Multigrid Project

Author: Guy Erez
Academic advisor: Eran Treister

Background

Multigrid methods were originally applied to simple boundary value problems that arise in many physical applications. In such problems an analytic derivative was replaced with a numerical approximation. In that way a differential equation can be transform into a linear system. Our focus in this project is to solve the Laplace equation with Dirichlet boundary conditions on a two-dimensional grid.

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1}$$

The resulting linear system has many properties, it is symmetric, sparse and positive definite – all of those come in handy when using iterative methods.

In order to understand a problem many iterative methods suffer from we will introduce the idea of a Fourier series.

Each function can be represented as a sum of sin and cos of different frequencies. This decomposition is called the Fourier series of the function. It is useful for the next discussion to think of the error of an iterative method as composed of higher frequencies sin and cos (oscillatory modes) and lower frequencies sin and cos (smooth modes).

It was shown that iterative methods attempting to solve the linear system presented above, manage to eliminate the oscillatory part of the error while struggling when the error is composed only from smooth modes.

Multigrid schemes will try to overcome this smoothness property by moving the problem to a coarser grid. Assume an iterative method was applied until the error composed mainly from smooth components. Then the error is projected on a coarser grid, there its “seems” more oscillatory and can be eliminated. This correction needs to be interpolated back to the fine grid - the interpolation will be successful due to the smoothness of the error. This is the Multigrid idea in a nutshell.

In this project we implemented two multigrid schemes – V cycle and a two-grid correction scheme.

Notations:

$\Omega_h - a\ grid\ with\ spaces\ of\ length\ h$

$Au = f - the\ linear\ system\ to\ be\ solved$

$v - the\ approximation\ of\ u$

$e = u - v\ , the\ error$

$r = f - Av_h\ , the\ residual$

A subscript denotes to which grid the value is restricted.

Two-Grid Correction Scheme: $\boldsymbol{v_h \leftarrow MG(v_h, f_h)}$

1. Relax ν_1 times on $A_h u_h = f_h$ on Ω_h with initial guess v_h.
2. Compute the fine-grid residual $r_h = f_h - A_h v_h$ and restrict it to the coarse grid.
3. Relax $\nu 2$ times on $A_{2h} e_{2h} = r_{2h}$ on Ω_{2h}.
4. Interpolate the coarse grid error to the fine grid and correct the fine grid approximation by $v_h \leftarrow v_h + e_h$.
5. Relax ν_1 times on $A_h u_h = f_h$ on Ω_h with initial guess v_h.

V Cycle: $\boldsymbol{v_h \leftarrow V^h(v_h, f_h)}$

1. Relax ν_1 times on $A_h u_h = f_h$ with a given initial guess v_h.
2. If $(\Omega_h = coarsest\ grid)$
   - Relax ν_2 times on $A_h u_h = f_h$ with initial guess v_h

   Else
   - $f_{2h} \leftarrow f_h - A_h v_h\ (restricted\ to\ \Omega_{2h})$
   - $v_{2h} \leftarrow 0$
   - $v_{2h} \leftarrow V^{2h}(v_{2h}, f_{2h})$
   - Correct $v_h \leftarrow v_h + v_{2h}(interpolated\ to\ \Omega_h)$
3. Relax ν_1 times on $A_h u_h = f_h$ with initial guess v_h

**ν_1 is usually small, in our implementation it is fixed as 2

**ν_2 is usually larger, in our implementation it is a parameter of the method and taken around 10-20.

** "Relax ν_1 times" means call Jacobi method with ν_1 iterations.

Results

We will compare the performance of the V cycle scheme oppose to the two-grid correction scheme and the classic Jacobi iterations.
Each scheme will start with the same (random) initial guess and try to solve the homogeneous Laplace equation $\nabla^2 u = 0$ with zero boundary conditions (the only solution is the constant $u = 0$, thus the result is also the error).
The total number of Jacobi iterations is equal for all, this was achieved by adjustments to the number of inner iterations for each scheme.
A sample was taken after each sweep of the V cycle and the Two-Grid correction schemes. For comparison issues a sample was taken from the classic Jacobi scheme after a similar number of iterations was done.

<u>The details</u>:
Grid size: $2^8 \times 2^8$ nodes
$v1\ (V\ Cycle) = 1$
$v2\ (V\ Cycle) = 8$
$v1\ (2\ Grid\ Scheme) = 1$
$v2\ (2\ Grid\ Scheme) = 24$
$total\ number\ of\ Jacobi\ steps\ (for\ all\ three\ schemes) = 500$

Each scheme was applied on its output twenty times.
V Cycle iterations go throw all 8 possible levels, while the two-grid correction scheme preforms only one grid transfer to a $2^7 \times 2^7$ grid and back.

