-order upwind biased discretizations the smoothing becomes even more troublesome and good convergence is only observed when using at most four grid levels.

Another way to cope with the problems associated with inadequate coarse grid representation of the fine grid operator is to use algebraic multi-grid (AMG). In AMG the choice of coarse grids is not determined beforehand, but rather based on the actual problem itself. Interpolation and restriction operators are then determined based on the chosen grid and the problem. The coarse grid operator follows from the Galerkin principle.

For Poisson and Poisson-like problems AMG and geometric multi-grid show comparable performance. AMG, however, has the advantage of being more flexible and robust. There is no need to adjust the solver for different partial differential equations. This, however, comes with the disadvantage of a higher computational cost. Therefore it is preferred use geometric multi-grid techniques whenever possible. AMG methods can efficiently be used as a black box solver for a wider class of problems.

AMG can be used directly for the first-order upwind discretization of the convection-diffusion problem. Extremely fast, mesh-size independent, convergence is observed for all problems considered. For higher-order discretizations promising results have been obtained whenever the interpolation operators are not truncated. Not truncating the interpolation operator, however, leads to large stencils for coarse grid levels, increasing the computational time required to solve the problem. Therefore it is recommended to further study the effect of AMG coarsening for higher-order

discretizations of convective terms. Higher-order discretizations also have the drawback of being non-linear, if monotonicity is required. This non-linearity has not yet been implemented in the presented algorithm.

When also higher-order discretizations of convection-diffusion equation can be solved efficiently a good step has been made in the direction of the ultimated goal of solving the Navier-Stokes or Euler equations efficiently. However, before these problems can be solved it is also required that systems of equations and non-linear equations can be solved efficiently.

Appendix A

Discretization convective terms

In this appendix different conservative discretizations (central, first-order upwind and κ-schemes) are considered for convective terms with non constant convective speeds:

$$\frac{\partial au}{\partial x} \tag{A.1}$$

All discretization are based on a finite volume method, where volume-averaged values are stored in the memory and boundary values are reconstructed by interpolation or extrapolation of the cell averages. The derivatives can then be approximated as:

$$\frac{\partial au}{\partial x} = \frac{1}{h}\left(-\left(au\right)_{i-\frac{1}{2}} + \left(au\right)_{i+\frac{1}{2}}\right) \tag{A.2}$$

With $a_{i\pm\frac{1}{2}}$ the convective speed at the cell interfaces (which are known when an analytic function for the convective speed is specified, otherwise they have to be reconstructed as well) and $u_{i\pm\frac{1}{2}}$ the reconstructed value at the cell interfaces. Different methods use different reconstruction methods for the values at the interfaces.

A.1 Central differencing

For central differencing the values at the interfaces are obtained by using the average of the two neighbors.

$$\begin{aligned} u_{i-\frac{1}{2}} &= \frac{1}{2}\left(u_{i-1} + u_i\right) \\ u_{i+\frac{1}{2}} &= \frac{1}{2}\left(u_i + u_{i+1}\right) \end{aligned} \tag{A.3}$$

Now the derivative becomes:

$$\begin{aligned} \frac{\partial au}{\partial x} &= \frac{1}{h}\left(-\left(au\right)_{i-\frac{1}{2}} + \left(au\right)_{i+\frac{1}{2}}\right) \\ &= \frac{1}{2h}\left(-a_{i-\frac{1}{2}}\left(u_{i-1} + u_i\right) + a_{i+\frac{1}{2}}\left(u_i + u_{i+1}\right)\right) \\ &= \frac{1}{2h}\left(-u_{i-1}a_{i-\frac{1}{2}} + u_i\left(-a_{i-\frac{1}{2}} + a_{i+\frac{1}{2}}\right) + u_{i+1}a_{i+\frac{1}{2}}\right) \end{aligned} \tag{A.4}$$

Or in stencil notation:

$$\frac{\partial au}{\partial x} = \frac{1}{2h}\left[\begin{array}{ccc} -a_{i-\frac{1}{2}} & -a_{i-\frac{1}{2}} + a_{i+\frac{1}{2}} & a_{i+\frac{1}{2}} \end{array}\right] \tag{A.5}$$

