Proceedings of Science and Mathematics
Faculty of Science, Universiti Teknologi Malaysia
Vol. 11, 2022, page 12 - 25

Visualization of 3D Objects by Using MATLAB

Norazlin Ahmad Kamal, Taufiq Khairi Ahmad Khairuddin*
Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia
*Corresponding author: taufiq@utm.my

Abstract
3D visualization is an important process in interpreting real-life objects. In this study, MATLAB software is used to visualize 2D and 3D graphs, and reconstruct 2D and 3D shapes while studying the importance of symmetries in visualization. 2D visualization was first explored by plotting two equations in the Cartesian coordinate. 2D plot based on symmetrical property of the graph was then conducted. Meanwhile, 3D visualization was investigated based on a surface. A set of data that consists a few coordinates of the surface in the first octant was imported in MATLAB to partially reconstruct the surface. The whole shape was then constructed by using symmetries. The symmetries of the plot were studied and the importance of symmetries in 3D visualization were highlighted. The reconstructed 2D shapes and 3D objects shown that MATLAB software could be used as an alternative to interactively visualize 3D objects.

Keywords: 2D Visualization, 3D Visualization, MATLAB

1. Introduction

Visualization becomes possible for practically all applications using this type of software. On the other hand, the 3D graphics sector is working on effective solutions to the majority of the problems associated with large-scale and dynamic data representations. The use of dynamism in visualisation is greatly strengthened by 3D technology. High-quality real-time rendering techniques have enabled interactive exploration and interactive animated representations. To create a sensation of virtual reality, the capacity to change 3D perspective in real-time is essential. The use of 3D in visualisation can benefit from advanced 3D visual mapping tactics and methodologies.

Recently, 3D visualization has been used widely in science and technology to produce a 3D digital representation of any object or surface. Notably, a few researchers are interested in the study of 3D visualization software due to its important role in modern development. However, there is no research that involves visualization of 3D objects by using MATLAB and at the same time taking account into the importance symmetries in visualizing 3D objects. Therefore, this study will develop the visualization of 3D objects using MATLAB and explore the importance of symmetries in objects. The scope was focused on the application of MATLAB programming as tools for objects visualization and reconstruction based on the symmetries. This study explores the command interface to show how to visualize scientific data using MATLAB graphics commands which consists of 2D and 3D plots with major visualization techniques. The symmetry of the objects is obtained by applying the symmetry concept into the calculation in MATLAB programming. The symmetry may be observed with respect to the axes.

2. Literature Review

2.1. MATLAB for Visualization

There are several tools in MATLAB for visualising data and information. For wire-frame objects, space-curves, and shaded surfaces, there are built-in charting functions. Automatic contour generation, volumetric data presentation, light sources, colour interpolation, and externally created image display are all provided. Above the graphic programme interface of the local host, MATLAB provides an abstract graphics layer. This enables cross-platform interoperability and creates a device-independent graphics layer. In MATLAB, graphics objects are used to create visual representations of data. There are two types of graphics objects in the MATLAB graphics paradigm: Core graphics objects and Composite graphics objects. The main graphics objects include basic drawing primitives such as line, text, rectangles, patches (filled polygons), surfaces (3D grid of vertices), pictures (2D matrix representation of an image), light sources, and axes (define the coordinate system). Composite graphics objects are made up of core graphics objects that have been combined to make them more user-friendly.

2.2. Symmetries of Objects

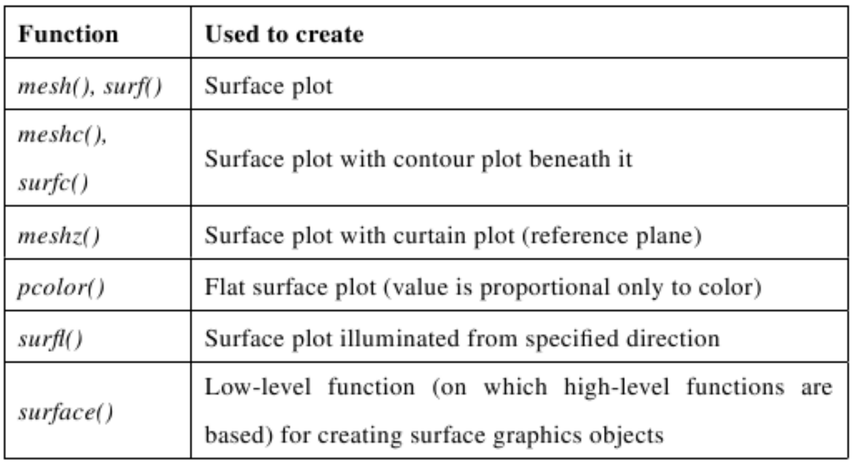
In geometry, symmetry is defined as an action or transformation that transfers a figure or object onto itself. Reflectional symmetry, rotation symmetry, translational symmetry, and glide reflection symmetry are some of the symmetry types examined in fundamental geometry. Symmetry of a 3D object facilitates shape constancy across changes in viewpoint [8]. Symmetrical shapes or figures are created by drawing a line so that the representations on both sides of the line mirror each other. A line of symmetry is an imaginary line or axis along which a figure can be folded to produce symmetrical halves. There could be one or more symmetry lines, and they could be vertical, horizontal, or diagonal. However, there are no lines of symmetry in three-dimensional objects. Three-dimensional objects, on the other hand, have a symmetry plane, which is a two-dimensional area. A balanced and proportionate likeness between two sides of an item is characterised as symmetry in geometry. It means one half is the polar opposite of the other.

