數學流體力學之理論與計算

Homework Assignment 2

Due date: Prob.1-5 on Oct. 26.
Prob.6 on Nov. 9.

Part I: Theoretical assignments

Problem 1. In class we have checked that

$$\text{curlcurl}u = -\Delta u + \nabla \text{div} u$$

if the space dimension is 3. State and prove a similar identity for the two dimensional case. Remember that when the space dimension is 2, the curl operator is defined as

$$\text{curl}u = \frac{\partial u^2}{\partial x_1} - \frac{\partial u^1}{\partial x_2}\,.$$

Part II: Numerical assignments

Problem 2. Modify the function "identify_points" from the last assignment so that it also take 3 as the value of periodic_type. When periodic_type is 3, one identifies the left and right boundaries, and identifies the top and bottom boundaries. In other words, when periodic_type is 3, the domain under consideration is a periodic box.

Problem 3. From the last assignment, one already has two matrices "int_ext" and "adjacent_pts" that might contain wrong information when periodic_type is 1 or 2. For example, suppose that one discretize the unit square as in the following figure

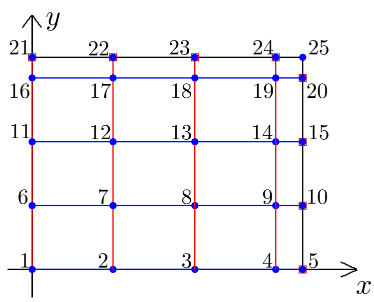


Figure 1: A discretization of the unit square

and identifies the left and right boundaries; that is, periodic_type = 1. From the last assignment one should obtain int_ext(6) = int_ext(10) = int_ext(11) = int_ext(15) = int_ext(16) = int_ext(20)= 0, while after identifying the left and right boundaries, these six numbers should be 1 which indicates that 6th, the 10th, the 11th, the 15th, the 16th and the 20th intersection points are interior points instead of boundary points. Moreover, from the last assignment one should also obtain

$$\text{adjacent\_pts}(6,:) = [7\ 11\ 0\ 1]$$
$$\text{adjacent\_pts}(10,:) = [0\ 15\ 9\ 5]$$

and etc, while since the 6th and the 10th intersection points are now interior points, the correct adjacent points of these two points should be

$$\text{adjacent\_pts(6,:)} = \text{adjacent\_pts(10,:)} = [7\ 11\ 9\ 1].$$

Complete the following.

1. Write a matlab® function "correct_int_ext" which, when evaluating at the wrong "int_ext", will produce the correct "int_ext".
2. Write a matlab® function "correct_adjacent_points" which, when evaluating at the wrong "adjacent_pts", will produce the correct "adjacent_pts".

Note that these two functions also depends on the variable "periodic_type". Moreover, you should expect that your code works for any other given data (Later on we will provide data such as "position", "int_ext", "adjacent_pts" and so on, and you will be asked to work on these data directly).

Problem 4. Suppose that "position", "int_ext", and corrected "adjacent_pts", "identify_pts" are given (not necessary those from discretizing the unit square). Complete the the following.

1. Suppose a (discrete) vector field u_val is given on the intersection points. In other words, if the size of the "position" matrix is $N \times 2$, then u_val is also an $N \times 2$ matrix whose i-th row is the vector associated with the i-th intersection point. Write a matlab® program plot_u_val to plot the discrete vector field on the intersection points. You will need the matlab® command **quiver** to do the job.
2. Suppose a scalar function p_val is given on the intersection points. Write a matlab® program plot_p_val to plot the discrete scalar field on the intersection points. You will need the matlab® command **fill** to do the job.

The outputs of these two functions should look like the following if the domain under consideration is the unit square:

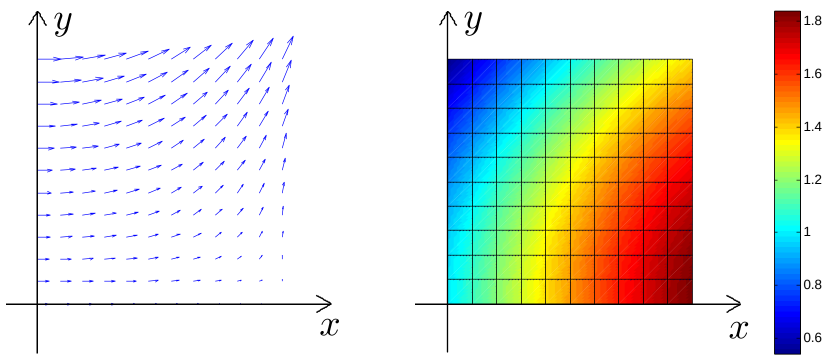
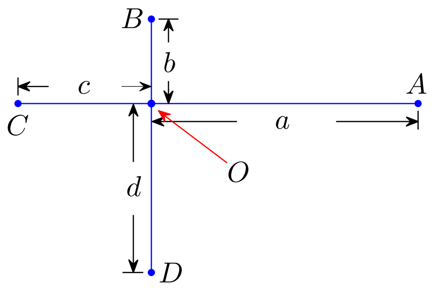


Figure 2: The discrete vector field u_val$(x, y) = (y^2 - x^2, 2xy)$ and scalar field p_val$(x, y) = \sin x + \cos y$ on the unit square

If you have already finished Problem 6, you may want to try your codes for this case.

