Fast Iterative Solvers

Marcelo Cabral Filho
Mat. 404864
marcelo.cabral.filho@rwth-aachen.de

Multigrid Solvers
Project 2
SS 2023

Abstract

Project in order to implement a Multigrid solver to solve the Poisson equation. The algorithm was implemented in Python. Some analyses and characteristics of the given problem will be further discussed here.

1 Introduction

In this project, an algorithm for a Multigrid solver was coded to solve the Poisson equation using finite difference discretization on a Cartesian grid. With the provided equation for the forcing function $f(x, y)$ and a Dirichlet's boundary condition, the algorithm was tested for different input parameters and their results were analyzed.

1.1 Problem Description

The 2-D Poisson equation is given as follows:

$$\begin{cases} \nabla^2 u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{1}$$

Where $\Omega = (0, 1)^2$ with a finite difference discretization we have:

$$\begin{cases} -f_{i,j} = \frac{1}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1}{h^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) & \text{i,j = 1, ..., N-1,} \\ u_{i,j} = 0 & \text{otherwise} \end{cases} \tag{2}$$

With $f(x, y) = 8\pi^2 sin(2\pi x)sin(2\pi y)$ and the exact solution $u(x, y) = sin(2\pi x)sin(2\pi y)$.

2 Results and Discussion

The algorithm was tested for different settings that can be characterized into two classes W-cycle and V-Cycle.

2.1 W-Cycle

The parameters for the analysis of the algorithm based on this are presented in Figure 1. It can be seen that increasing the number of initial smoothing decreases the number of iterations that are needed to reach the minimum residual threshold. The initial smoothing parameter ν_1 relates to the number of times the Gauss-Seidel function is applied to the solution to reduce high-frequency error which contributes highly to the total residual. However, it does not have the same smoothing effect on low-frequency error. Figure 2 shows the relative residual convergence plot, which relative residual is given by: $\frac{||r^m||}{||r^0||}$, against the iteration count where it can be seen that increasing the smoothing count leads to a faster convergence rate.

It should also be noted that refining the mesh leads to slightly fewer iterations being required to reach convergence. However, in literature, the rate of convergence will be bounded by a number $\rho(\nu) < 1$ which depends on the number of pre-smoothing steps and which is independent of the finest step size $'h'$ and of the number of levels involved in the multigrid scheme. Nevertheless, in the project, the number of iterations needed to reach convergence differed. This can be justified by understanding that a nodal value is calculated based on its neighbouring nodes. For a coarse mesh, the difference between the value of the neighbour node and our node of interest will be very high for high-frequency problems and therefore, increases the total residual in each iteration step, unlike a fine mesh where we will obtain a smoother approximation of the solution.

W-Cycle

	Mesh Refinement	Pre- and Post- Smoothers			Iterations
	n	v_1	v_2	γ	m
Setting 1	4	1	1	2	12
Setting 2	4	2	1	2	10
Setting 3	7	1	1	2	10
Setting 4	7	2	1	2	8

Figure 1: Overview of the algorithm settings and iterations needed to solve the problem

