Studying the Finite Difference Methods (FDMs) for the Laplace Equation

Computational Physics (P346) | Instructor: *Dr. Subhasis Basak*[1] | DIY Project

Palla Shubhang Sharma (2011106)[1]

[1] *School of Physical Sciences, National Institute of Science Education and Research (NISER), HBNI, Jatni-752050, India*

(Dated: December 1, 2022)

A partial differential equation (PDE) involves derivatives of higher order and of multiple variables. Approximating these derivatives upon discretizing the domain under which the DE is required to solve occupy the essentials of a Finite Difference Method. It comes with a major shortcoming though – solutions to the PDE in an irregular domain are beyond its scope, although few generalizations exist. In the following report the Laplace equation is solved numerically in regular domains (rectangles) $\mathcal{Q} \subset \mathbb{R}^2$ using the FDMs with three improvements namely, Gauss-Seidel (GS), Successive over-relaxation (SOR), Multigrid methods (MG). The reason why FDM is widely used and remains important is its simplicity for regular domains in the sense that they use vastly general iterative methods which under crucial improvements as listed above perform exponentially better. Thus to extend the solutions to PDEs as good as possible retaining simplicity this method is studied here. Finite Element method is an immediate successor to this method, which carries the same notion of discretization but applicable to irregular domains and involves approximation functions in the elements instead of derivatives.

Keywords: : FDM, Laplace Equation, Gauss-Seidel, Numerical methods, Harmonic functions

I. INTRODUCTION

Laplace Equation in $\mathbb{R}^2$, under Cartesian coordinate system reads,

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

Such solutions U to the above PDE are a more general class of functions called harmonic functions, which upon given boundary conditions (marked blue in the figure-(1)) take a specific form.

We consider a rectangular domain $\mathcal{Q}$ and divide the region into a discretized region consisting of points separated by distance h.

Upon discretizing the region into fine h one can approximate the derivative of a function as a finite difference from the neighbouring functional values.

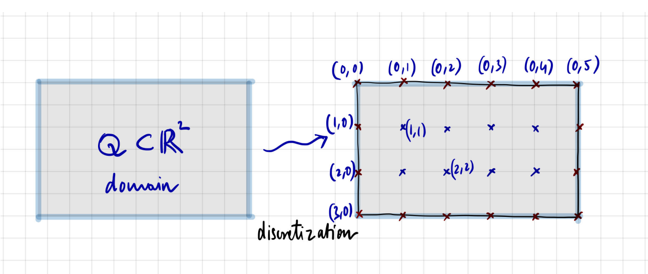


FIG. 1: The points in the discretized domain are labelled via (i,j); $0 \le i \le l_x$ and $0 \le j \le l_y$; where l_x, l_y completely specify the rectangular domain. Each point (i, j) is essentially a point $(x = i.h, y = j.h) \in Q \subset \mathbb{R}^2$. We take $h_x = h_y = h$ as the refinement of the grid.

Taking a finite difference we thus have,

$$\frac{\partial U}{\partial x}|_{x,y} = \frac{U(x+h,y) - U(x,y)}{h} \tag{2}$$

Similarly evaluating $\frac{\partial^2 U}{\partial x^2}$ and substituting in (1) yields,

$$U_{i,j} = \frac{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}}{4} \tag{3}$$

The above equation essentially reflects the *Mean Value Theorem* for solutions to Laplace Equation (Harmonic Functions).

II. NUMERICAL METHODS

A. Gauss-Seidel (GS)

U, solution to the Laplace equation is stored as a matrix of dimensions $n_x + 1 \times n_y + 1$ (where $n_x = l_x/h, n_y = l_y/h$, i.e number of divisions in X, Y axes). We first give guess values (typically 0) to the solution U $\forall i, j$ and then using equation-(6) we iteratively calculate the $U_{i,j}$ using the latest updated values. All the methods adopted here solve by iteration.

```
if abs(deltaU) < e:  #convergence check
    conv += 1

u[i][j] = 0.25 * (u[i+1][j] + u[i-1][j] +
                  u[i][j+1] + u[i][j-1])
```

B. Successive over-relaxation (SOR)

The iterative solution indicated above take large iterations to converge, one can force the changes or so called doing an over-'relaxation' by adding a term proportional to the change in U for an iteration weighted by a relaxation parameter. The code below summarizes the same.

```
deltaU = 0.25*(u[i+1][j] + u[i-1][j] + u[i
    ][j+1] + u[i][j-1]) - u[i][j]

if abs(deltaU) < e:  #convergence check
    conv += 1

u[i][j] = relax_par*deltaU + u[i][j]
                        #U_new = U_old +
                            relax_par * deltaU
```

Numerical Analysis studies prove that relax_par $= \frac{2}{1+\frac{\pi}{N}}$, where N is the dimension of the matrix $\approx n_y$. The same was used in the calculations.