Symmetry is a general concept in mathematics [9]. A definition of symmetry in mathematics is that whether one shape is moved, rotated, or flipped, it is identical to the other shape. Asymmetrical shapes are those that aren't symmetrical. Symmetrical shapes can be seen in nature, architecture, and art. Psychological studies suggest that symmetry plays a crucial role in the human visual perception system [10, 11]. Finding symmetry in such geometric data is thus an important topic in geometry processing that has recently received a lot of attention. The symmetry information in shapes has benefited a lot of current technology.

3. Methodology

3.1. 2D & 3D Visualization

MATLAB tools and functions that can be utilised in 2D and 3D visualization includes plot, scatter, surf and mesh functions.



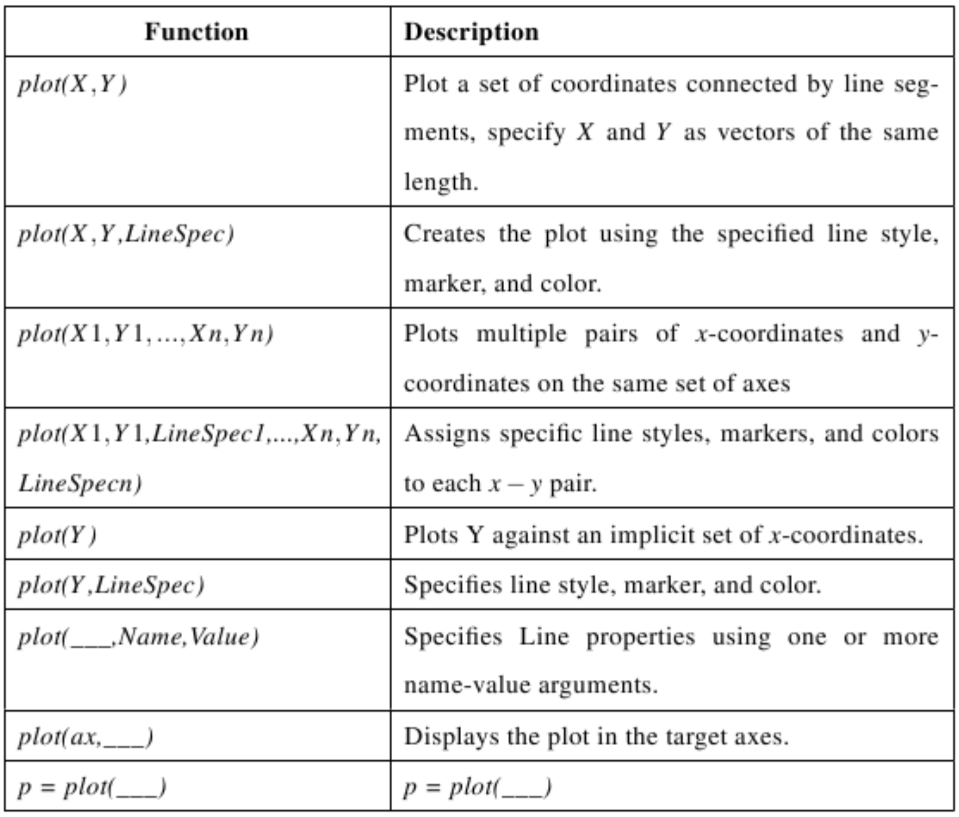
Function	Used to create
mesh(), surf()	Surface plot
meshc(), surfc()	Surface plot with contour plot beneath it
meshz()	Surface plot with curtain plot (reference plane)
pcolor()	Flat surface plot (value is proportional only to color)
surfl()	Surface plot illuminated from specified direction
surface()	Low-level function (on which high-level functions are based) for creating surface graphics objects

Figure 1 Scatter plot functions in MATLAB.

Mesh and surf commands will be explored to create 3-D surface plots of matrix data. If Z is a matrix with the entries $Z(i,j)$ defining the height of a surface over an underlying (i,j) grid, we will be using $mesh(Z)$ to generate a coloured, wire-frame image of the surface and shows it in a three-dimensional view. Meanwhile $Surf(Z)$ will be used to create a colorful, faceted picture of the surface and presents it in three dimensions. Additional control over the visual appearance of the surface is provided via surface object characteristics.

3.2. Symmetries in 2D Plots

Various functions can be used to plot many types of plots in MATLAB. In this study, function $plot(x,y)$ will be used mostly to create a 2D line plot of the equation with y versus the corresponding values in x. From function $plot(x,y)$, many plots customization can be made.