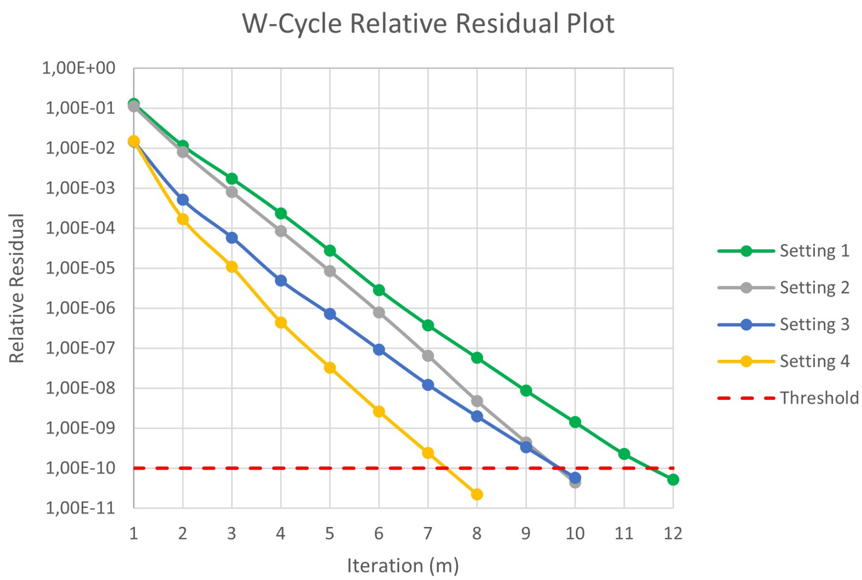

Figure 2: Relative Residual plotted over the iteration index

Nevertheless, there is a trade-off between the number of iterations needed to reach convergence and the total run-time of the solver. Figure 3 depicts the behaviour noted in the algorithm for different settings, where can be concluded that any sub-sequence increase in γ we reduce the amount of iterations needed to solve the problem, although the time to solve it increased. Recall that γ represents the number of recursive iterations or cycles that are performed on the coarse grid during the multi-grid algorithm. It affects the efficiency and effectiveness of the multigrid solver. A larger value of γ means more iterations on the coarse grid, which may help eliminate low-frequency errors more effectively. However, setting γ too high may lead to excessive computational costs. This trade can be seen in the timing when $\gamma = 3$.

	Mesh Refinement	Pre- and Post- Smoothers			Iterations	Timing		Iterations	Timing		Iterations	Timing
	n	ν_1	ν_2	γ	m	(s)	γ	m	(s)	γ	m	(s)
Setting 1	9	1	1	3	9	14,81	2	9	11,70	1	16	14,59
Setting 2	9	2	1	3	7	13,33	2	7	9,31	1	13	13,69
Setting 3	9	3	1	3	6	13,05	2	7	9,63	1	12	14,97
Setting 4	9	4	1	3	6	16,28	2	6	9,45	1	11	14,97

Figure 3: Overview of the algorithm setting in order to compare the run-time and iterations needed to solve the problem for different γ parameter

Figure 4 shows the run-time vs the convergence according to the set-up demonstrated on Figure 3.

