CZ2003: Computer Graphics and Visualization

Lab Report 5:

Transformations and Motions

Submitted by: Ng Chi Hui

Matriculation Number: U1922243C

Group Number: 1

School of Computer Science and Engineering

1. a. Derive a transformation matrix performing rotation by $\frac{\pi}{2}$ about an axis parallel to axis Y and passing through the point with coordinates (**M**+5, 0, 0).

$$\begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} z+8 \\ y \\ -x+8 \\ 1 \end{bmatrix}$$

The transformation matrix is consists of first a translation from (8, 0, 0) to the origin first, followed by rotation about y axis by $\frac{\pi}{2}$ and then translation of the rotated figure back to the point (8, 0, 0). Note that M used her is 3 and M + 5 = 8.

b. Apply this matrix to the parametric definitions of the curve obtained in experiment 1 (exercise 3). Derive the transformed definitions $x(u)$, $y(u)$, $z(u)$, $u \in [0,1]$ of the rotated curve and display it

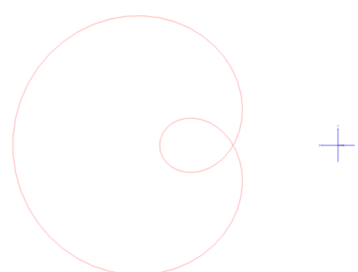


Fig 1.0 Snapshot of "Q1B1.wrl" which defines a limacon *parameters [0 1] and resolution of [75]*

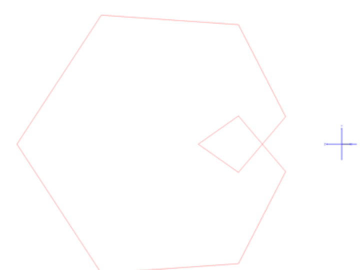


Fig 1.1 Snapshot of "Q1B2.wrl" which defines a limacon *parameters [0 1] and resolution of [10]*

Parametric Function Definition:

```
definition "
function parametric_x(u,v,w,t)
      { return 8;}

 function parametric_y(u,v,w,t)
      {return (sin(2*pi*u))*(4-8*cos(2*pi*u));}

function parametric_z(u,v,w,t)
      {return - (cos(2*pi*u))*(4-8*cos(2*pi*u)) +8; }"

 }
```

Remarks

The surface is created by joining multiple straight lines together between the points defined in the formula to form a curved-liked shape. For a curve to be sufficiently approximated to a straight line, there must be sufficiently larger number of sampling points. The higher the number of samples use, the more accurate and smoother the object will be. The minimum sampling resolution for displaying the transformed limacon is [75].

2. Modify the parametric definitions to $x(u,t)$, $y(u,t)$, $z(u,t)$, $u, t \in [0,1]$ so that the rotation of the curve will be displayed as a 5 seconds rotation motion with some deceleration.

