Project 3: Scientific Computing

Gianmaria Lucca
MAT: 241440

In this report we analyze the results obtained in **project3.py**.

1. **jacobi_step_1d(uh, fh, omega)**: function that implements one step of the weighted Jacobi method with weight ω.

2. We report the experiments using the previous method with $\omega \in \{\frac{1}{3}, \frac{2}{3}\}$, using as initial guess $\hat{u_h}^{(0)} = 0$ and grid size $N = 2^l$ ($h = \frac{1}{N}$).

   - $\omega = \frac{1}{3}$
     - For $l = 3$: 317 iterations, time = 0.02444 seconds
     - For $l = 4$: 935 iterations, time = 0.11186 seconds
     - For $l = 5$: 2200 iterations, time = 0.21062 seconds
     - For $l = 6$: 4180 iterations, time = 0.49946 seconds
     - For $l = 7$: 13743 iterations, time = 2.60450 seconds
   - $\omega = \frac{2}{3}$
     - For $l = 3$: 169 iterations, time = 0.01291 seconds
     - For $l = 4$: 514 iterations, time = 0.05761 seconds
     - For $l = 5$: 1287 iterations, time = 0.26048 seconds
     - For $l = 6$: 2309 iterations, time = 0.39325 seconds
     - For $l = 7$: 7874 iterations, time = 1.60872 seconds

   As we can observe, as h decreases, the number of iterations and the time are nearly halved using $\omega = \frac{2}{3}$.

3. Consider the two–grid correction method, where we used ten weighted Jacobi steps on the coarser grid instead of solving the associated linear system for the error $\hat{A}_{2h}\hat{e}_{2h} = \hat{r}_{2h}$.

   - $\omega = \frac{1}{3}$
     - For $l = 3$: 310 iterations, time = 0.23722 seconds
     - For $l = 4$: 927 iterations, time = 0.69657 seconds
     - For $l = 5$: 2192 iterations, time = 1.61577 seconds
     - For $l = 6$: 4176 iterations, time = 5.02033 seconds
     - For $l = 7$: 13739 iterations, time = 22.92056 seconds

- $\omega = \frac{2}{3}$
  - For $l = 3$: 165 iterations, time = 0.14705 seconds
  - For $l = 4$: 508 iterations, time = 0.46970 seconds
  - For $l = 5$: 1281 iterations, time = 0.96510 seconds
  - For $l = 6$: 2306 iterations, time = 2.43478 seconds
  - For $l = 7$: 7871 iterations, time = 13.40730 seconds

Comparing the case $\omega = \frac{2}{3}$ with $\omega = \frac{1}{3}$, the time and the number of iterations are halved, as before.
Moreover, comparing the results obtained using the iterated weighted Jacobi method and the two–grid correction scheme, we can observe that the latter needs less iterations in order to satisfy the pseudo–residual stopping criteria, but in general it's more expensive than the iterated Jacobi method (since we do more calls of the Jacobi method).

4. **w_cycle_step_1d(uh, fh, omega, alpha1, alpha2)**: method that performs one W–cycle for the associated linear system using α_1 pre–smoothing steps and α_2 post–smoothing steps.

5. We now report the results obtained using iteratively the W–cycle.

- $\omega = \frac{1}{3},\ \alpha_1 = 1,\ \alpha_2 = 1$
  - For $l = 3$: 15 iterations, time = 0.01185 seconds
  - For $l = 4$: 14 iterations, time = 0.02575 seconds
  - For $l = 5$: 12 iterations, time = 0.07072 seconds
  - For $l = 6$: 11 iterations, time = 0.10238 seconds
  - For $l = 7$: 9 iterations, time = 0.11839 seconds
  - For $l = 8$: 8 iterations, time = 0.16577 seconds
  - For $l = 9$: 6 iterations, time = 0.25186 seconds
  - For $l = 10$: 5 iterations, time = 0.38336 seconds
  - For $l = 11$: 4 iterations, time = 0.59764 seconds
  - For $l = 12$: 3 iterations, time = 0.82027 seconds
  - For $l = 13$: 2 iterations, time = 0.85615 seconds
  - For $l = 14$: 2 iterations, time = 1.77671 seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 1,\ \alpha_2 = 1$
  - For $l = 3$: 8 iterations, time = 0.00518 seconds
  - For $l = 4$: 7 iterations, time = 0.01134 seconds
  - For $l = 5$: 7 iterations, time = 0.02797 seconds
  - For $l = 6$: 7 iterations, time = 0.06347 seconds
  - For $l = 7$: 6 iterations, time = 0.09594 seconds
  - For $l = 8$: 5 iterations, time = 0.11467 seconds
  - For $l = 9$: 5 iterations, time = 0.19929 seconds

