Math4315 Spring 2023 **Project 2** due: Wed, March 22, 2023 by noon

Project Description

This project studies solving 2-D boundary value problems with finite difference schemes and iterative methods. Consider a metal plate on the unit square $D = \{(x, y) : 0 \leq x, y \leq 1\}$. The temperature $\phi(x, y)$ satisfies the Poisson's equation

$$\phi_{xx} + \phi_{yy} = f(x, y) \tag{1}$$

on D with boundary conditions $\phi(x, 0) = \cdots$, $\phi(x, 1) = \cdots$, $\phi(0, y) = \cdots$, $\phi(1, y) = \cdots$. We can solve for the temperature using the finite-difference scheme $D_+^x D_-^x w_{ij} + D_+^y D_-^y w_{ij} = f_{ij}$ with mesh size $h = \frac{1}{n+1}$. This yields a linear system $A_h w_h = f_h$, where $w_h = \{w_{ij}\}$ is the numerical solution with components $w_{ij} \approx \phi(x_i, y_j)$. The mesh points are given by $x_i = ih, y_j = jh, i, j = 0 : n + 1$. Your work is composed three parts.

Part I: Pick a function $\phi(x, y)$ on your own and generate $f(x, y)$, $\phi(x, 0)$, $\phi(x, 1)$, $\phi(0, y)$, $\phi(1, y)$ based on your pick. The picked function should be smooth e.g. the combination of trigonometric function, exponential, logarithmic, etc. Avoid simply polynomial or function with singularities in the computational domain e.g. $\log(x - 0.5)$. You need to provide $\phi(x, y)$, $f(x, y)$, and the boundary conditions in your project report.

Part II: Use PYTHON to solve the linear system by Jacobi's method with $h = \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{64}$. Take $w_{ij}^{(0)} = 0$ at step $k = 0$. Take $||r_k||_\infty / ||r_0||_\infty \leq 10^{-8}$ for the stopping criterion, where $r_k = f_h - A_h w_h^{(k)}$ is the residual at step k. Note we use "$||r||_\infty = \max(\texttt{abs}(r))$" in PYTHON to return the infinite norm of r (why?) since r is a vector-in-matrix form. The abs() and max() function should be loaded from the numpy function. To keep the code simple, code the numerical solution $w_h = \{w_{ij}\}$ as a matrix of dimension $(n + 2) \times (n + 2)$ containing the unknown interior temperature values and the known boundary values. Hence in PYTHON the numerical solution has the form $w(i, j)$. For $i = 0 : n+1, j = 0 : n+1$, use the component form of Jacobi's method,

$$\frac{1}{h^2}(4w_{ij}^{(k+1)} - w_{i+1,j}^{(k)} - w_{i-1,j}^{(k)} - w_{i,j+1}^{(k)} - w_{i,j-1}^{(k)}) = f_{ij} \tag{2}$$

and solve for $w_{ij}^{(k+1)}$ using loops over i and j. Do not form the full matrix A_h (because it is sparse and that would waste memory). You will need two versions of the vector w_{ij}, one for the current step $(k + 1)$ and one for the previous step (k). For this part, you result will include one figure composed of six sub-figures and a table as described below.

(1) For each value of h from $\frac{1}{4}, \frac{1}{8}$, and $\frac{1}{16}$, plot the computed temperature w_{ij} (including the boundary values) at the final step using a contour plot and a mesh plot (type `help contour` and `help mesh` in PYTHON for instructions). Use `subplot` to get several graphs on one plot, as in the lecture notes.
(2) Find the errors of the numerical solution by comparing with the true values and study the order of accuracy (error=max(abs(w-$w$true))). A tabular output of four columns and six rows is required as:

h	error	ratio	order
$\vdots$	$\vdots$	$\vdots$	$\vdots$

where ratio=error($2h$)/error(h) and order=$\log_2$(ratio).

(3) Give a brief writeup (use the Markdown of Jupyter notebook) to describe your results, connecting to the theory. For example, give a column by column description of the data pattern in the table, predict the asymptotic behavior and explained why.

Part II: Repeat Part I with Gauss-Seidel's method and SOR method ($\omega_* = \frac{2}{1+\sqrt{1-\cos^2(\pi * h)}}$). You need to first figure out the component form of both methods as in Eq. (2). You can check errors and graphs to make sure the coding is correct. However, for this part, you only need to submit a 6 by 4 tabular data of iteration numbers of all three methods at different h. Describe the table and explain the pattern.

h	Jocobi	G-S	SOR
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Project Specification

Your project report should be written with Jupyter noterbook including the following components as described. It is also okay to use a Word document to include all required information and save it as a pdf file.

(1) General information: project title, course, date, name, affiliation (school and department/program)

(2) Project Description: Use your own language to briefly describe the project including the problem you try to solve and the numerical methods you use.

(3) Concepts and Theory: describe the physical problem you try to solve, with the differential equation, boundary values, and parameters listed, as well as the analytical solution.

(4) Numerical Results and Discussion: show the graphs and tables obtained (one figure with 6 subplots) and discuss what you have observed, explain data patterns by connecting to (3). Carefully choose the format and how many significant digits after decimal point to keep for the data in the table.

(5) Concluding Remarks: summarize what you have done, what you learned and raise your questions if there are any.

Note: please submit your report via canvas with the provided link. The report should be converted into a pdf file with the filename: **p4315_yourlastname**. This file name is also your submission title on canvas. Please submit your code as well since I might need to run it to figure out some problems.

Jacobi:

$$w_{ij}^{(k+1)} = \frac{1}{4}((w_{i+1,j}^{(k)} + w_{i-1,j}^{(k)} + w_{i,j+1}^{(k)} + w_{i,j-1}^{(k)}) + f_{ij}h^2) \tag{3}$$

G-S:

$$w_{ij}^{(k+1)} = \frac{1}{4}((w_{i+1,j}^{(k)} + w_{i-1,j}^{(k+1)} + w_{i,j+1}^{(k)} + w_{i,j-1}^{(k+1)}) + f_{ij}h^2) \tag{4}$$

SOR:

$$\frac{4}{h^2}w_{ij}^{(k+1)} = \frac{4}{h^2}w_{ij}^{(k)} + \omega^*(\frac{1}{h^2}(4w_{ij}^{(k)} - w_{i+1,j}^{(k)} - w_{i-1,j}^{(k+1)} - w_{i,j+1}^{(k)} - w_{i,j-1}^{(k+1)}) - f_{ij}) \tag{5}$$

thus

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} + \frac{1}{4}\omega^*(4w_{ij}^{(k)} - w_{i+1,j}^{(k)} - w_{i-1,j}^{(k+1)} - w_{i,j+1}^{(k)} - w_{i,j-1}^{(k+1)} - f_{ij}h^2) \tag{6}$$

Note you need to use two version of w for Jacobi's method but only one version of w for G-S and SOR methods (when updating w_{ij}, all needed are ready in w).