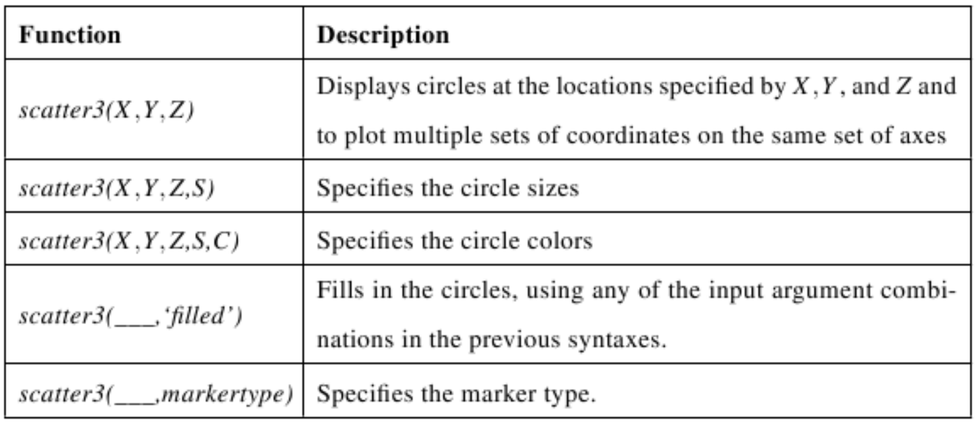
Function	Description
plot(X,Y)	Plot a set of coordinates connected by line segments, specify X and Y as vectors of the same length.
plot(X,Y,LineSpec)	Creates the plot using the specified line style, marker, and color.
plot(X1,Y1,...,Xn,Yn)	Plots multiple pairs of x-coordinates and y-coordinates on the same set of axes
plot(X1,Y1,LineSpec1,...,Xn,Yn, LineSpecn)	Assigns specific line styles, markers, and colors to each $x-y$ pair.
plot(Y)	Plots Y against an implicit set of x-coordinates.
plot(Y,LineSpec)	Specifies line style, marker, and color.
plot(___,Name,Value)	Specifies Line properties using one or more name-value arguments.
plot(ax,___)	Displays the plot in the target axes.
p = plot(___)	*p = plot(___)*

Figure 2: 2D Plot Functions in MATLAB

By using functions in the table above, symmetry with respect to x-axis and y-axis can be obtained correspondingly to visualize the whole shape for 2D visualization.

3.3. Symmetries in 3D Plots.

Visualization of 3D objects is obtained by proposing a set of data that only consists of the first octant of the object. Then the symmetric of the object are tested by using the symmetries where MATLAB function $scatter3()$ will be used in other to plot the coordinates in the data set. The data set must first need to be imported as a matrix. $scatter3()$ is a function in MATLAB that allow a set of coordinates to be plotted in 3D by specifying x, y, and z as a matrix.



Function	**Description**
scatter3(X,Y,Z)	Displays circles at the locations specified by X, Y, and Z and to plot multiple sets of coordinates on the same set of axes
scatter3(X,Y,Z,S)	Specifies the circle sizes
scatter3(X,Y,Z,S,C)	Specifies the circle colors
scatter3(___,'filled')	Fills in the circles, using any of the input argument combinations in the previous syntaxes.
scatter3(___,markertype)	Specifies the marker type.

Figure 3: Scatter3() Functions for MATLAB's graphics

4. Results and discussion

4.1. 2D Visualization by MATLAB

4.1.1. Symmetries With Respect to y-Axis

By using MATLAB, the equation $y = \sqrt{x^2 - 1}$ is plotted by using function $plot(x, y)$. The x-axis range was set to [0,1].

MATLAB coding:

```
x = [0:0.01:1];

y = sqrt(1-x.^2);

hold on

plot(x,y)

grid on
```

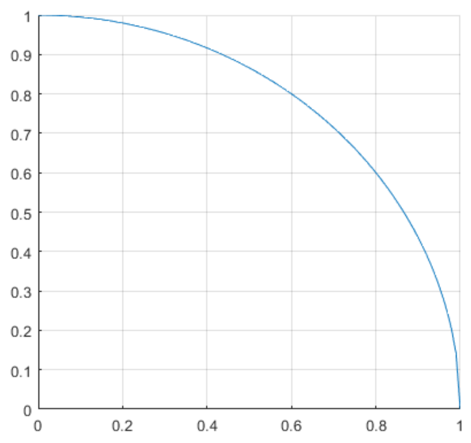


Figure 4: Plot of equation y=√(x^2-1)

The equation is only symmetric with respect to y-axis. Therefore, the 2D visualization of the whole shape was interpreted by applying the symmetries concept which is by changing x sign to $-x$ sign.

MATLAB coding:

```
x = [0:0.01:1];

y = sqrt(1-x.ˆ2);

hold on

plot(x,y)

grid on

plot(-x,y)
```

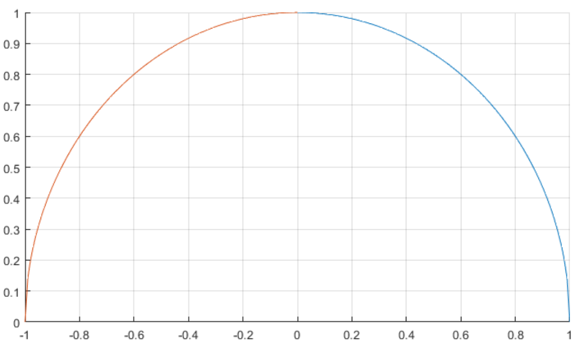


Figure 5: Plot of equation $y = \sqrt{x^2 - 1}$ with symmetry plots w.r.t y-axis.

