ID2090 : Introduction to Scientific Computing
Summer-2021

Assignment – 5

Instructions for submission remain same as for first assignment. Don't forget to provide your name, roll number and assignment number on the first page of the PDF you are uploading. Each question carries 5 marks.

[1] Take a selfie and load it as an array. Make appropriate datatype conversions and plot contours of this array using a grayscale as colormap. Adjust the settings till it looks like a pensil sketch with minimal number of lines. Extract the path of these contour lines from the properties of the children of the graphics object (gca). Confirm that the sketch can be reproduced by plotting these paths.

Output required: The octave code, the original image and the one that looks like pensil sketch, a report containing any assumptions made.

Application: Image manipulation is important for not only the entertainment industry but also for computer vision and automated operations in a plant. Such codes need to be fast and reliable if they should be applied on a live video. Object detection, edge detection, metrics on moving objects etc., are part of industrial automation.

[2] A potential function has been defined as follows:

$\Phi = \frac{-A\cos\theta}{r}$

The velocity components of a flow field are defined by the gradient of this function. Make a quiver plot of the flow field represented by this function. You are welcome to do the gradient calculation numerically. The value of the constant A can be chosen in a way that captures the flow field properly.

Output required: The octave code, a report showing representative snapshots of the quiver plot.

Application: Complex flow patterns in nature and in industrial situations can be understood as a combination of few model flow fields such as source, sink, doublet, vortex etc. Understanding these patterns helps in recognizing the reasons for such flow in the actual problem. Flow fields control heat transfer, mass transfer and even stress on the walls that bound their domain. Understanding and controlling these fields is an important aspect of multiple engineering domains.

[3] Consider the following data:

List	X	`[1.1, 2.2, 3.3, 4.4, 5.5]`
String	S	`"Authentication code for this file is XzmBqr"`
Struct	P	`{model:"Avrami"; n:4; A:1.05; system:"sample binary"}`
Matrix	Q	$\begin{bmatrix} 1.1 & 1.2 & 1.3 \\ 2.1 & 2.2 & 2.3 \\ 3.1 & 3.2 & 3.3 \end{bmatrix}$

Create these in a python code, find a way to write them to a data file (eg., mydata.dat). Create an octave file that reads mydata.dat and prints out the data as well as the list of variables. Confirm this

with a similar output from the python code to ensure that the process of writing and reading the data is accurate. The file mydata.dat should be self describing so that the octave code **need not assume** what is present in the file before reading it. You can assume the above four are the only types of data that would be present but not necessarily in any order or multiplicity.

Output required: The octave and python codes, a report illustrating how the data has been transferred correctly.

Application: Data format for exchange of information between tools is an important aspect of scientific computing. Commercial tools are often blackboxes and precompiled codes but need to provide functionalities such as this to enable users to exchange information across tools. Self describing formats are important as they eliminate manual intervention. Emerging standards such as HDF5 allow for a common format on how this can be done in a tool/language/platform agnostic manner.

--oo**oo--