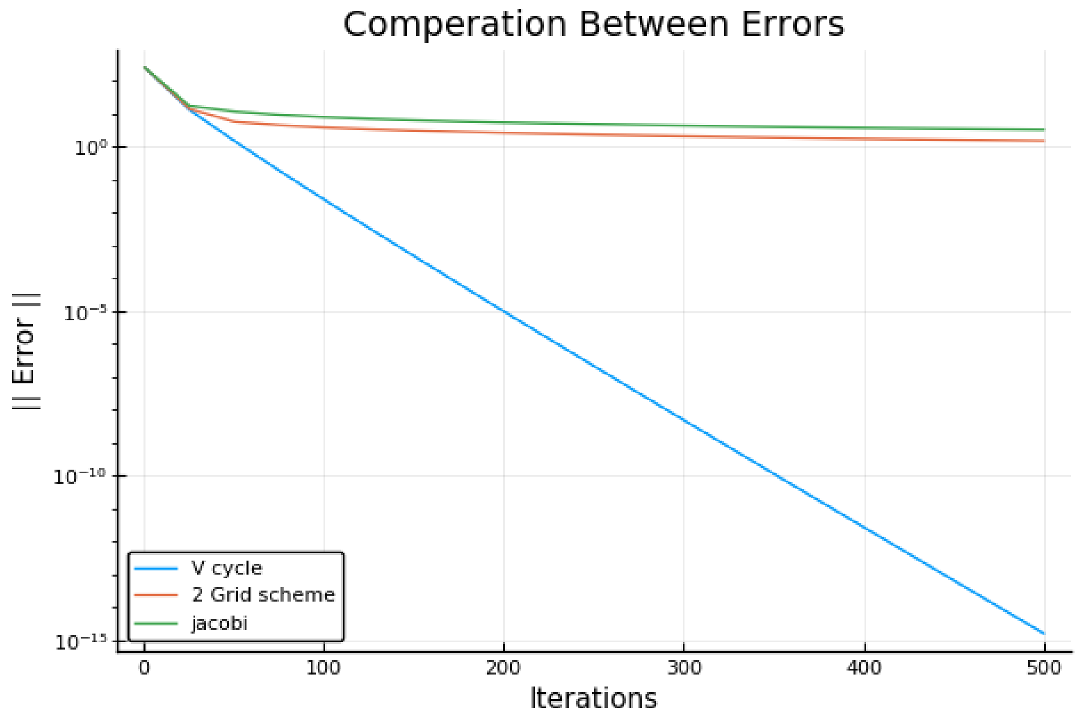

Figure 1 – the output error of all three schemes on a logarithmic scale. V Cycle is far more effective for the tested scenario.