- For $l = 10 : 4$ iterations, time $= 0.30179$ seconds
- For $l = 11 : 3$ iterations, time $= 0.41161$ seconds
- For $l = 12 : 3$ iterations, time $= 0.81150$ seconds
- For $l = 13 : 2$ iterations, time $= 0.84284$ seconds
- For $l = 14 : 2$ iterations, time $= 1.75557$ seconds

- $\omega = \frac{1}{3},\ \alpha_1 = 1,\ \alpha_2 = 2$
  - For $l = 3 : 11$ iterations, time $= 0.00867$ seconds
  - For $l = 4 : 10$ iterations, time $= 0.02315$ seconds
  - For $l = 5 : 9$ iterations, time $= 0.04581$ seconds
  - For $l = 6 : 8$ iterations, time $= 0.08801$ seconds
  - For $l = 7 : 7$ iterations, time $= 0.12529$ seconds
  - For $l = 8 : 6$ iterations, time $= 0.23833$ seconds
  - For $l = 9 : 5$ iterations, time $= 0.30467$ seconds
  - For $l = 10 : 4$ iterations, time $= 0.37165$ seconds
  - For $l = 11 : 3$ iterations, time $= 0.51217$ seconds
  - For $l = 12 : 2$ iterations, time $= 0.53705$ seconds
  - For $l = 13 : 2$ iterations, time $= 1.07864$ seconds
  - For $l = 14 : 2$ iterations, time $= 2.24883$ seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 1,\ \alpha_2 = 2$
  - For $l = 3 : 6$ iterations, time $= 0.00251$ seconds
  - For $l = 4 : 6$ iterations, time $= 0.00716$ seconds
  - For $l = 5 : 5$ iterations, time $= 0.01314$ seconds
  - For $l = 6 : 5$ iterations, time $= 0.04382$ seconds
  - For $l = 7 : 4$ iterations, time $= 0.07822$ seconds
  - For $l = 8 : 4$ iterations, time $= 0.10824$ seconds
  - For $l = 9 : 4$ iterations, time $= 0.19415$ seconds
  - For $l = 10 : 3$ iterations, time $= 0.25977$ seconds
  - For $l = 11 : 3$ iterations, time $= 0.51007$ seconds
  - For $l = 12 : 2$ iterations, time $= 0.52623$ seconds
  - For $l = 13 : 2$ iterations, time $= 1.07984$ seconds
  - For $l = 14 : 2$ iterations, time $= 2.23432$ seconds
- $\omega = \frac{1}{3},\ \alpha_1 = 2,\ \alpha_2 = 1$
  - For $l = 3 : 11$ iterations, time $= 0.00503$ seconds
  - For $l = 4 : 10$ iterations, time $= 0.01268$ seconds
  - For $l = 5 : 9$ iterations, time $= 0.03524$ seconds
  - For $l = 6 : 8$ iterations, time $= 0.06872$ seconds
  - For $l = 7 : 7$ iterations, time $= 0.09578$ seconds
  - For $l = 8 : 6$ iterations, time $= 0.14419$ seconds
  - For $l = 9 : 5$ iterations, time $= 0.23667$ seconds