This drastically reduces the iteration number from Gauss-Seidel.

C. Multigrid Method (MG)

The time complexity quickly increases with the dimension of the matrix taken, i.e for a large domain or highly fine $h << 1$. To improve the iteration number further, we coarse grain the domain into larger grids of refinement say H and iteratively estimate the solution U. We use this known estimate as a guess to the values in a domain of refinement $H/2$, recursively performing this coarse-graining as guess to the fine-grained iteration, the iteration number falls to very low number.

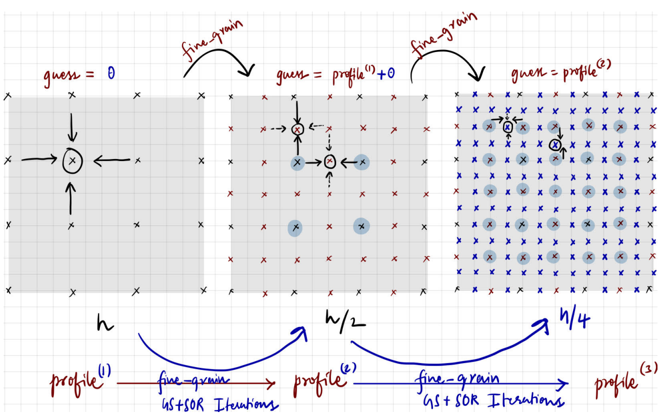


FIG. 2: The above picture summarizes the functions fine_grain and Laplace_2D_SOR_MG written in the FDM_Library.py

III. RESULTS

All the plots are under the periodic boundary conditions, which can be looked at the project repository in FDM_library.py where b_func holds the given boundary values. The results depicted in the figure-(4) and figure-(5) show a drastic improvement in iteration number when compared to the plain Gauss-Seidel. When one carefully analyses the refinement h in the results of Multigrid Method. One can actually notice a large improvement in sometimes iterations and mostly the time required. It was noticed that, under constant boundary conditions, in the domain $lx = 1 = ly$, Multigrid beats SOR exponentially (MG time $\approx 5sec$, SOR time $\approx 25sec$). Second set of plots are given because, it is noticed that the MG function doesn't consistently work with different h values, under 2π domain h was required to π and for $lx = 1$ domain h to be 1 an integer.

We thus have the FDM methods ready to be applied for a given problem, under smooth conditions.

TABLE I: Iteration Count for various h values occurred in Multigrid Method with relaxation parameter $c = 5$

h	π	$\pi/2$	$\pi/4$	$\pi/8$	$\pi/16$	$\pi/32$
GS + SOR + MG	1	15	25	45	85	159

TABLE II: Iteration Count for various h values occurred in Multigrid Method with relaxation parameter $c = 6$

h	1/2	1/4	1/8	1/16	1/32	1/64	1/128
GS + SOR + MG	10	15	24	45	83	158	309

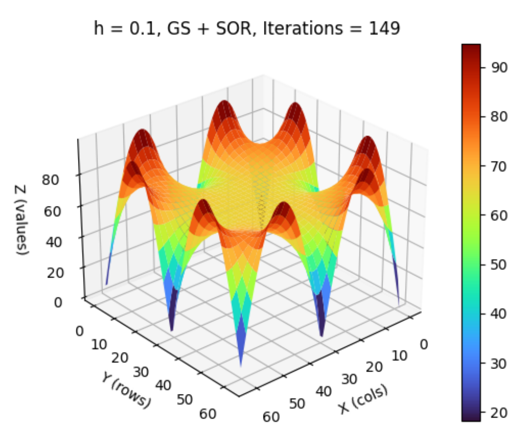
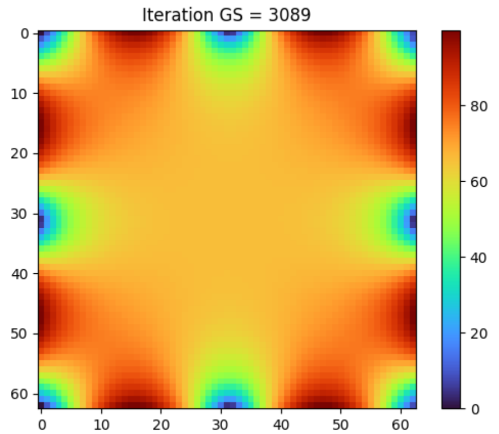
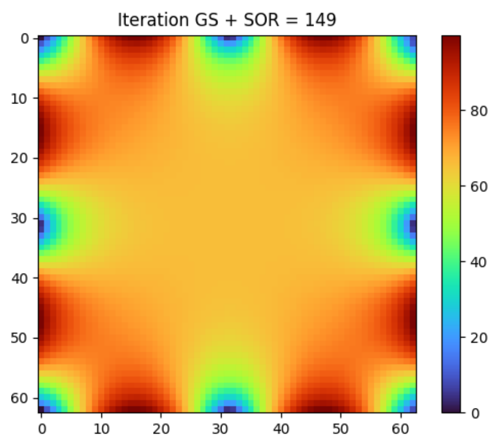