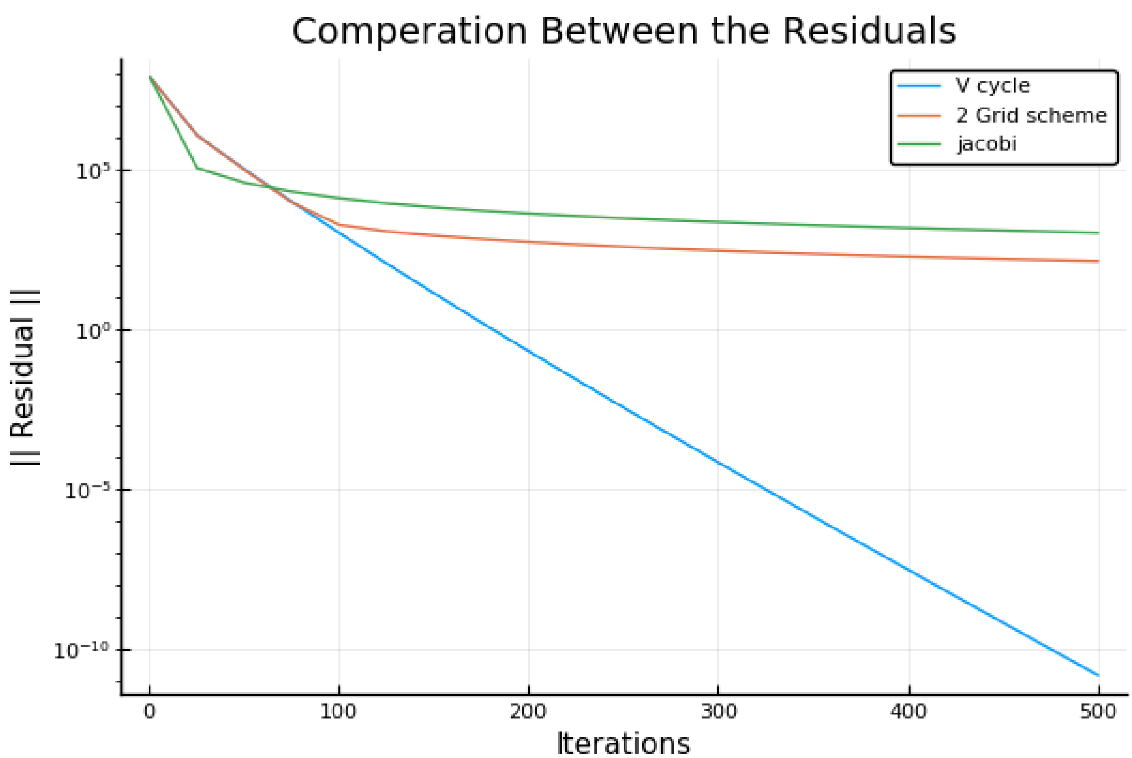

Figure 2 – the residual of all three schemes. At the very beginning it seems as the classic Jacobi will win this race, but the true winner is once again the V Cycle.

Figures 1,2 show that V cycle not only decent first (in residual and error manners) but also converges to the lowest value.
The numbers:

	Final residual norm	Final error norm
Classic Jacobi	1060	3
2 Grid	133	1
V cycle	10^{-11}	10^{-15}

Table 1 – final errors and residuals after the same total number of basic Jacobi steps. A different of several magnitudes between V cycle and the other two schemes.

All three methods converge to a final zero error, obviously V cycle get there faster. After a major decent there is a long way to zero, how long? This question is answered by the convergence rate constants (all three methods have a linear convergence rate).

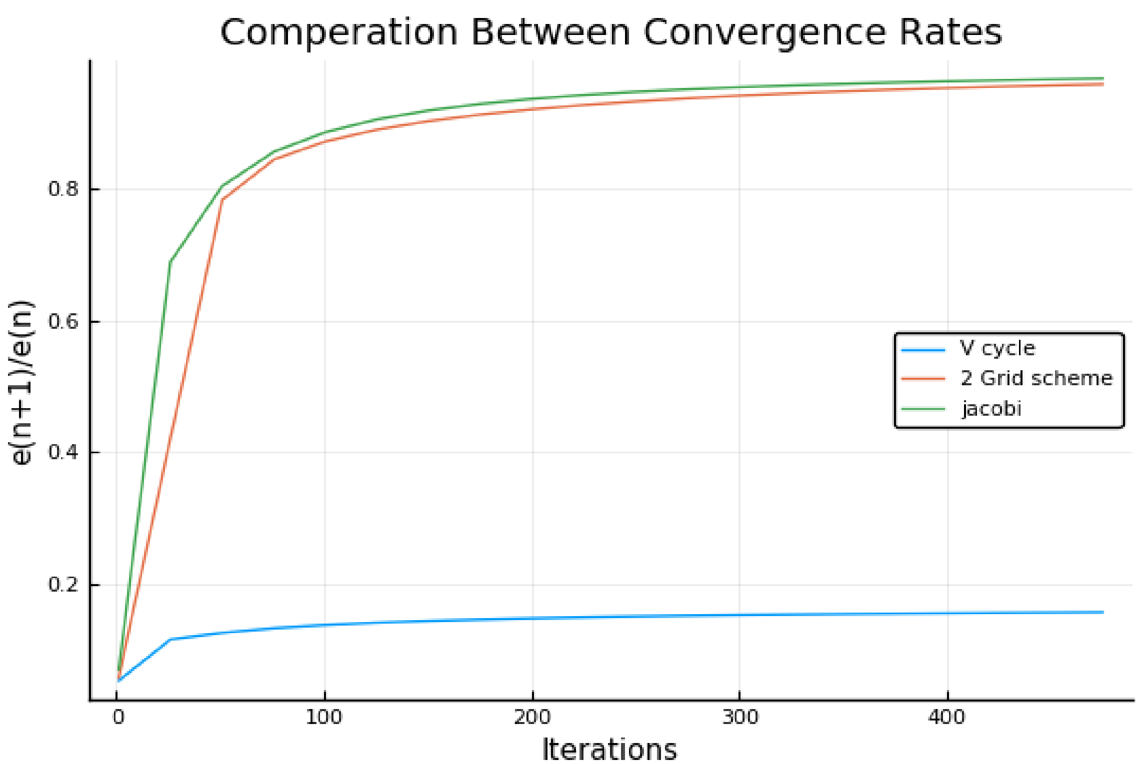

Figure 3 – the ratio between consecutive errors, with close to 1 convergences constant the Jacobi and 2 Grid methods are not practical.

Here it shown best that the V cycle is the winner scheme with $\frac{error(n+1)}{error(n)} \approx 0.2$
Although two-grid correction schemes appear as not practical in this case, it is a proof of concept that can test whether a multigrid scheme can work on a specific problem.

Bibliography

Briggs, William L., and Steve F. McCormick. *A multigrid tutorial*. Vol. 72. Siam, 2000.