- For $l = 10$: 4 iterations, time = 0.37043 seconds
- For $l = 11$: 3 iterations, time = 0.51884 seconds
- For $l = 12$: 3 iterations, time = 1.03698 seconds
- For $l = 13$: 2 iterations, time = 1.07838 seconds
- For $l = 14$: 2 iterations, time = 2.25933 seconds

- $\omega = \frac{2}{3},\ \alpha_1 = 2,\ \alpha_2 = 1$
  - For $l = 3$: 7 iterations, time = 0.00305 seconds
  - For $l = 4$: 6 iterations, time = 0.01027 seconds
  - For $l = 5$: 6 iterations, time = 0.03457 seconds
  - For $l = 6$: 5 iterations, time = 0.05403 seconds
  - For $l = 7$: 5 iterations, time = 0.09389 seconds
  - For $l = 8$: 4 iterations, time = 0.10591 seconds
  - For $l = 9$: 4 iterations, time = 0.18936 seconds
  - For $l = 10$: 4 iterations, time = 0.37394 seconds
  - For $l = 11$: 3 iterations, time = 0.51313 seconds
  - For $l = 12$: 3 iterations, time = 1.03678 seconds
  - For $l = 13$: 2 iterations, time = 1.07281 seconds
  - For $l = 14$: 2 iterations, time = 2.22828 seconds
- $\omega = \frac{1}{3},\ \alpha_1 = 2,\ \alpha_2 = 2$
  - For $l = 3$: 9 iterations, time = 0.00480 seconds
  - For $l = 4$: 8 iterations, time = 0.01086 seconds
  - For $l = 5$: 7 iterations, time = 0.02966 seconds
  - For $l = 6$: 6 iterations, time = 0.05591 seconds
  - For $l = 7$: 6 iterations, time = 0.08982 seconds
  - For $l = 8$: 5 iterations, time = 0.17923 seconds
  - For $l = 9$: 4 iterations, time = 0.20805 seconds
  - For $l = 10$: 3 iterations, time = 0.29020 seconds
  - For $l = 11$: 3 iterations, time = 0.59298 seconds
  - For $l = 12$: 2 iterations, time = 0.61404 seconds
  - For $l = 13$: 2 iterations, time = 1.41734 seconds
  - For $l = 14$: 2 iterations, time = 2.79822 seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 2,\ \alpha_2 = 2$
  - For $l = 3$: 6 iterations, time = 0.00361 seconds
  - For $l = 4$: 6 iterations, time = 0.01221 seconds
  - For $l = 5$: 5 iterations, time = 0.02409 seconds
  - For $l = 6$: 5 iterations, time = 0.05660 seconds
  - For $l = 7$: 4 iterations, time = 0.07223 seconds
  - For $l = 8$: 4 iterations, time = 0.11543 seconds
  - For $l = 9$: 4 iterations, time = 0.21874 seconds

- For $l = 10$: 3 iterations, time = 0.32473 seconds
- For $l = 11$: 3 iterations, time = 0.61071 seconds
- For $l = 12$: 2 iterations, time = 0.64032 seconds
- For $l = 13$: 2 iterations, time = 1.34002 seconds
- For $l = 14$: 2 iterations, time = 2.81034 seconds

As we can observe, decreasing h (or increasing l) gives us a more refined discretization of the domain: we get that the method needs less iterations to satisfy the stopping criteria, but the CPU time roughly doubles each time we divide h in half.

6. **full_mg_1d(uh, fh, omega, alpha1, alpha2, nu)**: function that performs a full multigrid step, using the previous W–cycle to perform ν W–cycle steps.