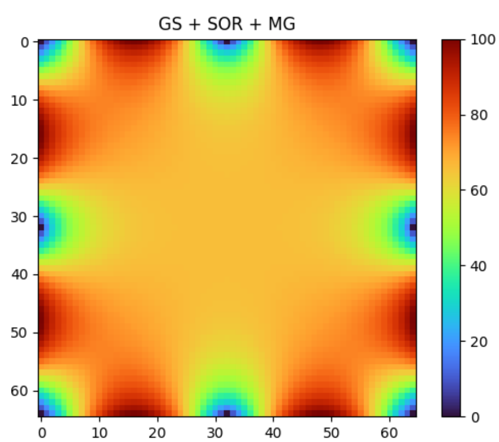
FIG. 3: The 3D plot of the solution to Laplace equation with periodic boundary conditions.



(a) $h = 0.1$, Domain: $l_x = 2\pi = l_y$

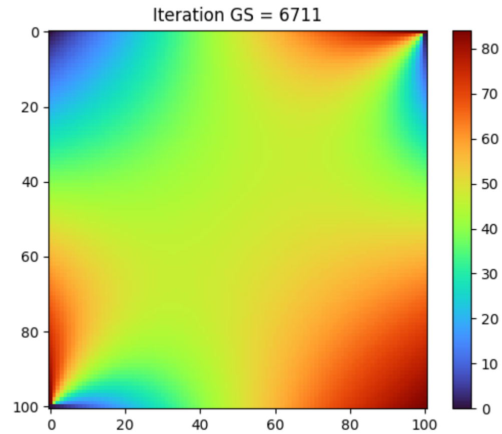


(b) $h = 0.1$, Domain: $l_x = 2\pi = l_y$

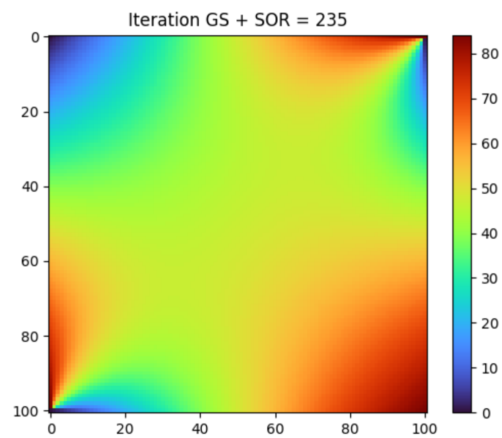


(c) $h = \pi$, $c = 5$, Domain: $l_x = 2\pi = l_y$ (Refer TABLE-I)

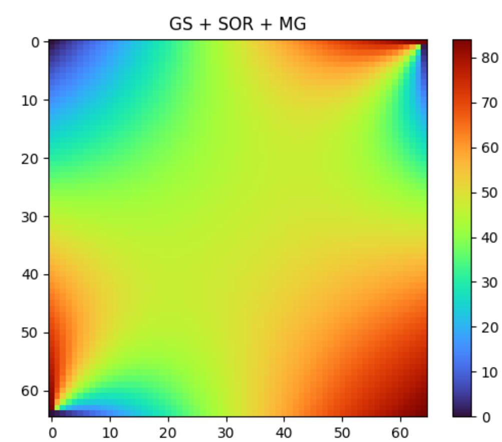
FIG. 4



(a) $h = 0.01$, Domain: $l_x = 1 = l_y$



(b) $h = 0.01$, Domain: $l_x = 1 = l_y$



(c) $h = 1$, $c = 7$, Domain: $l_x = 1 = l_y$ (Refer TABLE-II)

FIG. 5

IV. REFERENCES

1. https://gist.github.com/CMCDragonkai/dd420c0800cba33142505eff5a7d2589

2. Successive Over-Relaxation, Timothy Jones

3. Relaxation (iterative method). (2022, November 27). In Wikipedia. https://en.wikipedia.org/wiki/Relaxation_(iterative_method)