Problem 5. Complete the following.

1. Suppose that the adjacent points of a point O is A, B, C, D (in a counterclockwise order), and $\overline{OA} = a$, $\overline{OB} = b$, $\overline{OC} = c$, $\overline{OD} = d$.



Compute the discrete version of Δu at O using the finite difference method; that is, find α, β, γ, δ such that

$$\Delta u(O) \approx \alpha u(A) + \beta u(B) + \gamma u(C) + \delta u(D) + \eta u(O)\,.$$

Note that α, β, γ, δ, η should depend on a, b, c, d.

2. Suppose that "position", "int_ext", and corrected "adjacent_pts", "identify_pts" are given (not necessary those from discretizing the unit square). Form the discrete laplace operator. In other words, suppose that p_val denotes the restriction of a function p on all the intersection points, find a matrix lap_fdm such that

lap_fdm * p_val

denotes the discrete value of Δp on all the **interior** intersection points.

3. Solve the Poisson equation

$$\begin{aligned} -\Delta p = f \qquad &\text{in} \quad D\,, \\ p = g \qquad &\text{on} \quad \partial D \end{aligned}$$

numerically under the following settings:

   (a) $D = [0,1] \times [0,1]$, mesh_size $= 0.1$, $f = 0$ and $g = x + y$.

   (b) $D = \mathbb{T} \times [0,1]$; that is, identify the left and right boundaries of $[0,1] \times [0,1]$, mesh_size $= 0.1$, $f = y\sin(\pi x)$ and $g = \sin(\pi x) + y$.

You only need to plot p using the function "plot_p_val" you just developed.

If you have already finished Problem 6, you may want to try your codes for this case.

Problem 6. In this problem we discretize the unit disk. Let mesh_size be a given positive number which is less than 1, and G be a system of grid lines on the x-y plane in which the distance between two closest parallel lines is mesh_size. Write a matlab® function "genmesh_fdm_disk" with input "mesh_size" and outputs "position", "int_ext", "adjacent_pts"in which the meanings of the variables are as follows:

1. Let n be the number of intersection points of grid lines of G inside the unit disk, and m be the number of points where the grid lines of G intersect with the unit circle. The first output "position" is an $(n + m) \times 2$ matrix in which each row vector is the coordinates of one of the $(n + m)$ intersection points.

2. The second output "int_ext" is an $(n+m) \times 1$ column vector whose entries have values 0 or 1. The value of "int_ext(j)", the j-th component of the column vector, is 1 if the j-th intersection point belongs to the interior of the unit disk, and is 0 if the j-th intersection point is on the unit circle.

3. The j-th row of the matrix "adjacent_pts" is the label of the adjacent intersection points (defined in previous assignment) in counter-clockwise order, starting from the right adjacent point.

For example, the following figure is for the case that mesh_size = 0.3. Then n = 37 and m

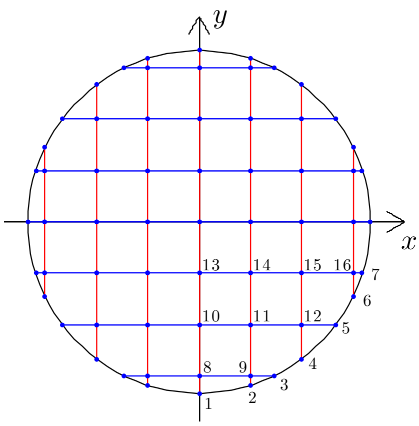


Figure 3: A discretization of the unit square

= 28. Suppose that one labels part of the intersection points as shown in the figure above, then

(a) The matrix "position(1:16,:)", or equivalently the first 16 rows of the position matrix, is the coordinates of these 16 intersection points.

(b) The matrix "int_ext(1:16,:)", or equivalently the first 16 components of the matrix, is [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1]′.

(c) According to the definition (of the matrix "adjacent_pts"), we must have

$$\text{adjacent\_pts}(9,:) = [3\ 11\ 8\ 2];$$
$$\text{adjacent\_pts}(11,:) = [12\ 14\ 10\ 9];$$
$$\text{adjacent\_pts}(12,:) = [5\ 15\ 11\ 4];$$

and

$$\text{adjacent\_pts}(2{:}6,:) = \begin{bmatrix} 0 & 9 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 12 & 0 & 0 \\ 0 & 0 & 12 & 0 \\ 0 & 16 & 0 & 0 \end{bmatrix}.$$

Hint: You may want to use the matlab® command **find** or **intersect** to simplify your code.