7. Let's fix $\omega = \frac{2}{3}$.

- $\alpha_1 = 1,\ \alpha_2 = 1,\ \nu = 1$
  - For $l = 3$: pseudo-residual = 0.0003994699717496447, residual = 0.023026837310095027, time = 0.00057 seconds
  - For $l = 4$: pseudo–residual = 0.0001268425063600375, residual = 0.03167059891296335, time = 0.00154 seconds
  - For $l = 5$: pseudo–residual = 5.4508824234781976e-05, residual = 0.04751511796205619, time = 0.00581 seconds
  - For $l = 6$: pseudo–residual = 1.8946665036470925e-05, residual = 0.07258766604110609, time = 0.01742 seconds
  - For $l = 7$: pseudo–residual = 5.59808539385637e-06, residual = 0.0900020856002904, time = 0.03386 seconds
  - For $l = 8$: pseudo–residual = 1.6040145215862923e-06, residual = 0.10232102727485765, time = 0.08256 seconds
  - For $l = 9$: pseudo–residual = 4.3649769338556174e-07, residual = 0.10963295901968426, time = 0.11593 seconds
  - For $l = 10$: pseudo–residual = 1.1378304332369457e-07, residual = 0.11358806591781626, time = 0.20428 seconds
  - For $l = 11$: pseudo–residual = 2.904215623192449e-08, residual = 0.11560663371812914, time = 0.38911 seconds
  - For $l = 12$: pseudo–residual = 7.335977840629672e-09, residual = 0.11662630003453911, time = 1.12124 seconds
  - For $l = 13$: pseudo–residual = 1.8434803458739326e-09, residual = 0.11713874717444232, time = 1.70492 seconds
  - For $l = 14$: pseudo–residual = 4.6205933642364276e-10, residual = 0.11751107338353393, time = 3.29620 seconds
- $\alpha_1 = 1,\ \alpha_2 = 1,\ \nu = 2$

- For $l = 3$: pseudo–residual = 4.14977848063958e-05, residual = 0.002500562335326525, time = 0.00115 seconds
- For $l = 4$: pseudo–residual = 1.5901950102798976e-05, residual = 0.003486104821883336, time = 0.00308 seconds
- For $l = 5$: pseudo–residual = 6.553727754102061e-06, residual = 0.005542683729586391, time = 0.00813 seconds
- For $l = 6$: pseudo–residual = 2.2008289744675745e-06, residual = 0.008293276362518419, time = 0.02695 seconds
- For $l = 7$: pseudo–residual = 6.396965559975383e-07, residual = 0.010137103714941257, time = 0.06828 seconds
- For $l = 8$: pseudo–residual = 1.724281883739697e-07, residual = 0.011190923550946382, time = 0.10706 seconds
- For $l = 9$: pseudo–residual = 4.475589085114768e-08, residual = 0.011752578673595021, time = 0.18624 seconds
- For $l = 10$: pseudo–residual = 1.1505463855298118e-08, residual = 0.01204230722628907, time = 0.36202 seconds
- For $l = 11$: pseudo–residual = 2.9377301602635797e-09, residual = 0.012211147887850618, time = 0.74618 seconds
- For $l = 12$: pseudo–residual = 7.421927677205963e-10, residual = 0.012358601206497106, time = 1.52563 seconds
- For $l = 13$: pseudo–residual = 1.8652385472479101e-10, residual = 0.012432811241336668, time = 3.17765 seconds
- For $l = 14$: pseudo–residual = 4.6753274544421426e-11, residual = 0.012470037438424572, time = 6.55711 seconds

- $\alpha_1 = 2,\ \alpha_2 = 2,\ \nu = 1$
  - For $l = 3$: pseudo–residual = 4.140510596016729e-05, residual = 0.00526257032048183, time = 0.00138 seconds
  - For $l = 4$: pseudo–residual = 2.2845189877312294e-05, residual = 0.008503857587737713, time = 0.00481 seconds
  - For $l = 5$: pseudo–residual = 8.619970676434704e-06, residual = 0.010574656762928092, time = 0.01709 seconds
  - For $l = 6$: pseudo–residual = 2.6275013598896398e-06, residual = 0.01174881273644171, time = 0.03890 seconds
  - For $l = 7$: pseudo–residual = 7.229965241039964e-07, residual = 0.012373402361686504, time = 0.07755 seconds
  - For $l = 8$: pseudo–residual = 1.8946445887071776e-07, residual = 0.012695317647215414, time = 0.12017 seconds
  - For $l = 9$: pseudo–residual = 4.848388844222451e-08, residual = 0.012858687458459205, time = 0.19731 seconds
  - For $l = 10$: pseudo–residual = 1.2262470059655993e-08, residual = 0.012940973713678042, time = 0.29080 seconds
  - For $l = 11$: pseudo–residual = 3.083415841352425e-09, residual = 0.012982266741823035, time = 0.57950 seconds