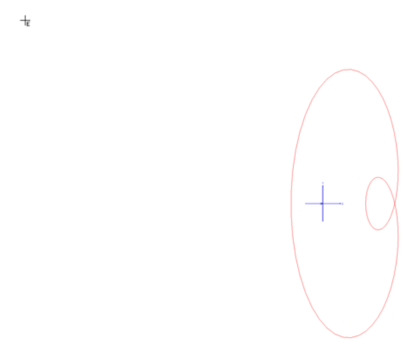


Fig 2.0 Snapshot of "2.wrl" at different time period

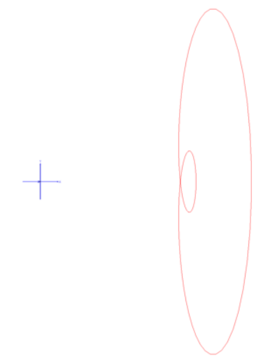


Fig 2.1 Snapshot of "2.wrl" at different time period

<u>Final Parametric Function Definition:</u>

```
definition "
 function parametric_x(u,v,w,t)
{x1 = cos(2*pi*u)*(4 - 8*cos(2*pi*u));   x2=8;  return x1+(x2-x1)*sin(pi/2*t);}

function parametric_y(u,v,w,t)
{y1 = sin(2*pi*u)*(4 - 8*cos(2*pi*u));   y2=sin(2*pi*u)*(4 - 8*cos(2*pi*u));
return y1+(y2-y1)*sin(pi/2*t);}

function parametric_z(u,v,w,t)
{z1 = 0; z2 = -cos(2*pi*u)*(4 - 8*cos(2*pi*u)) + 8 ; return  z1+(z2-z1)*sin(pi/2*t);}"
```

<u>Remarks</u>

A minimum sampling of [75] will be sufficient. To achieve deceleration use sin(pi/2*t) and set cycle Interval to 5 so that the rotation motion will take 5 seconds.

3. a. With reference to Table 3, convert to (u,v), $y(u,v)$, $z(u,v)$, $u, v \in [0,1]$ definitions of surfaces **M** and (**N+M**) and display them.

<u>Surface 3</u>

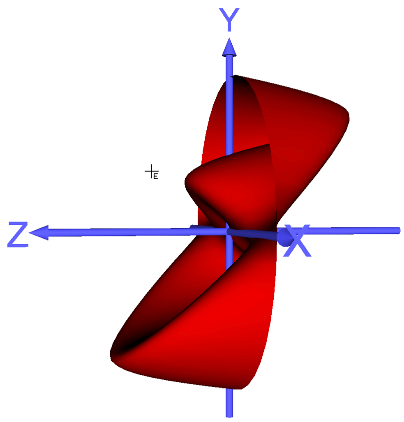


Fig 3.0 Snapshot of "3A1.wrl" has parameters [0 1 0 1] and resolution of [40 40] when graphic mode is set to "wireframe"

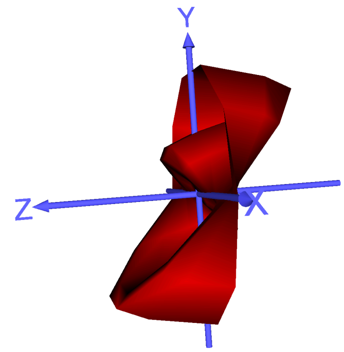


Fig 3.1 Snapshot of "3A12.wrl" has parameters [0 1 0 1] and resolution of [10 10]

<u>*Parametric Function Definition:*</u>

definition "
function parametric_x(u,v,w,t)
{x1 = u*cos(2*pi*v); return x1;}

function parametric_y(u,v,w,t)
{y1 = u*sin(2*pi*v);   return y1;}

function parametric_z(u,v,w,t)
{z1 = u*sin(2*u*pi)*sin(2*pi*v); return z1;}"

$u, v \in [0,1]$

<u>*Remarks*</u>

A minimum resolution of 40 will be sufficient to display the surface. The surface is created by joining multiple straight lines together between the points defined in the formula to form a curved-liked shape. For a curve to be sufficiently approximated to a straight line, there must be sufficiently larger number of sampling points. The higher the number of samples use, the more accurate and smoother the surface will be.

<u>Surface 7</u>

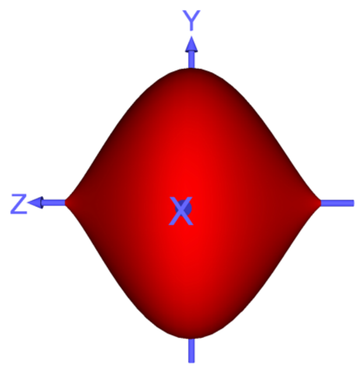
<table>
<tr>
<td>

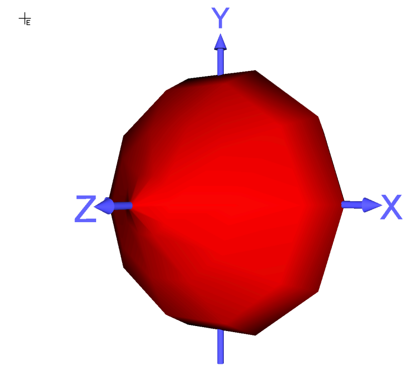
Fig 3.2 Snapshot of “3A2.wrl” has parameters [0 1 0 1] and resolution of [40 40]</td>
<td>