4.1.2. Symmetries With Respect to x-Axis

Another equation that is symmetric along the x-axis will be used which is $y^2 = -x$. $y^2 = -x$ is an implicit equation that cannot be plotted by using MATLAB $plot(x, y)$ function directly. Instead, MATLAB anonymous function whose data type is function handle was used. By letting variable f as a function handle, the equation was plotted by using fimplicit function over the interval [-10 0] for x and [0 5] for y. The interval for y axis is limited to [0 5] since the other half will be constructed by using the symmetries with respect to x-axis. 2D visualization of the whole shape was obtained by using symmetries which is by changing y value to -y.

For this implicit equation, MATLAB does not allow the sign of y to simply be changed in the anonymous function such as f = $@(x, y)\ y: 2 + x$. Therefore, the alternative that was used to change construct the 2D visualization is by extracting the x and y values from the plot above and reconstruct the plot by using $plot(x, y)$ function with y value changed to $-y$. By using MATLAB, the $fimplicit()$ plot was assigned to variable 'h'. Then, the x and y data from the fimplicit plot was extracted and assigned to variable 'X' and 'Y' respectively. The symmetries can now be obtained by using $plot(x, y)$ function with y interval set to [-5 5].

MATLAB coding:

```
f= @(x,y) y.ˆ2 + x;

figure

h=fimplicit(f,[-10 0 0 5],'b');

X = get(h, 'XData');

16

Y = get(h, 'YData');

hold on

plot(X,-Y)

ylim([-5 5])

xline(0)

yline(0)

xlabel x

ylabel y

grid on
```

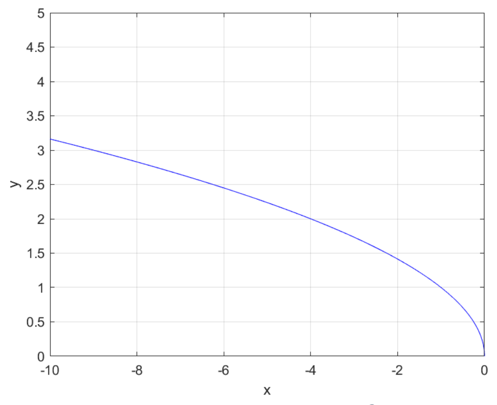


Figure 6 Plot of equation $y^2 = -x$

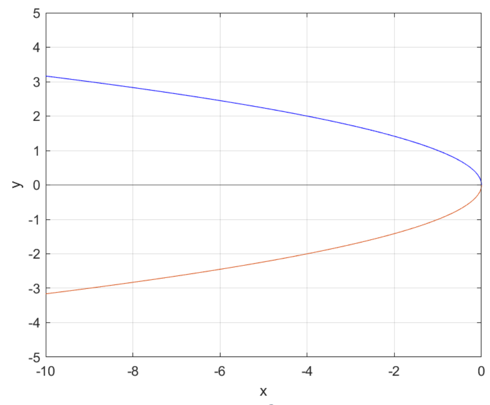


Figure 7: Plot of equation $y^2 = -x$ with symmetry plots w.r.t y-axis.

4.2. Plotting According to Coordinate

4.2.1 Symmetries With Respect to y-Axis

The graph is then plotted with five points within x= 0 to 1. The points was obtained randomly using function $randperm()$ in MATLAB programming. Then, by replacing x value with $-x$ value, the symmetry of the points are obtained with respect to y axis.

MATLAB coding:

```
x=[0:0.01:1];

y=sqrt(1-x.ˆ2);

hold on

plot(x,y)

grid on

s1 = 5

randomIndexes = randperm(length(y), s1)

xRandom = x(randomIndexes);

yRandom = y(randomIndexes);

plot(xRandom, yRandom, ’ro’, ’LineWidth’, 2,

’MarkerSize’, 2);

hold on

plot(-xRandom, yRandom, ’ro’, ’LineWidth’, 2, plot( ’MarkerSize’, 2);

grid on
```