- For $l = 12$: pseudo–residual = 7.730856949411343e-10, residual = 0.013002950658940759, time = 1.24921 seconds
- For $l = 13$: pseudo–residual = 1.9355082456467415e-10, residual = 0.013021069782575728, time = 2.48281 seconds
- For $l = 14$: pseudo–residual = 4.842265836959019e-11, residual = 0.013034656594026873, time = 5.19477 seconds

- $\alpha_1 = 2,\ \alpha_2 = 2,\ \nu = 2$
  - For $l = 3$: pseudo–residual = 1.979344479062506e-06, residual = 0.00019622204035232675, time = 0.00206 seconds
  - For $l = 4$: pseudo–residual = 1.0565550559223785e-06, residual = 0.0003364087111888264, time = 0.00584 seconds
  - For $l = 5$: pseudo–residual = 3.6845609844172317e-07, residual = 0.0004123480644659975, time = 0.02000 seconds
  - For $l = 6$: pseudo–residual = 1.0398330784139667e-07, residual = 0.0004425636811607645, time = 0.04916 seconds
  - For $l = 7$: pseudo–residual = 2.7250561975347698e-08, residual = 0.0004543541088742574, time = 0.09440 seconds
  - For $l = 8$: pseudo–residual = 6.949531956737358e-09, residual = 0.00045922429471957345, time = 0.14462 seconds
  - For $l = 9$: pseudo–residual = 1.7530468807277697e-09, residual = 0.00046231388107337095, time = 0.28373 seconds
  - For $l = 10$: pseudo–residual = 4.4012249991428123e-10, residual = 0.00046520920545446937, time = 0.57211 seconds
  - For $l = 11$: pseudo–residual = 1.1025692856002182e-10, residual = 0.00046662483363774164, time = 1.17542 seconds
  - For $l = 12$: pseudo–residual = 2.759211976466943e-11, residual = 0.0004673244683678354, time = 2.45842 seconds
  - For $l = 13$: pseudo–residual = 6.901490656903691e-12, residual = 0.00046767222024351015, time = 5.05593 seconds
  - For $l = 14$: pseudo–residual = 1.7258036701648695e-12, residual = 0.00046784557620540884, time = 10.55916 seconds

  As we can observe, the CPU time grows like the order of the discretization: what's interesting is the fact that the residual defined as $|r_h|_\infty = |f_h - A_h u_h|_\infty$ seems to grow as l grows.

8. Using a full multigrid step over a grid with $l = 14,\ \alpha_1 = 2,\ \alpha_2 = 2$ and $\nu = 2$, we get that

   $\min_{x\in\overline{\Omega}} u(x) \approx -0.006408964457407074$

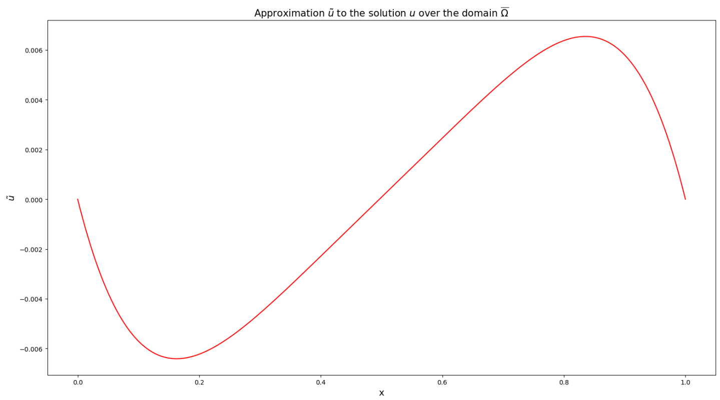

Figure 1: Plot of the approximation $\tilde{u}$ of the elliptic boundary problem in the domain $\overline{\Omega} = [0,\ 1]$

Let's now consider the elliptic problem in 2D.