A.2 First-order upwind

For first-order upwind schemes the values at the interfaces are obtained by using either the left or the right value depending on the sign of the convection velocity:

$$\begin{aligned} u_{i-\frac{1}{2}} &= \begin{cases} u_{i-1}, & \text{if} \quad a_{i-\frac{1}{2}} \geq 0 \\ u_i, & \text{if} \quad a_{i-\frac{1}{2}} < 0 \end{cases} \\ u_{i+\frac{1}{2}} &= \begin{cases} u_i, & \text{if} \quad a_{i+\frac{1}{2}} \geq 0 \\ u_{i+1}, & \text{if} \quad a_{i+\frac{1}{2}} < 0 \end{cases} \end{aligned} \tag{A.6}$$

Now the derivative becomes:

$$\begin{aligned} \frac{\partial au}{\partial x} &= \frac{1}{h}\left(-\left(au\right)_{i-\frac{1}{2}} + \left(au\right)_{i+\frac{1}{2}}\right) \\ &= \begin{cases} \frac{1}{h}\left(-u_{i-1}a_{i-\frac{1}{2}} + u_i a_{i+\frac{1}{2}}\right), & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \frac{1}{h}\left(-u_{i-1}a_{i-\frac{1}{2}} + u_{i+1} a_{i+\frac{1}{2}}\right), & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} < 0 \\ \frac{1}{h}\left(-u_i a_{i-\frac{1}{2}} + u_i a_{i+\frac{1}{2}}\right), & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \frac{1}{h}\left(-u_i a_{i-\frac{1}{2}} + u_{i+1} a_{i+\frac{1}{2}}\right), & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} < 0 \end{cases} \end{aligned} \tag{A.7}$$

Or in stencil notation:

$$\frac{\partial au}{\partial x} = \begin{cases} \frac{1}{h}\left[\begin{array}{ccc} -a_{i-\frac{1}{2}} & a_{i+\frac{1}{2}} & 0 \end{array}\right], & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \frac{1}{h}\left[\begin{array}{ccc} -a_{i-\frac{1}{2}} & 0 & a_{i+\frac{1}{2}} \end{array}\right], & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} < 0 \\ \frac{1}{h}\left[\begin{array}{ccc} 0 & -a_{i-\frac{1}{2}} + a_{i+\frac{1}{2}} & 0 \end{array}\right], & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \frac{1}{h}\left[\begin{array}{ccc} 0 & -a_{i-\frac{1}{2}} & a_{i+\frac{1}{2}} \end{array}\right], & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} < 0 \end{cases} \tag{A.8}$$

A.3 Kappa-schemes

For the kappa-schemes the values at the interface are calculated by a more accurate (upwind biased) interpolation:

$$\begin{aligned} u_{i-\frac{1}{2}} &= \begin{cases} u_{i-1} + \frac{1+\kappa}{4}\left(u_i - u_{i-1}\right) + \frac{1-\kappa}{4}\left(u_{i-1} - u_{i-2}\right), & \text{if} \quad a_{i-\frac{1}{2}} \geq 0 \\ u_i - \frac{1+\kappa}{4}\left(u_i - u_{i-1}\right) - \frac{1-\kappa}{4}\left(u_{i+1} - u_i\right), & \text{if} \quad a_{i-\frac{1}{2}} < 0 \end{cases} \\ u_{i+\frac{1}{2}} &= \begin{cases} u_i + \frac{1+\kappa}{4}\left(u_{i+1} - u_i\right) + \frac{1-\kappa}{4}\left(u_i - u_{i-1}\right), & \text{if} \quad a_{i+\frac{1}{2}} \geq 0 \\ u_{i+1} - \frac{1+\kappa}{4}\left(u_{i+1} - u_i\right) - \frac{1-\kappa}{4}\left(u_{i+2} - u_{i+1}\right), & \text{if} \quad a_{i+\frac{1}{2}} < 0 \end{cases} \end{aligned} \tag{A.9}$$

