Assignment-2

Q2) Write a code for the surface roughness prediction in FDM parts and display the same as colour map on the surface

Step1: Convert your .stl file into .txt file, at the same time create one empty spread sheet with any name, now upload both files in to Matlab.Now Execute below mentioned code in
your command window prompt

```
data = readtable(‘your filename.txt');
writetable(data, ‘your filename.xlsx');
```

Step2: Execute below mentioned code in command window for finding the Surface roughness.

```
% Load the Excel file into MATLAB as a table
filename = ' your filename.xlsx';
sheet = 1;
myTable = readtable(filename, 'Sheet', sheet);

% Set the column name in the table
myTable.Properties.VariableNames{1} = 'MyColumn';

% Write the modified data back to the Excel file
writetable(myTable, filename, 'Sheet', sheet);

% Read the data from the Excel file
data = readtable(filename);

% Pad the values in each row with NaNs
maxValues = max(cellfun(@(x) numel(strsplit(x)), data.MyColumn));
paddedData = cellfun(@(x) [strsplit(x) repmat({NaN}, 1, maxValues-numel(strsplit(x)))], data.MyColumn, 'UniformOutput', false);

% Split the padded data into separate columns
splitData = vertcat(paddedData{:});

% Create a new table with the split data
newData = table(splitData(:,1), str2double(splitData(:,2)), str2double(splitData(:,3)), 'VariableNames', {'StringColumn',
'ValueColumn1', 'ValueColumn2'});

% Filter rows containing "facet"
filteredData = splitData(contains(splitData(:,1), 'facet'), :);

% Convert the second column of splitData to text
textData = string(splitData(:, 2));

% Filter rows containing "normal"
filteredData = splitData(contains(textData, 'normal'), :);

% Convert matrix or array to table
filteredData = array2table(filteredData);

% Rename variables
filteredData.Properties.VariableNames = {'v', 'w', 'x', 'y', 'z'};

% Create new variables with same length as existing columns
k = zeros(size(filteredData.v));
l = zeros(size(filteredData.w));
m = 80 * ones(size(filteredData.x));

% Add new variables to table
filteredData.k = k;
filteredData.l = l;
filteredData.m = m;
```

```
% Initialize new column with zeros
filteredData{:, end+1} = zeros(size(filteredData, 1), 1);

for i = 1:size(filteredData, 1)
% Get vectors x, y, z and k, l, m
vector1 = [filteredData{i, 'x'}, filteredData{i, 'y'}, filteredData{i, 'z'}];
vector2 = [filteredData{i, 'k'}, filteredData{i, 'l'}, filteredData{i, 'm'}];

% Convert any cell arrays to numeric arrays
if iscell(vector1)
vector1 = cell2mat(vector1);
end
if iscell(vector2)
vector2 = cell2mat(vector2);
end
% Make sure the vectors have the same length by padding with zeros if necessary
if numel(vector1) > numel(vector2)
vector2(end+1:numel(vector1)) = 0;
elseif numel(vector2) > numel(vector1)
vector1(end+1:numel(vector2)) = 0;
end

% Calculate dot product and norms
dotProduct = sum(vector1 .* vector2);
norm1 = norm(double(vector1));
norm2 = norm(double(vector2));

% Calculate the angle in radians and convert it to degrees
angleInDegrees = rad2deg(acos(dotProduct / (norm1 * norm2)));

% Calculate R based on angle value
if angleInDegrees == 0
R = 118.07 * 10^3 * 0.25;
elseif angleInDegrees > 0 && angleInDegrees < 30
R = ((-2.442*10^-4)*angleInDegrees^4+(0.134*10^-1)*angleInDegrees^3-(1.777*10^-
1)*angleInDegrees^2+(2.1)*angleInDegrees-1.681*10^-1)*((0.25*1)/sind(angleInDegrees));
elseif angleInDegrees > 30 && angleInDegrees < 90
R = 71.472*((0.25*1)/sind(angleInDegrees));
else
R = NaN;
end% Save the angle and R values in the new columns
filteredData{i, 9} = angleInDegrees;
filteredData{i, 10} = R;end
% Save the filtered data to a new Excel sheet
filteredFilename = 'filtered_data.xlsx';
writetable(filteredData, filteredFilename, 'Sheet', 1);

% Display a message to indicate the process is complete
disp('Processing complete.');
% Extract the first 50 R and angle values from the filteredData table
R_values = filteredData{1:50, 10};
angle_degrees = filteredData{1:50, 9};

% Convert angle values to radians
angle_radians = deg2rad(angle_degrees);

% Define the grid for the scatter plot
[X,Y] = meshgrid(1:5,1:10);

% Reshape R_values to match the grid
Z = reshape(R_values, [10, 5]);

% Create the scatter plot
```

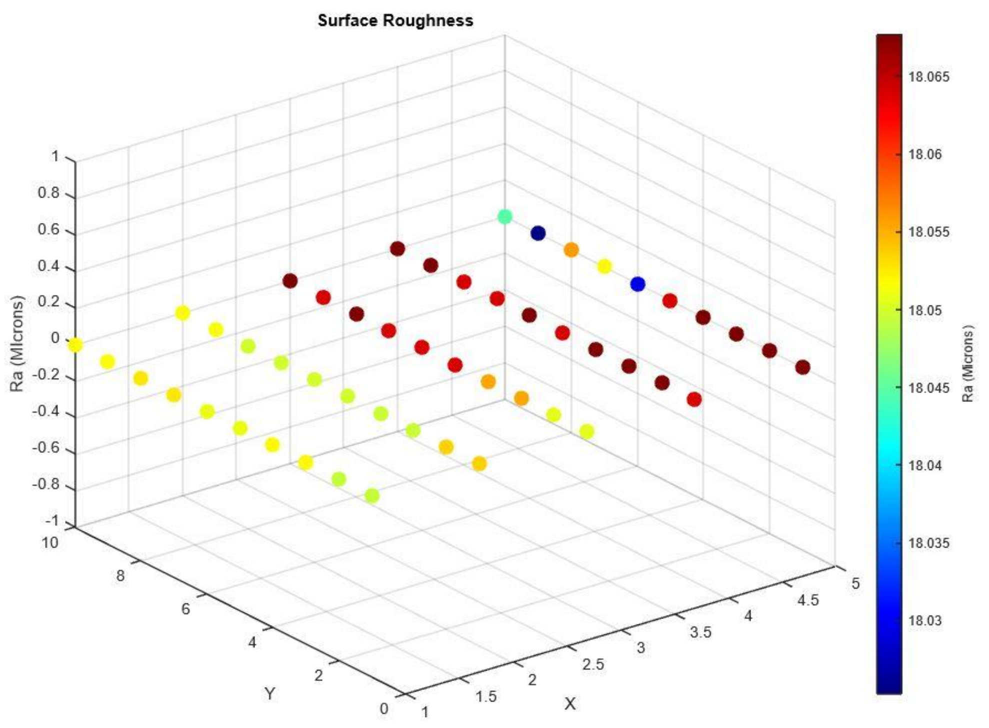
```
scatter3(X(:), Y(:), zeros(numel(Z), 1), 100, Z(:), 'filled');

% Set axis labels
xlabel('X');
ylabel('Y');
zlabel('R value');

% Set colormap
colormap(jet);

% Add colorbar
c = colorbar;
c.Label.String = 'R value';
```

Result: **Figure:**Surface Roughness values mapping



Name: K Sowjanya
Roll No: MM22M023

.... **END** ....