1. **jacobi_step_2d(uh, fh, omega)**: function that implements one step of the weighted Jacobi method with weight ω for the 2D problem.

2. We report the results obtained for the Jacobi method in 2D.

   - $\omega = \frac{1}{3}$
     - For $l = 2$: 104 iterations, time $=$ 0.02227 seconds
     - For $l = 3$: 369 iterations, time $=$ 0.14774 seconds
     - For $l = 4$: 1278 iterations, time $=$ 0.72446 seconds
     - For $l = 5$: 4304 iterations, time $=$ 9.21846 seconds
     - For $l = 6$: 13976 iterations, time $=$ 108.43262 seconds
   - $\omega = \frac{2}{3}$
     - For $l = 2$: 53 iterations, time $=$ 0.01178 seconds
     - For $l = 3$: 196 iterations, time $=$ 0.07062 seconds
     - For $l = 4$: 689 iterations, time $=$ 0.57743 seconds
     - For $l = 5$: 2354 iterations, time $=$ 4.58964 seconds
     - For $l = 6$: 7799 iterations, time $=$ 58.99173 seconds

3. **w_cycle_step_2d(uh, fh, omega, alpha1, alpha2)**: method that performs one W–cycle for the associated linear system using α_1 pre–smoothing steps and α_2 post–smoothing steps, for the 2D problem.

4. We report below the results for the W–cycle method in 2D.

- $\omega = \frac{1}{3},\ \alpha_1 = 1,\ \alpha_2 = 1$
  - For $l = 2$: 28 iterations, time = 0.02443 seconds
  - For $l = 3$: 31 iterations, time = 0.08099 seconds
  - For $l = 4$: 29 iterations, time = 0.21159 seconds
  - For $l = 5$: 26 iterations, time = 0.53172 seconds
  - For $l = 6$: 23 iterations, time = 1.61273 seconds
  - For $l = 7$: 20 iterations, time = 4.04239 seconds
  - For $l = 8$: 16 iterations, time = 12.55293 seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 1,\ \alpha_2 = 1$
  - For $l = 2$: 14 iterations, time = 0.00994 seconds
  - For $l = 3$: 15 iterations, time = 0.03285 seconds
  - For $l = 4$: 15 iterations, time = 0.12227 seconds
  - For $l = 5$: 14 iterations, time = 0.27033 seconds
  - For $l = 6$: 12 iterations, time = 0.61115 seconds
  - For $l = 7$: 11 iterations, time = 2.47350 seconds
  - For $l = 8$: 9 iterations, time = 6.49378 seconds
- $\omega = \frac{1}{3},\ \alpha_1 = 1,\ \alpha_2 = 2$
  - For $l = 2$: 19 iterations, time = 0.01884 seconds
  - For $l = 3$: 21 iterations, time = 0.08890 seconds
  - For $l = 4$: 20 iterations, time = 0.17393 seconds
  - For $l = 5$: 18 iterations, time = 0.35672 seconds
  - For $l = 6$: 16 iterations, time = 1.23578 seconds
  - For $l = 7$: 13 iterations, time = 3.23221 seconds
  - For $l = 8$: 11 iterations, time = 10.31302 seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 1,\ \alpha_2 = 2$
  - For $l = 2$: 10 iterations, time = 0.01018 seconds
  - For $l = 3$: 11 iterations, time = 0.04382 seconds
  - For $l = 4$: 11 iterations, time = 0.13351 seconds
  - For $l = 5$: 10 iterations, time = 0.22939 seconds
  - For $l = 6$: 9 iterations, time = 0.68925 seconds
  - For $l = 7$: 8 iterations, time = 2.16696 seconds
  - For $l = 8$: 6 iterations, time = 5.93974 seconds
- $\omega = \frac{1}{3},\ \alpha_1 = 2,\ \alpha_2 = 1$
  - For $l = 2$: 19 iterations, time = 0.02110 seconds
  - For $l = 3$: 21 iterations, time = 0.07337 seconds
  - For $l = 4$: 20 iterations, time = 0.26555 seconds
  - For $l = 5$: 18 iterations, time = 0.60394 seconds
  - For $l = 6$: 16 iterations, time = 1.07503 seconds