Now the derivatives become (directly in stencil notation):

$$\frac{\partial au}{\partial x} = \frac{1}{4h} \begin{cases} \begin{bmatrix} a_{i-\frac{1}{2}}(1-\kappa) \\ a_{i-\frac{1}{2}}(-4+2\kappa) + a_{i+\frac{1}{2}}(-1+\kappa) \\ a_{i-\frac{1}{2}}(-1-\kappa) + a_{i+\frac{1}{2}}(4-2\kappa) \\ a_{i+\frac{1}{2}}(1+\kappa) \\ 0 \end{bmatrix}^T & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \begin{bmatrix} a_{i-\frac{1}{2}}(1-\kappa) \\ a_{i-\frac{1}{2}}(-4+2\kappa) \\ a_{i-\frac{1}{2}}(-1-\kappa) + a_{i+\frac{1}{2}}(1+\kappa) \\ a_{i+\frac{1}{2}}(4-2\kappa) \\ a_{i+\frac{1}{2}}(-1+\kappa) \end{bmatrix}^T & \text{if } a_{i-\frac{1}{2}} \geq 0 \text{ and } a_{i+\frac{1}{2}} < 0 \\ \begin{bmatrix} 0 \\ a_{i-\frac{1}{2}}(-1-\kappa) + a_{i+\frac{1}{2}}(-1+\kappa) \\ a_{i-\frac{1}{2}}(-4+2\kappa) + a_{i+\frac{1}{2}}(4-2\kappa) \\ a_{i-\frac{1}{2}}(1-\kappa) + a_{i+\frac{1}{2}}(1+\kappa) \\ 0 \end{bmatrix}^T & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} \geq 0 \\ \begin{bmatrix} 0 \\ a_{i-\frac{1}{2}}(-1-\kappa) \\ a_{i-\frac{1}{2}}(-4+2\kappa) + a_{i+\frac{1}{2}}(1+\kappa) \\ a_{i-\frac{1}{2}}(1-\kappa) + a_{i+\frac{1}{2}}(4-2\kappa) \\ a_{i+\frac{1}{2}}(-1+\kappa) \end{bmatrix}^T & \text{if } a_{i-\frac{1}{2}} < 0 \text{ and } a_{i+\frac{1}{2}} < 0 \end{cases} \tag{A.10}$$

Appendix B

Linked List

In this appendix the linked list used in the coloring algorithm for AMG is described. The list is used to determine the point with maximum λ without looping over the entire set of undecided points. The linked List consist of 3 Arrays:

$Prev[i]$ A pointer to the previous element with the same value of λ or -1 if no such element exist

$Next[i]$ A pointer to the next element with the same value of λ or -1 if no such element exist

$First[i]$ A pointer to the first element with $\lambda = i$ or -1 if no such element exist

When the linked list is initialized, no elements are in it, so all $First[i]$ should be -1. Therafter all variables are added to the list.

Algorithm 2 Initializing the linked list

```
for i ∈ λ_pos do
  First[i] = −1
end for
for i ∈ U do
  Add point i
end for
```

Adding a point i to the list is also quite easy (algorithm 3). A new element is always added as the first element of the list with that value of λ. When already an element with the same value of λ is present in the list, pointers for that element and for the new element have to be adjusted accordingly.

Algorithm 3 Adding point i with $\lambda = \lambda[i]$ to the linked list

```
Prev[i] = −1
Next[i] = First[λ[i]]
if Next[i] ≠ −1 then
  Prev[Next[i]] = i;
end if
First[λ[i]] = i
```

Removing a point i from the list is also quit easy (algorithm 4). The reference of the previous element, indicating the next element ($Next[Prev[i]]$) and the reference of the next element, indicating the previous element ($Prev[Next[i]]$) have to be adjusted (if they exist). Also if element i is the first element with $\lambda = \lambda_i$, the reference of $First[\lambda[i]]$ has to point to the next element with the same value of λ. An example of removing a point is shown in figure B.1. Here point $i = 7$ is removed from the list. From the previous point (point 3), the next pointer value (it was 7), is

changed to point to the next point (point 1). Also from the next point (point 1), the previous pointer value (it was 7), is changed to point to the previous point (point 3).