Fig 3.3 Snapshot of “3A22.wrl” has parameters [0 1 0 1] and resolution of [10 10]</td>
</tr>
<tr>
<td colspan="2"><u>Parametric Function Definition:</u>

definition "
function parametric_x(u,v,w,t)
{x2 = cos(2*pi*u)*(sin(pi*v))^3; return x2;}

function parametric_y(u,v,w,t)
{y2 = sin(2*pi*u)*(sin(pi*v))^3; return y2;}

function parametric_z(u,v,w,t)
{z2 = cos(pi*v); return z2;}"

$u, v \in [0,1]$

<u>Remarks</u>
A minimum resolution of 40 will be sufficient to display the surface. The surface is created by joining multiple straight lines together between the points defined in the formula to form a curved-liked shape. For a curve to be sufficiently approximated to a straight line, there must be sufficiently larger number of sampling points. The higher the number of samples use, the more accurate and smoother the surface will be.</td>
</tr>
</table>

b. Define parametrically using (u, v, t), $y(u, v, t)$, $z(u, v, t)$, $u, v, t \in [0,1]$ a swing (back and forth) morphing transformation between surfaces **M** and (**N+M**). The morphing animation has to take 5 seconds and has be done with a uniform speed.

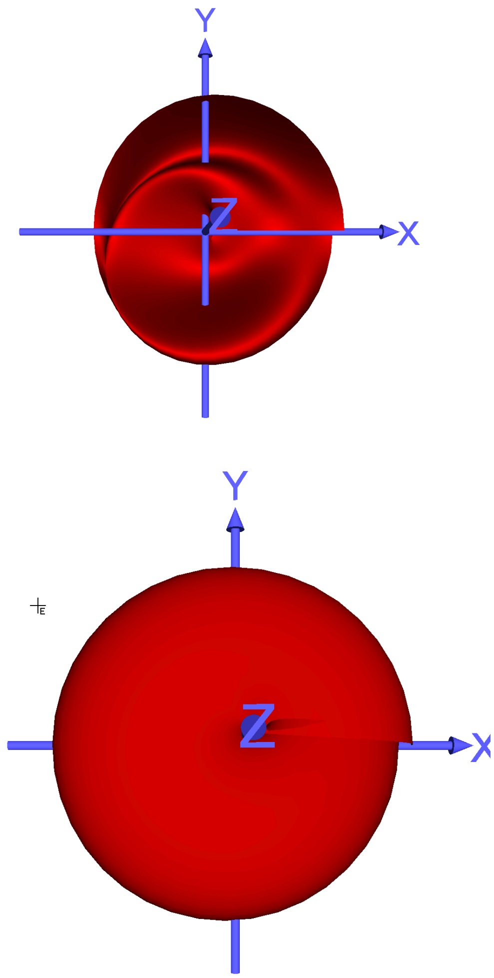


Fig 3.4 Snapshots of "3B1.wrl" defines the morphing transformation with parameters [0 1 0 1] and resolution [40 40] at different instances in time

