Programming Computer Graphics and Visualization

Module 2
Lecture 2

We Have Learnt:

- Polygon-based visualization (e.g., by OpenGL) is fast but it compromises on precision of presentation. Geometry of fine details is often replaced by image textures (patterns) displayed on the surfaces
- Python graphics is slow unless uses C libraries like OpenGL
- Ray-tracing (e.g., by POV-Ray) is very precise but slow. It is mostly used for making images and not designed to be an interactive visualization tool.
- VRML/X3D is polygon-based but declarative style virtual scene description language. Compared to OpenGL and other graphics libraries, it requires very little time to start programming for both web-enabled and local visualization problems. It is full of technological solutions but may require a deeper learning to master them.
  However …

It is Not so Easy to Make a Watermelon in VRML

What is actually needed?

- Ability to make a solid model of a watermelon
- Ability to cut away any piece of it
- Ability to see the respective 3D color anywhere we cut the watermelon

?

Not available in VRML

- 3D-party interactive modeling tool to make a smooth polygonal mesh (e.g. 3D Studio MAX)
- 2D image texture mapping

Neither in OpenGL and other polygon-based graphics libraries.
Though Boolean operations are available in POVRay, it is slow

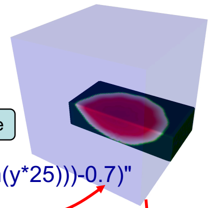
Watermelon Made by Functions

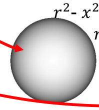
```
FShape {
  appearance FAppearance {
    material FMaterial {
  diffuseColor
  "sqrt(x^2+y^2+z^2)
  +0.01*(sin(12*atan2(x,z+0.04*sin(y*25)))-0.7)"
    patternKey
  [0   0.8  0.96   0.98   1  ]
    patternColor
  [0.4 0 0   1 0 0   1  1  1   0  1  0   0  0.2  0]
  }  }
   geometry FGeometry   {
     definition
      "min(1-x^2-y^2-z^2,
  -min(-x+z, min(y, z))"
     }
}
```

Distance function + noise

Linear color mapping

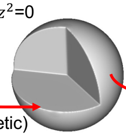
Sphere without a piece

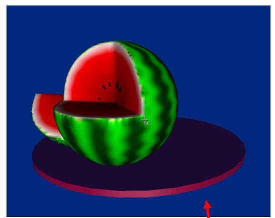


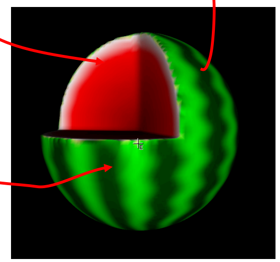


$r^2 - x^2 - y^2 - z^2 = 0$
$r = 1$

Boolean (Set-Theoretic) operations

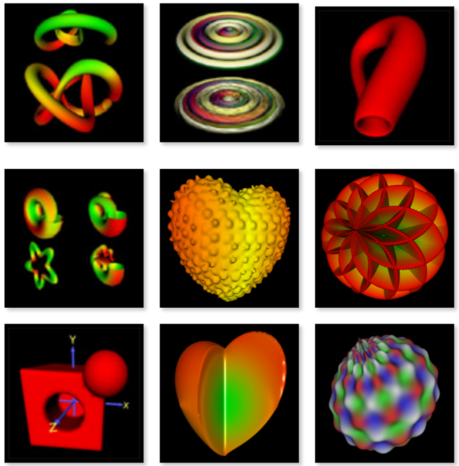






Function-based Extension of VRML and X3D

Geometry and colors are interpreted as entities defined by mathematical functions in their own coordinate domains which are then merged together into one virtual object



FVRML

- Geometry and appearance can be defined by mathematical functions of Cartesian x,y,z and parametric u,v,w coordinates as well as time t.
- Implicit, explicit and parametric analytical functions can be used interchangeably
- Implicit functions:
  - $f(x, y, z, t) = 0$, where x, y, z are Cartesian coordinates and t is the time
  - To define 3D surface
- Explicit functions:
  - $g = f(x, y, z, t)$
    To define solid objects $g \geq 0$; r, g, b values of a 3D color $0 \leq g \leq 1$ and 3D geometric texture displacement value (positive and negative).
- Parametric functions:
  - $x = f_x(u, v, w, t)$; $y = f_y(u, v, w, t)$; $z = f_z(u, v, w, t)$. Where x, y, z are Cartesian coordinates of the points, u, v and w are parametric coordinates, and t is the time.
  - Curves, surfaces, solids, r, g, b values of 3D color, 3D textures.

FVRML

- Only float variables
- Variables x,y,z are reserved for Cartesian coordinates, while variables u,v,w are parametric coordinates. Variable t is reserved for defining the time.
- *abs(x), fabs(x), sqrt(x), exp(x), log(x), sin(x), cos(x), tan(x), acos(x), asin(x), atan(x), ceil(x), floor(x), round(x), max(x,y), min(x,y), atan2(y,x), mod(x,y), cosh(x), sinh(x), tanh(x), log10(x)*
- *for-loops*, *while-loops*, *do-while-loops*, *break*, *continue*, *if-else* *(however you will unlikely use them)*
- Recursions are allowed *(however you will unlikely use it)*
- $f(x,y,z)=0$ and $f(x,y,z)\geq 0$ have to be named '*function frep*'. Parametric functions:
  '*function parametric_x*',
  '*function parametric_y*',
  '*function parametric_z*'

