```matlab
function[madesubplot] = Visualizesection(M_in,zbot,ztop,n,Version)
%This function requires inputs(1: nx5 Matrix with z,r,Theta,MC,GC, 2,3:
%z-range around center (+- tolerance), 4: Number of Slices, 5: Version).
%There are 4 Version options which make different plots. The Number of
%slices are used in two of the plots to slice the data into a number of
%sections with similar z value. Each Version, when executed, makes two
%plots, one with MC and one with GC.

%For all Versions except 2 (which uses cartesian coordinates), the
%circumference angle Theta is displayed in degrees, not Rad:
if Version ~= 2
    M_in(:,3) = rad2deg(M_in(:,3));
end

% Version 4 displays a 3D surface made of [z Theta Curvature]. This is essentially
% a "Central Cylindrical Projection" which ignores r to map 3D geometric
% properties to 2D and then adds the parameter of curvature as the new z
% component. To view this new surface in 2D, it is colored according to MC
% or GC and viewed in -z direction (from above). The result is a rectangle
% colored according to curvature, which gives a good overview of curvature
% distribution and trends, while curvature itself is not really
% quantifiable and only displayed through color.
    if Version == 4
        Msorted = sortrows(M_in,1); %Polar coordinates data, sort rows by z
        Mindices = Msorted(:,1)<ztop & Msorted(:,1)>zbot;
        %Select only the specified region of interest around the neck. M_GC
        %has nothing to do with Gaussian curvature.
        M_GC = Msorted(Mindices,:);
        %Call the function slicefiguremakesurface with M_GC and p.
        figure;
        %This function makes the cylindrical projection and plots
        %it.
        slicefiguremakesurface(M_GC,zbot,ztop,4);
        figure;
        slicefiguremakesurface(M_GC,zbot,ztop,5);
    end

% Version 2 makes 3D Pointclouds with GC and MC Coloring, giving the most
% direct representation of the data. This figure can be compared to the
% visualization in Amira to see if the sample is represented accurately.
% The third entry refers to the point size rather than the number of
% slices. Cartesian data is used insted of polar.
    if Version == 2
        Msorted = sortrows(M_in,3);
        Indices = Msorted(:,3)<ztop & Msorted(:,3) > zbot;
        %Select only the specified region of interest around the neck. M_GC
        %has nothing to do with Gaussian curvature.
        M_GC = Msorted(Indices,:);
        %Extract values from the big matrix for better overview
        x = M_GC(:,1);
        y = M_GC(:,2);
        z = M_GC(:,3);
        MC = M_GC(:,4);
        GC = M_GC(:,5);
        %This figure uses scatter3 to make a colored pointcloud with a
        %point size of n and a colormap for MC. The caxis is the color
        %dimension, which uses the predefined boundaries for MC.
```

```
        figure;
        scatter3(x,y,z,n,MC,'filled');
        colormap('jet'); % Choose a colormap (e.g., jet, parula, etc.)
        colorbar; % Add colorbar to show the "dim" values
        title('Mean curvature');
        axis equal;
        caxis([MC_min MC_max])%('auto');
        zlabel('z');
        xlabel('x');
        ylabel('y');
        %The same for GC with the appointed color boundaries for GC
        figure;
        scatter3(x,y,z,n,GC,'filled');
        colormap('jet'); % Choose a colormap (e.g., jet, parula, etc.)
        colorbar; % Add colorbar to show the "dim" values
        title('Gaussian curvature');
        axis equal;
        caxis([GC_min GC_max])%('auto');%
        zlabel('z');
        xlabel('x');
        ylabel('y');
    end

% Version 3 takes the polar data and separates it into n sections of equal
% deltaz. In these sections, the Curvature values are averaged and a 2D
% graph is plotted: MC(z) or GC(z). For each section, the min, mean and max
% value are plotted.
    if Version == 1
        Msorted = sortrows(M_in,1);
        Mindices = Msorted(:,1)<ztop & Msorted(:,1)>zbot;
        %Select only the specified region of interest around the neck. M_GC
        %has nothing to do with Gaussian curvature.
        M_GC = Msorted(Mindices,:);
        %The function "slicefigureCOL1" is used to plot Curvature(Theta).
        %It calls the function "compressionCOL1" which takes M_GC between
        %zbot and ztop and turns it into a cell array with n slice
        %matrixes, each matrix having z-values within a certain range. The
        %resulting cell array is then processed in a loop and for each of
        %the n slice matrices, n plots are made in the same figure of
        %Curvature(Theta). Two figures are made, one or MC and one for GC.
        slicefigureCOL1(M_GC,zbot,ztop,n);
    end

% Version 3 takes the polar data and separates it into n sections of equal
% deltaz. In these sections, the Curvature values are averaged and a 2D
% graph is plotted: MC(z) or GC(z). For each section, the min, mean and max
% value are plotted.
     if Version == 3
        Msorted = sortrows(M_in,1);
        Mindices = Msorted(:,1)<ztop & Msorted(:,1)>zbot;
        %Select only the specified region of interest around the neck. M_GC
        %has nothing to do with Gaussian curvature.
        M_GC = Msorted(Mindices,:);
        %The function "meanvalslicefigure" is used to plot Curvature(z).
        %It calls the function "slicefigureCOL1" which takes M_GC between
        %zbot and ztop and turns it into a cell array with n slice
        %matrixes, each matrix having z-values within a certain range. The
        %resulting cell array is then processed in a loop and for each of
```

```
        %the n slice matrices. For every slice matrix, the min, mean and
        %max for both GC and MC are stored, and then plotted in two
        %figures, one or MC and one for GC.
        meanvalslicefigure(M_GC,zbot,ztop,n);
    end
end
```