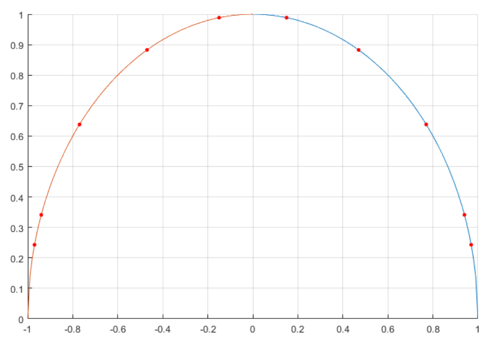


Figure 7: Random plot along $y = \sqrt{x^2 - 1}$

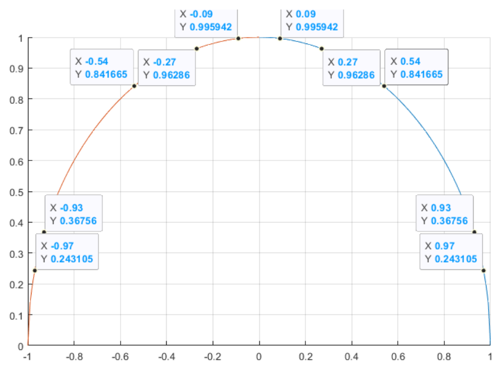


Figure 8: Random Plots along $y = \sqrt{x^2 - 1}$ with details

4.2.2 Symmetries With Respect to y-Axis

By applying the same steps as in previous section, the graph for $y^2 = -x$ is plotted with five points within y= -5 to 5. The points was obtained randomly using function \emph{randperm()} in MATLAB programming. Then, by replacing y value with $-y$ value, the symmetry of the coordinates are obtained with respect to x axis. The MATLAB coding for this step is as below.

```
\verb|f = @(x,y) y.^2 + x;|

\verb|figure|

\verb|hold on|

\verb|X = get(h, 'XData');|

\verb|Y = get(h, 'YData');|

\verb|plot(X, -Y)|

\verb|hold on|

\verb|s1=5|

\verb|randomIndexes = randperm(length(Y), s1);|

\verb|xRandom = X(randomIndexes);|

\verb|yRandom = Y(randomIndexes);|

\verb|plot(xRandom, yRandom, 'ro', 'LineWidth', 2,|

\verb|'MarketSize', 2);|

\verb|hold on|

\verb|plot(xRandom, -yRandom, 'ro', LineWidth', 2,|

\verb|'MarketSize', 2);|

\verb|ylim([-5 5])|

\verb|grid on|

\verb|xline(0)|
```

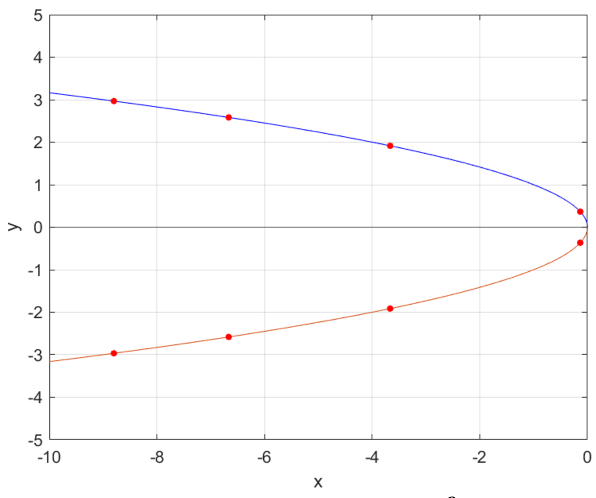


Figure 9: Random plots along $y^2 = -x$

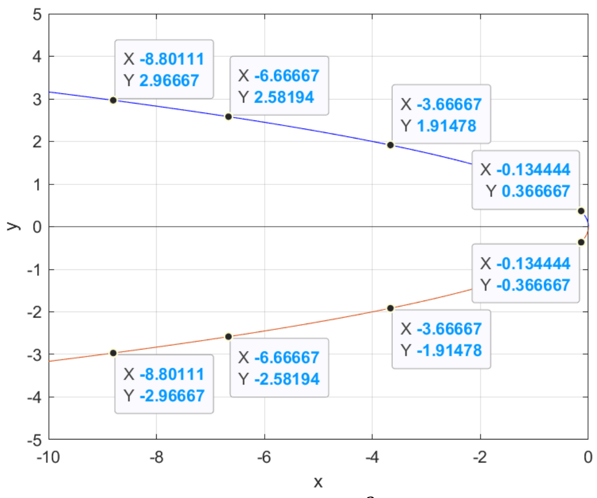


Figure 10: Random plots along $y^2 = -x$ with details

4.3. Symmetries Plots in Graph

In this study, the sphere with radius equal to 3 was identified whether it will has plane of symmetry with respect to xy-plane, xz-plane, and yz-plane. An axis of symmetry can be defined as a line in space around which an object can be turned 360 degrees and then repeated. It will, in other terms, fit inside itself. By using MATLAB, the sphere was proved to has symmetries along xy-plane, xz-plane, and yz-plane since it fit into itself in each plane. The sphere in figures below was plotted by using MATLAB $sphere()$ function.

