MAP55672 (2024-25) — Case studies 4

The Multigrid method

Yu Liu (liuy41@tcd.ie)

4.1 The Poisson Problem

We consider the 2D Poisson equation on the unit square with zero Dirichlet boundary conditions.
Using central finite differences and a uniform grid with step size $h = 1/N$, we discretize the Laplace operator with a 5-point stencil. This leads to a linear system $Ay = b$, where:

- A is a sparse matrix of size $(N-1)^2 \times (N-1)^2$, built using Kronecker products,
- y is the vector of unknowns at interior grid points,
- b is constructed by evaluating the source function $f(x, y) = 2\pi^2 \sin(\pi x)\sin(\pi y)$ at each grid point and scaled by h^2.
- We use row-major ordering to map 2D indices ((i,j)) to the 1D index

$$k = (j-1)(N-1) + (i-1)\,. \tag{1}$$

This setup is the same as in Case Study 3 (Section 3.1).

4.2 Serial implementation of a recursive V-cycle multigrid

We implemented a serial recursive V-cycle multigrid solver for the 2D Poisson problem. The grid levels are controlled by lmax, with the coarsest level fixed at 8×8 to avoid deep recursion.

The smoothing step uses weighted Jacobi with $\omega = 2/3$ and $\nu = 2$, which works well and is easy to implement. The residual restriction uses a 3×3 full-weighting stencil, and the correction is prolongated back to the fine grid by bilinear interpolation.

On the coarsest grid, we solve the error equation $A_c e_c = r_c$ using `spsolve`, so that remaining low-frequency error modes are removed before prolongation. The main loop stops when the residual norm satisfies $|r| < 10^{-7}$.

```
Cycle 0 residual = 1.54e-01
Cycle 1 residual = 1.13e-03
Cycle 2 residual = 4.28e-05
Cycle 3 residual = 3.63e-06
Cycle 4 residual = 3.09e-07
Cycle 5 residual = 2.62e-08
Converged.
Solution complete.
```

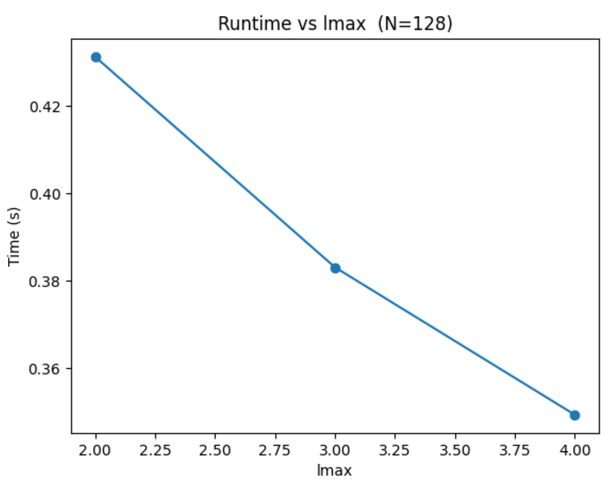
4.3 Convergence of MG.

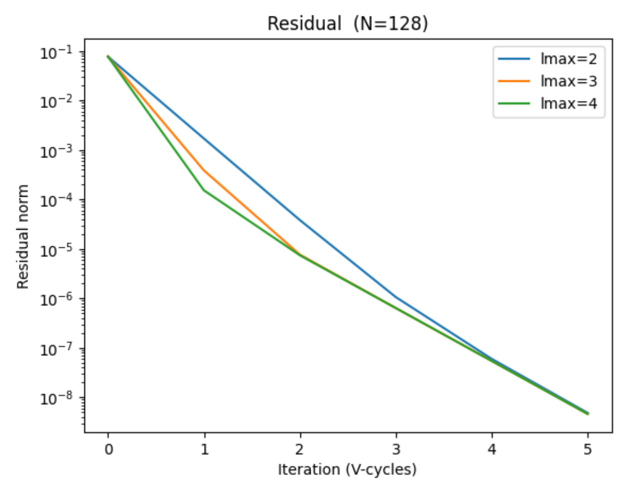
Fix the grid size at N=128, we try different values of lmax=(2, 3, 4) to observe. The stopping condition is a residual norm less than 10^{-7}.