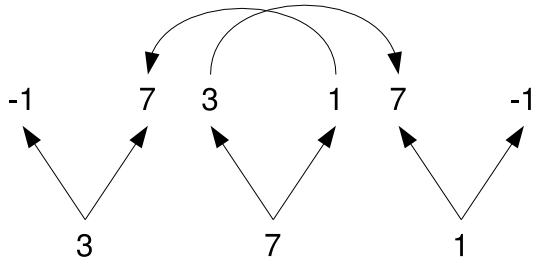


Figure B.1: Removing point $i = 7$ from the linked list

Algorithm 4 Removing point i from the linked list

if $Prev[i] \neq -1$ **then**
  $Next[Prev[i]] = Next[i]$
else
  $First[\lambda[i]] = Next[i]$
end if
if $Next[i] \neq -1$ **then**
  $Prev[Next[i]] = Prev[i]$
end if

Last but not least, a routine for determining the point with maximum λ has to be made (algorithm 5). This can be done by looping over all possible values of λ (starting from the maximum) and checking if $First[i]$ points to an element.

Algorithm 5 Search for the point i with maximum $\lambda[i]$

for $i \in \lambda_{pos}$ descending **do**
  if $First[i] \neq -1$ **then**
    return $First[i]$
  end if
end for
$First[\lambda[i]] = i$

Appendix C

Kaczmarz relaxation

In this appendix Kaczmarz relaxation is described. Kaczmarz relaxation is conventional Gauß-Seidel relaxation, but applied to a transformed system of equations. This set of equations is obtained by defining a new grid variable y^h as:

$$u^h = A^T y^h \tag{C.1}$$

with A the matrix corresponding to the discretization stencil. The transformed set of equations is now obtained by substituting C.1 into the original equation:

$$\begin{aligned} Au^h &= f^h \\ A\left(A^T y^h\right) &= f^h \\ \left(AA^T\right) y^h &= f^h \end{aligned} \tag{C.2}$$

Matrix $A_{km} = AA^T$ is called the Kaczmarz-matrix and has several interesting properties. First, it is symmetric:

$$A_{km}^T = \left(AA^T\right)^T = \left(AA^T\right) = A_{km} \tag{C.3}$$

Secondly it is positive definite ($\vec{u}^T A_{km}\vec{u} > 0$ for all $\vec{u} \in \mathbb{R}^n$):

$$\vec{u}^T A_{km}\vec{u} = \vec{u}^T AA^T\vec{u} = \left(\vec{u}^T A\right)\left(A^T\vec{u}\right) = \left(A^T\vec{u}\right)^T\left(A^T\vec{u}\right) = A^T\vec{u}\cdot A^T\vec{u} = \left|A^T\vec{u}\right|^2 \tag{C.4}$$

Therefore it can be proven that both Gauß-Seidel and ω-Jacobi relaxation on the new grid variable do not diverge [12].

For point Kaczmarz relaxation the required correction (for y^h) for point i can be calculated as:

$$\begin{aligned} \delta_i &= \frac{f_i^h - \left(AA^T y^h\right)_i}{\left(AA^T\right)_{i,i}} \\ &= \frac{f_i^h - \left(Au^h\right)_i}{\left(AA^T\right)_{i,i}} \end{aligned} \tag{C.5}$$

Now the required correction for points u_j is:

$$\begin{aligned} \bar{u}_j &= u_j + A_{j,i}^T\delta_i \\ &= u_j + A_{i,j}\delta_i \end{aligned} \tag{C.6}$$