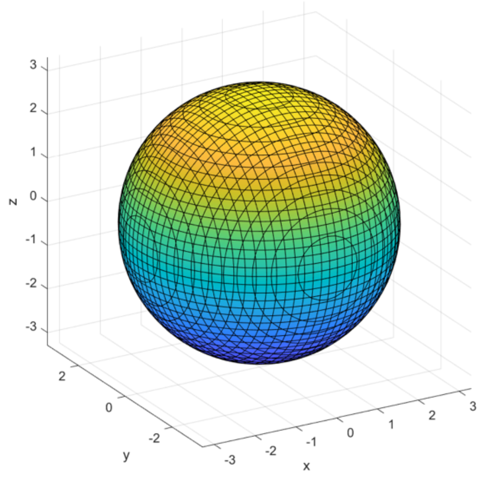


Figure 11: Sphere in xyz-plane

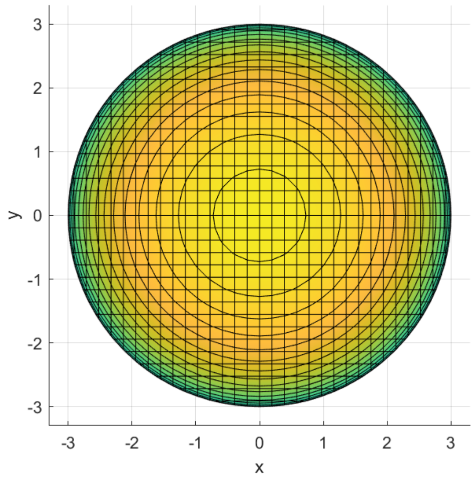


Figure 12: Sphere in xy-plane

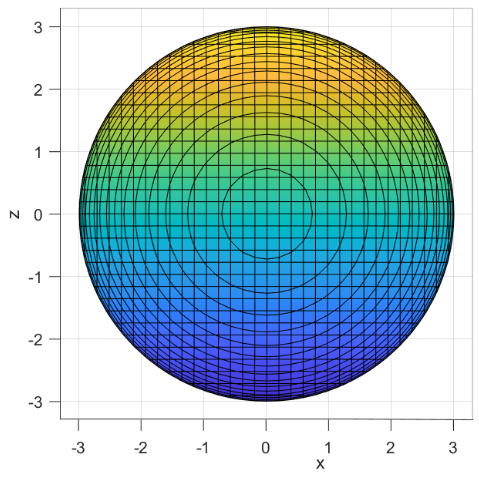


Figure 13: Sphere in xz-plane

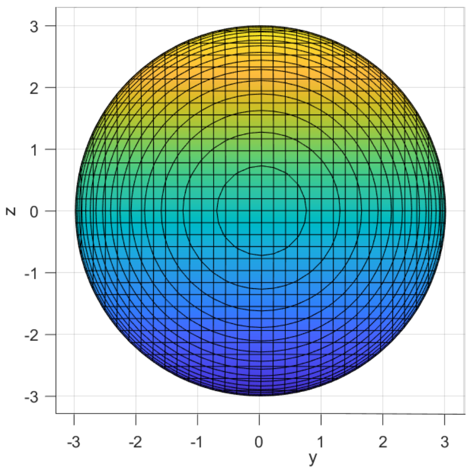


Figure 14: Sphere in yz-plane

4.3.1 Visualization of 3D Object Using Symmetries

This section explained the 3D visualization of the sphere which was constructed by using the symmetries. A set of data that consists of hundreds random coordinates within the first octant was used to study the importance of symmetries in reconstructing 3D objects. The coordinates were plotted in 3D plots by using scatter plot. When variables x, y, and z were assigned, the symmetries in octant II can be obtained changing x values to negative sign. Then, to obtain symmetries in octant III and IV, we combine both data set in octane I and II and assigned it to a new variable. Finally, octant V, VI, VII, and VIII can be obtained by applying the same concept.

Step 1: Import a set of data in matrix form into MATLAB and specified x, y, and z.

```
>> x=data(:,1)
>> y=data(:,2)
>> z=data(:,3)
```

Step 2: Plot the data set using scatter3 function in MATLAB.

Variable x, y, and z are now assigned to its respective values so we can plot the coordinates by using function scatter3.

```
>>scatter3(x,y,z)
```

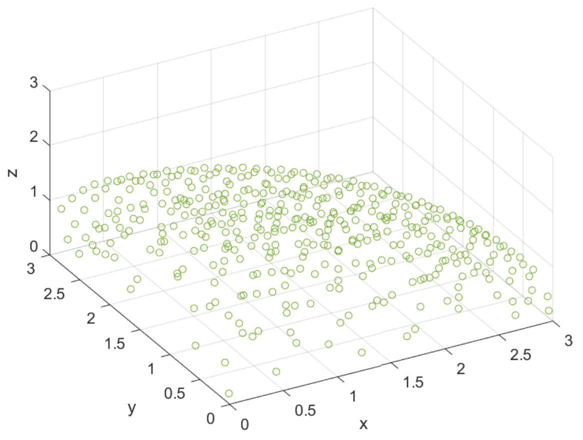


Figure 15: Plots in octant I (xyz-plane)

Step 3: Obtain the symmetry of the plots with respect to y-axis.

In step 2, the data set already has the assigned variable to each x, y, and z value which are 'x', 'y', and 'z'. Now, in order to obtain the plots' symmetry with respect to y-axis, a negative sign was added to variable 'y'.

```
scatter3(x,y,z)

hold on

scatter3(-x,y,z)
```

From the figures below, we can see that by changing x variable to negative sign, we only obtained the symmetry plots in octant II. This is because the coordinates are only within octant I and all values for variable y and z are still positive.