- For $l = 7$: 14 iterations, time = 3.61020 seconds
- For $l = 8$: 12 iterations, time = 11.35092 seconds

- $\omega = \frac{2}{3},\ \alpha_1 = 2,\ \alpha_2 = 1$
  - For $l = 2$: 10 iterations, time = 0.00866 seconds
  - For $l = 3$: 11 iterations, time = 0.04784 seconds
  - For $l = 4$: 11 iterations, time = 0.11463 seconds
  - For $l = 5$: 10 iterations, time = 0.20715 seconds
  - For $l = 6$: 9 iterations, time = 0.58909 seconds
  - For $l = 7$: 8 iterations, time = 2.10458 seconds
  - For $l = 8$: 7 iterations, time = 6.36822 seconds
- $\omega = \frac{1}{3},\ \alpha_1 = 2,\ \alpha_2 = 2$
  - For $l = 2$: 15 iterations, time = 0.01588 seconds
  - For $l = 3$: 16 iterations, time = 0.06809 seconds
  - For $l = 4$: 15 iterations, time = 0.14386 seconds
  - For $l = 5$: 14 iterations, time = 0.33677 seconds
  - For $l = 6$: 12 iterations, time = 1.08482 seconds
  - For $l = 7$: 11 iterations, time = 3.41312 seconds
  - For $l = 8$: 9 iterations, time = 10.27822 seconds
- $\omega = \frac{2}{3},\ \alpha_1 = 2,\ \alpha_2 = 2$
  - For $l = 2$: 8 iterations, time = 0.01163 seconds
  - For $l = 3$: 9 iterations, time = 0.05286 seconds
  - For $l = 4$: 9 iterations, time = 0.18392 seconds
  - For $l = 5$: 8 iterations, time = 0.21532 seconds
  - For $l = 6$: 7 iterations, time = 0.57660 seconds
  - For $l = 7$: 6 iterations, time = 1.88686 seconds
  - For $l = 8$: 5 iterations, time = 5.36701 seconds

5. **full_mg_2d(uh, fh, omega, alpha1, alpha2, nu)**: function that performs a full multigrid step, using the previous W–cycle to perform ν W–cycle steps.

6. Fix $\omega = \frac{2}{3}$, we analyze the various experiments for the full multigrid step.

   - $\alpha_1 = 1,\ \alpha_2 = 1,\ \nu = 1$
     - For $l = 2$: pseudo-residual = 0.004305747812216005, residual = 0.22599991173417705, time = 0.00075 seconds
     - For $l = 3$: pseudo-residual = 0.0025709078339824602, residual = 0.5628578922936097, time = 0.00321 seconds
     - For $l = 4$: pseudo-residual = 0.0008994620531890159, residual = 0.7738109547915525, time = 0.01735 seconds
     - For $l = 5$: pseudo-residual = 0.0002595453573218595, residual = 0.8940216948309869, time = 0.04711 seconds

- For $l = 6$: pseudo-residual = 6.959795653692976e-05, residual = 0.952338030037239, time = 0.14613 seconds
- For $l = 7$: pseudo-residual = 1.8022995325181566e-05, residual = 0.9820487975935376, time = 0.37901 seconds
- For $l = 8$: pseudo-residual = 4.583893296431136e-06, residual = 0.9970493068734292, time = 1.23228 seconds