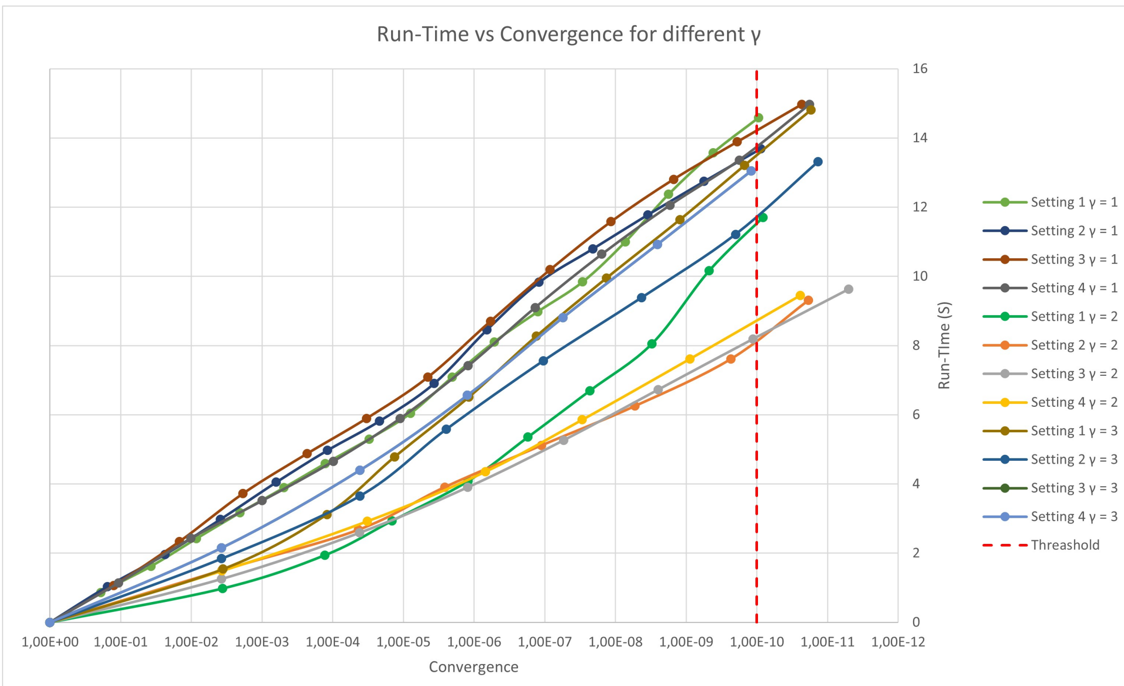

Figure 4: Run-time plotted over the relative residual for different γ parameter

2.2 V-Cycle

The performance of the V-Cycle was evaluated for the same input parameters as in Figure 1, except for the value of γ which was set to 1 as can be verified in Figure 5 below.

V-Cycle

	Mesh Refinement	Pre- and Post- Smoothers			Iterations
	n	v_1	v_2	γ	m
Setting 1	4	1	1	1	14
Setting 2	4	2	1	1	11
Setting 3	7	1	1	1	16
Setting 4	7	2	1	1	13

Figure 5: Overview of the algorithm settings and iterations needed to solve the problem

Due to the smaller size of the problem (i.e. the size of 'n'), run-time would not be meaningful to be compared for these parameters, so only the number of iterations and the residual plots were compared, as seen in Figure 6. Keeping the same trend as in W-Cycle, increasing the number of initial smoothing decreases the number of iterations to convergence. However, the overall number of iterations for convergence is higher for V-Cycle as compared to W-Cycle. This can be attributed to how in W-Cycle we solve the error equation twice on the coarse level. This gives a better initial guess of the solution for the next iteration and reduces the overall iterations required to achieve convergence. Also, the W-cycle performs multiple iterations on the coarse level, allowing it to smooth out high-frequency errors more effectively than the V-cycle, ergo performing fewer iterations.

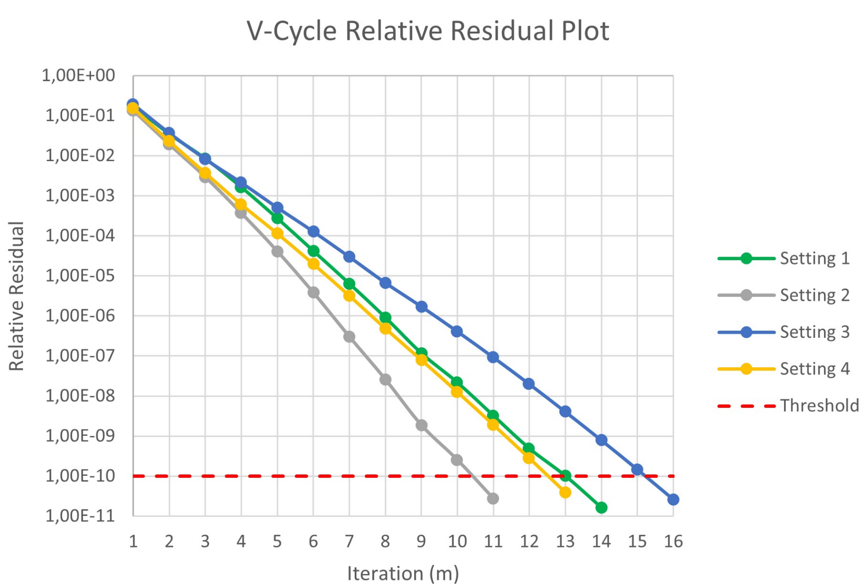

Figure 6: Relative Residual plotted against the iteration index