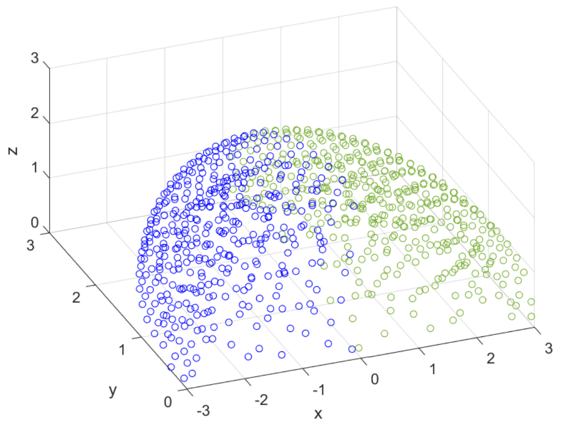


Figure 16: Symmetry plots with respect to y-axis (xyz-plane)

Step 4: Add new variables with updated data set to obtain symmetries of octant I and octant II.

The coordinates of the plots in octant II need to be stored in new variable ’b’ meanwhile the coordinates from both octants were combined and stored into a new variable named ’ab’. By using command window,

```
>> b = (-x,y,z)
>> ab = [a;b]
```

Now, assume that the combined plots’ coordinates of octant I and II are denoted by ‘x1’,’y1’, and ‘z1’. Then, by using command window, the values were assigned accordingly.

```
>> x1=ab(:,1)
>> y1=ab(:,2)
>> z1=ab(:,3)
```

Note that octant III and IV are the mirror of octant I and II with respect to x. By adding negative sign to ‘y1’, the symmetry plots of coordinates in octant I and II with respect to x can be obtained.

```
scatter3(x,y,z)

hold on

scatter3(-x,y,z)

hold on

scatter3(x1,-y1,z1)
```

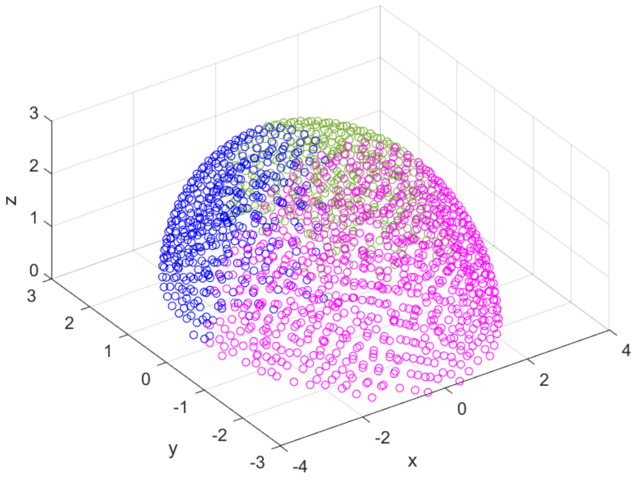


Figure 17: Symmetry plots with respect to x-axis (xyz-plane)

Step 5: Assign the data set of plots in octant IV in new variable ’c’.

```
>> c=[x1,-y1,z1]
```

Then, all the data sets were assigned to a new variable ’abc’. By applying the same concept in previous steps, the symmetry plots of octant I, II, III and IV will interpret the whole 3D visualization of the plots.

```
>> abc = [ab;c]
```

In data set 'abc', x values are assigned as x2, y values as y2, and z values as z2.

```
>> x2=abc(:,1)
>> y2=abc(:,2)
>> z2=abc(:,3)
```

Data set in variable 'abc' consists of the coordinates of the plots in the first four octants. All z values in the data set are now positive values. By adding negative sign to 'z2', we obtained the whole eight octants of the symmetry plots.

```
scatter3(x2,y2,-z2)
```

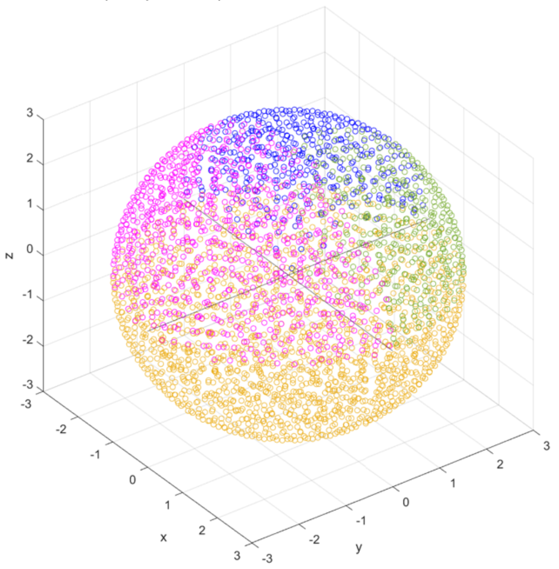


Figure 18: Symmetry plots with respect to y-axis (xyz-plane)

MATLAB can be used in interpreting 3D visualization of an object by finding symmetries of the plots. Symmetries are proven to show that objects have shape and structure.