- $\alpha_1 = 1,\ \alpha_2 = 1,\ \nu = 2$
  - For $l = 2$: pseudo-residual = 0.001278655587176155, residual = 0.0694729340066049, time = 0.00092 seconds
  - For $l = 3$: pseudo-residual = 0.0008283561626986381, residual = 0.18607056023899862, time = 0.00453 seconds
  - For $l = 4$: pseudo-residual = 0.0002778885576506362, residual = 0.2525923899443304, time = 0.01809 seconds
  - For $l = 5$: pseudo-residual = 7.741561985401615e-05, residual = 0.27861337735643, time = 0.08279 seconds
  - For $l = 6$: pseudo-residual = 2.02135892714006e-05, residual = 0.2895958504059103, time = 0.18652 seconds
  - For $l = 7$: pseudo-residual = 5.139720782050723e-06, residual = 0.29360632896525574, time = 0.67593 seconds
  - For $l = 8$: pseudo-residual = 1.2953733077720716e-06, residual = 0.2956007710882094, time = 2.37831 seconds
- $\alpha_1 = 2,\ \alpha_2 = 2,\ \nu = 1$
  - For $l = 2$: pseudo-residual = 0.0012929906600869998, residual = 0.07091795504805944, time = 0.00114 seconds
  - For $l = 3$: pseudo-residual = 0.0008710461318469885, residual = 0.20281563183402854, time = 0.00640 seconds
  - For $l = 4$: pseudo-residual = 0.00028854853702039995, residual = 0.2691828581691314, time = 0.02073 seconds
  - For $l = 5$: pseudo-residual = 8.038778986181432e-05, residual = 0.2976440311175773, time = 0.08209 seconds
  - For $l = 6$: pseudo-residual = 2.099312341478487e-05, residual = 0.3095180189420283, time = 0.24408 seconds
  - For $l = 7$: pseudo-residual = 5.347472038711668e-06, residual = 0.3147251493961746, time = 0.56678 seconds
  - For $l = 8$: pseudo-residual = 1.3484600592719026e-06, residual = 0.3170379319562989, time = 1.85953 seconds
- $\alpha_1 = 2,\ \alpha_2 = 2,\ \nu = 2$
  - For $l = 2$: pseudo-residual = 0.0001364256741170078, residual = 0.00755401950614782, time = 0.00155 seconds
  - For $l = 3$: pseudo-residual = 0.0001323846979462695, residual = 0.03149757219038721, time = 0.00732 seconds

- For $l = 4$: pseudo-residual = 4.291967167566885e-05, residual = 0.04140544091379722, time = 0.04224 seconds
- For $l = 5$: pseudo-residual = 1.1588437568684287e-05, residual = 0.04475414910049826, time = 0.11930 seconds
- For $l = 6$: pseudo-residual = 2.9693945954728518e-06, residual = 0.04582512804864678, time = 0.28366 seconds
- For $l = 7$: pseudo-residual = 7.487259435557704e-07, residual = 0.046194248579346765, time = 0.98483 seconds
- For $l = 8$: pseudo-residual = 1.8779678185145213e-07, residual = 0.04633360370655317, time = 4.13719 seconds

As we have seen before, the residual increases as l increases.

7. Using a full multigrid step over a grid with $l = 8$, $\alpha_1 = 2$, $\alpha_2 = 2$ and $\nu = 2$, we get that

$\min_{x \in \overline{\Omega}} u(x) \approx -0.0117486346379127 65$

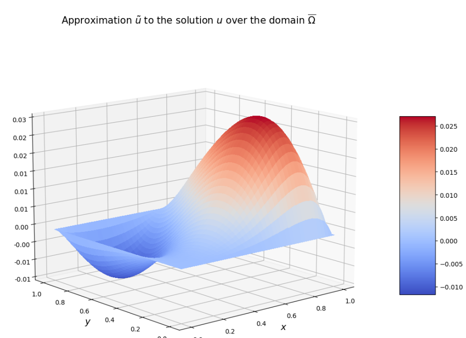

Figure 2: Plot of the approximation $\tilde{u}$ of the elliptic boundary problem in the domain $\overline{\Omega} = [0,\ 1] \times [0,\ 1]$