Since N=128, the coarsest grid we can reach is N=8, meaning the maximum valid lamx is 4.

Result

```
  lmax   |  Time (s)   |  Coarse solves
--------------------------------
   2     |    0.4313   |       6
   3     |    0.3831   |       6
   4     |    0.3492   |       6
```

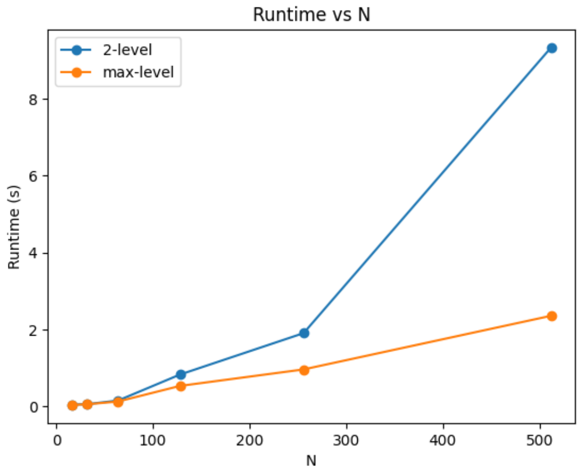


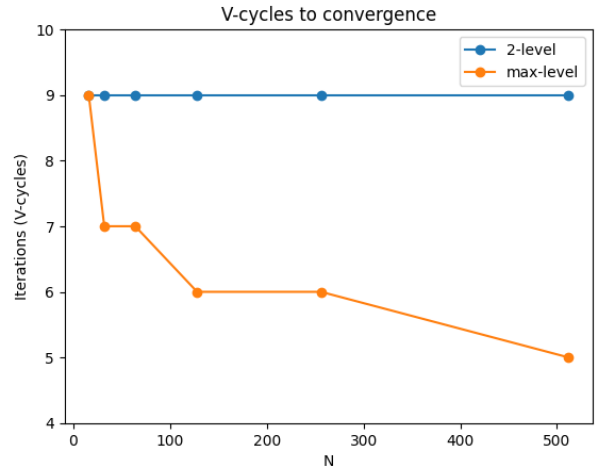


- All setups converge within 6 V-cycles, showing that the error reduction per cycle is high enough for each lmax.
- At $N = 128$, lmax=4 has the fastest convergence. Although all methods eventually converge to 10^{-8}, larger lmax lead to faster residual reduction in the first few V-cycles, indicating that more levels help speed up convergence.

Comparison between 2-Level and max-Level

```
  N     | Time(2lvl) | Cycles(2lvl)  | Res(2lvl)  | Time(max)  |  Cycles(max)  |  Res(max)
---------------------------------------------------------------------------------
  16    |   0.0265   |       9       |  2.89e-08  |   0.0268   |       9       |  2.89e-08
  32    |   0.0422   |       9       |  8.17e-09  |   0.0375   |       7       |  1.33e-08
  64    |   0.1144   |       9       |  3.43e-09  |   0.0887   |       7       |  2.22e-09
 128    |   0.5768   |       9       |  1.64e-09  |   0.3560   |       6       |  4.63e-09
 256    |   1.8829   |       9       |  8.14e-10  |   1.3378   |       6       |  8.19e-10
 512    |   8.1373   |       9       |  4.06e-10  |   3.0369   |       5       |  1.70e-09
```





- For the 2-level MG, the number of iterations stays at 9, regardless of N, meaning the algorithm doesn't improve with larger grids.
- In contrast, the max-level MG shows a reduction in iterations and runtime as N increases, with the residuals reaching $\leq 10^{-7}$.
- Although the number of iterations in 2-level MG does not depend on N, the cost of the direct coarse-grid solve grows rapidly with grid size, making the overall runtime increase roughly like $\mathcal{O}(N^3)$.

Summary

Using a max-level MG setup is a better choice for this problem. It gives fast and stable convergence with lower runtime, while still reaching the same accuracy as the 2-level setup.