4.3.2 Visualization of 3D Object With Surface

Consider the $sin(r)/\,r$ or $sin\ c$ function as an example of how meshgrid can be used. You only need to supply one vector argument to meshgrid to evaluate this function between -8 and 8 in both x and y, and it will be used in both directions.

```
[X,Y] = meshgrid(-8:.5:8);

R = sqrt(X.^2 + Y.^2) + eps;
```

The distance from the origin, which is the centre of the matrix, is contained in the matrix R. Adding eps prevents the data from being divided by zero (in the next step), which results in Inf values. The 3-D surface is created by forming the sinc function and plotting Z with $mesh()$.

```
Z = sin(R)./R;

figure

mesh(X,Y,Z)
```

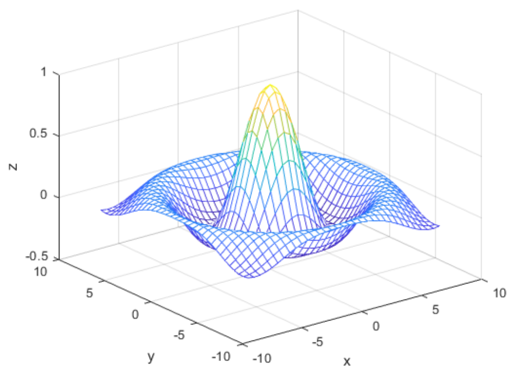


Figure 19: 3D Surface Plotting using Mesh Function

The 3D surfaces constructed in MATLAB allow the structures that are difficult to represent mathematically to be graphically shown and understood by studying the surfaces. Visualizing minimal surfaces is important because it allows users to view and handle physical objects representing various natural and mathematical phenomena [13].

Conclusion

The primary objective of this research was to visualize three-dimensional along with two dimensional and the importance of symmetries throughout the research. This research begins by understanding the tools provided by MATLAB software that can be used in visualization. Then, the visualization of 2D shapes and 3D object by MATLAB were explored in depth by exploring related tools and functions provided by MATLAB software such as $plot(x,y)$, $plot3(x,y,z)$, $scatter3(x,y,z)$, $fimplicit()$, $fimplicit3()$, $mesh()$ and $surf()$ functions. In order to construct the 2D and 3D visualization, the uses of symmetries were highlighted. This research explain how symmetries were used in the visualization process for both shape and plot given.

Acknowledgement

I would like to express my sincere aprreciation to my supervisor, Dr. Taufiq Khairi bin Ahmad Khairuddin, for his invalueable guidance, encouragement, and critics throughout this research. It was a great privilege and honour to work and study under his guidance. I am extremely grateful to my parents for their love, prayers, caring and continuous support to complete this research paper. Throughout this process, their trust in me has maintained my spirits and motivation strong.

References

[1] Birngruber, E., Donner, R., Langs, G., et al. (2009). matvtk3d visualization for matlab. In *Proceedings of the MICCAI 2009 Workshop on systems and architectures for CAI*, pages 1–8.

[2] Christara, C. and Ai, W. (2001). *ABRIEF INTRODUCTION TOMATLAB*.

[3] Dykes, J., MacEachren, A., and Kraak, M. (2005). Exploring geovisualization. *Exploring Geovisualization*, page 3.

[4] Higham, D. J. and Higham, N. J. (2016). *MATLAB* guide. SIAM.

[5] Knight, A. (2019). *Basics of MatLab R and beyond*. Chapman and Hall/CRC.

[6] Lee, Y. L. and Saunders, J. A. (2011). Symmetry facilitates 3d shape discrimination across changes in viewpoint. *i-Perception*, 2(4):403–403.

[7] Locher, P. and Nodine, C. (1989). The perceptual value of symmetry. In *Symmetry 2*, pages 475–484. Elsevier.

[8] Martinet, A., Soler, C., Holzschuch, N., and Sillion, F. X. Accurate detection of symmetries in 3d shapes. ACM Transactions on Graphics (TOG).

[9] Mathworks (2014). MATLAB 3-D Visualization R2014b.

[10] M¨uller, R. and Zeckzer, D. (2015). Past, present, and future of 3d software visualization-a systematic literature analysis. *In International Conference on Information Visualization Theory and Applications*, volume 2, pages 63–74. SCITEPRESS.

[11] Muna, N. and Patterson, A. E. (2018). Simple 3-d visualization of some common mathematical minimal surfaces using matlab. Technical report.

[12] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. (2006). A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2006 Papers*.

[12] Szabo, V. (2018). Knowledge in 3d: How 3d data visualization is reshaping our world. *Parameters*, 10. 43

[13] Teyseyre, A. R. and Campo, M. R. (2008). An overview of 3d software visualization. *IEEE transactions on visualization and computer graphics,* 15(1):87–105.

[14] Tyler, C. W. (2003). Human symmetry perception and its computational analysis. Psychology Press.

[15] Wang, H. (2016). Research on three dimensional visualization technologies. In Proceedings of the 2016 4th international conference on advanced materials and information technology processing, pages 261–265

[16] Weyl, H. (2015). Symmetry. In *Symmetry*. Princeton University Press.

[17] Wood, J., Kirschenbauer, S., D¨ollner, J., Lopes, A., and Bodum, L. (2005). Using 3d in visualization. *In Exploring geovisualization*, pages 293–312. Elsevier.