A further extension of point Kaczmarz relaxation is the line Kaczmarz relaxation. In this relaxation process (just as with line Gauß-Seidel) the required correction for lines of constant x or y are calculated simultaneously. To illustrate this, the matrix A and vectors u and f can be rewritten:

$$Au = \begin{bmatrix} A_{ll} & A_{lo} \\ A_{ol} & A_{oo} \end{bmatrix}\begin{bmatrix} u_l \\ u_o \end{bmatrix} = \begin{bmatrix} f_l \\ f_o \end{bmatrix} = f \tag{C.7}$$

where the subscript l denotes points which lie on the line $x = x_t$ (i.e. $l = \{i \in \Omega : x_i = x_t\}$), and subscript o for all other points ($o = \Omega \setminus l$). Using this splitting, the Kaczmarz-matrix can be calculated:

$$AA^T = \begin{bmatrix} A_{ll}A_{ll}^T + A_{lo}A_{lo}^T & A_{ll}A_{ol}^T + A_{lo}A_{oo}^T \\ A_{ol}A_{ll}^T + A_{oo}A_{lo}^T & A_{ol}A_{ol}^T + A_{oo}A_{oo}^T \end{bmatrix} \tag{C.8}$$

One step of Kaczmarz line relaxation now consists of calculating the required correction y_l from:

$$\left(A_{ll}A_{ll}^T + A_{lo}A_{lo}^T\right) y_l = f_l - \left(A_{ll}u_l + A_{lo}u_o\right); \tag{C.9}$$

And updating all points u:

$$\bar{u} = u + \begin{bmatrix} A_{ll}^T \\ A_{lo}^T \end{bmatrix} y_l \tag{C.10}$$

The most computationally expensive step is solving equation C.9. Here an nxn-matrix needs to be inverted, with n the number of grid points on a single line. However, for standard 9-point stencils both A_{ll} and A_{ol} are tri-diagonal matrices and thus the matrix that needs to be inverted is at most a 5-diagonal matrix. The system can be solved (just as for line Gauß-Seidel) by using a Thomas algorithm or by applying 1D multi-grid techniques.

Local mode analysis of the Kaczmarz relaxation can be performed by considering the effect of the relaxation on the new grid variable (y^h).

Bibliography

[1] K. Bernert. τ-extrapolation - theoretical foundation, numerical experiment, and application to Navier-Stokes equations. *Journal on Scientific Computing*, 18(2):460–478, 1997.

[2] A. Brandt. Multi-grid techniques: 1984 guide with applications to fluid dynamics. *GMD-Study*, 85, 1984.

[3] A. Brandt. Algebraic multi-grid theory: The symmetric case. *Applied Mathematics and computations*, 19:23–56, 1986.

[4] A. Brandt. Rigorous local mode analysis of multi-grid. *Preliminary Proceedings 4th Copper Mountain Conference on Multi-Grid Methods*, 1990.

[5] A. Brandt. General highly accurate algebraic coarsening. *Electronic Transactions on Numerical Anallysis*, 10:1–20, 2000.

[6] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A multi-grid tutorial*. SIAM, 2000.

[7] M.H. Carpenter, D. Gottlieb, and S. Abarbanel. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *Journal of Computational Physics*, 108(2):272–295, 1993.

[8] B. Leer, van. Upwind-difference methods for aerodynamics problems governed by the Euler equations. *Lectures in Applied Mathematics*, 1985.

[9] R.J. Leveque. *Finite volume methods for hyperbolic problems*. Cambridge University Press.

[10] C.W. Oosterlee, F.J. Gaspar, T. Washio, and R. Wienands. Multigrid line smoothers for higher-order upwind discretizations of convection-dominated problems. *Journal of Computational Physics*, 139(2):273–307, 1998.

[11] R.M. Smith and A.G. Hutton. The numerical treatment of advection: A performance comparison of current methods. *Numerical Heat Transfer*, 5:439–461, 1982.

[12] U. Trottenberg U., C.W. Oosterlee, and Schuller A. *Multigrid*. Elsevier Academic Press, 2001.

[13] C.H. Venner and A.A. Lubrecht. *Multilevel methods in lubrication*. Elsevier, 2000.

[14] E.T.A. Weide, van der. *Discretization schemes for hyperbolic partial differential equations*. Universiteit Twente.

[15] P. Wesseling. *An introduction to multi-grid methods*. Edwards, 2004.

[16] I. Yavneh. Coarse-grid correction for nonelliptic and singular perturbation problems. *Journal on Scientific Computing*, 19(5):1682–1699, 1998.