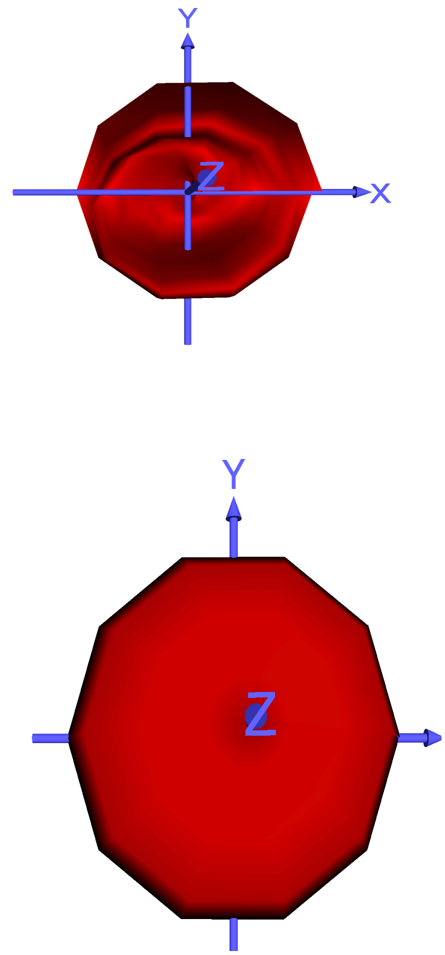


Fig 3.5 Snapshot of "3B2.wrl" defines the morphing transformation with parameters [0 1 0 1] and resolution [10 10] at different instances in time

Fig 3 are snapshots of the morphing process

<u>*Parametric Function Definition:*</u>

```
definition "
function parametric_x(u,v,w,t)
{x1=u*cos(2*pi*v);   x2 = cos(2*pi*u)*(sin(pi*v))^3;  return x1*(1-(1-fabs(1-2*t))) + x2*(1-fabs(1-2*t));}

function parametric_y(u,v,w,t)
{y1 = u*sin(2*pi*v);   y2 = sin(2*pi*u)*(sin(pi*v))^3;   return y1*(1-(1-fabs(1-2*t))) + y2*(1-fabs(1-2*t));}

function parametric_z(u,v,w,t)
{z1 = u*sin(2*u*pi)*sin(2*pi*v); z2 = cos(pi*v); return z1*(1-(1-fabs(1-2*t))) + z2*(1-fabs(1-2*t)); }"
```

<u>*Remarks*</u>

To achieve morphing transformation, we can use the linear interpolation model where V(T) = (1-T)(A) + T(B), where T ∈ [0,1]. To achieve swing of back and forth T = 1 - fabs(1-2*t) where the fabs

function can be used instead of abs. Both will give us absolute values. The cycle is set to 5 so that the rotation motion will take 5 seconds. The surfaces are created by joining multiple straight lines together between the points defined in the formula to form a curved-liked shape. For a curve to be sufficiently approximated to a straight line, there must be sufficiently larger number of sampling points. The higher the number of samples use, the more accurate and smoother the surface will be. Thus, the minimum sampling resolution is 40

4. Besides the above compulsory part, you are welcome to add any other shapes of parametric solids into folder Lab5/Extras. You are also welcome to add point lights and/or directional lights to achieve some special effects. These extra exercises may increase your total mark.

Ripple Liked Effect

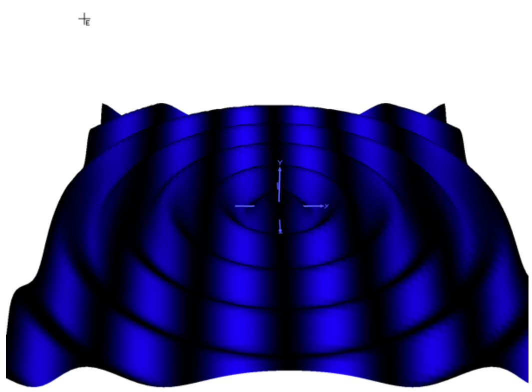


Figure 4.0 Ripple liked motion
	Open the file to view the ripple liked motion
	
	<u>Parametric Equation Definition:</u>
	
	x=3 *(4*u-2);
	
	y=0.30*sin(10*sqrt(((8*u - 4)^2) + ((8*v - 4)^2))*(1-t));
	
	z=3*(4*v-2);