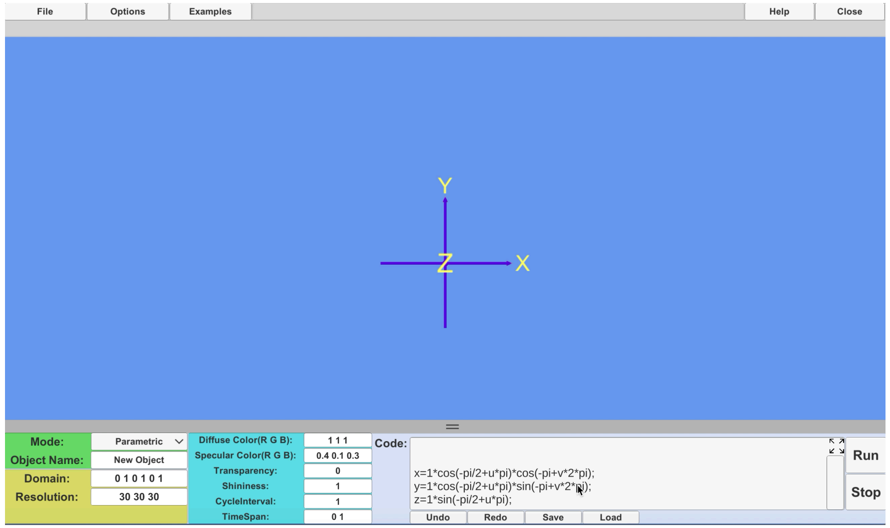
FVRML

- Install VRML viewer from Bitmanagement Software
- Install VRMLPad
- Install FVRML plugin
- Use VRMLPad or any text editor for editing VRML codes

Alternative software to use on both Windows and MacOS platforms

- Install from the course site
- Can be used for labs 1, 2, 3, 4 (partially), and 5
- Uses the same format of the content
- Work in progress (call for interested students to work on it)

New Software (work in progress)



Summary

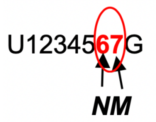
- We do not teach you graphics programming languages, we only overview them. We teach you fundamentals of visualization when mathematical functions (models) are used for defining geometry of the objects.
- The same principles of separation of geometry and visual appearance are used in other computer graphics libraries and tools, e.g., OpenGL, WebGL, Java 3D, POV-Ray, VRML, X3D, etc.
- VRML and FVRML are used as a teaching aid to illustrate theoretical principles and rather be a container for mathematical definitions
- No prior knowledge of VRML is required
- VRML/FVRML run on Windows only ☹
- Shape Explorer supports the same format for models and runs both on Windows and MacOS.

Labs Overview

Labs Overview

1. Parametric Curves (12 marks)
2. Parametric Surfaces (12 marks)
3. Parametric Solids (12 marks)
4. Implicit Surfaces and Solids (12 marks)
5. Transformations and Motions (12 marks)

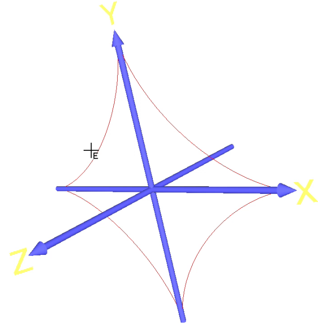
Personalized assignments



Submission of reports and files within one week (7*24 hours) after the end of each scheduled lab session (check your lab schedule)

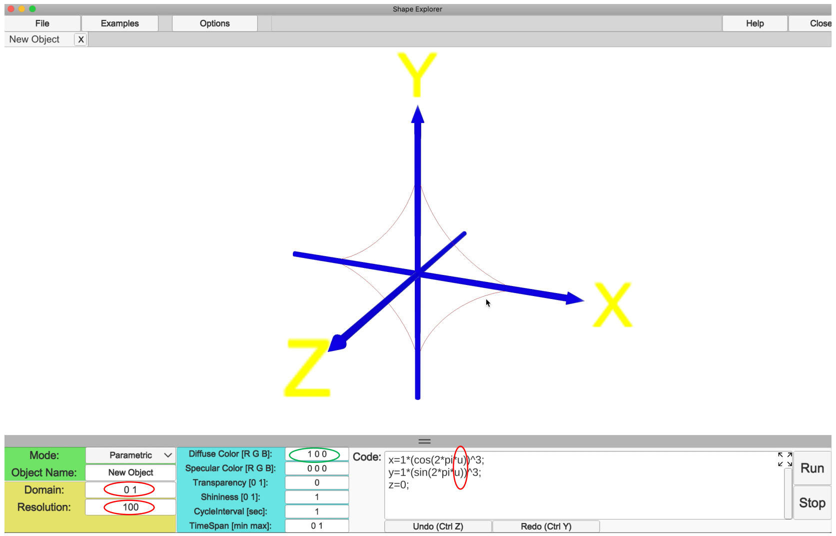
Lab1: Parametric Curves

```
FShape {
polygonizer        "analytical_curve"
geometry FGeometry {
definition "       x=1*(cos(2*pi*u))^3;
                   y=1*(sin(2*pi*u))^3;
                   z=0;"
parameters [0 1]
resolution [100]
 }
appearance FAppearance {
material FMaterial { diffuseColor
"r=1; g=0; b=0;"   }   }
}
```



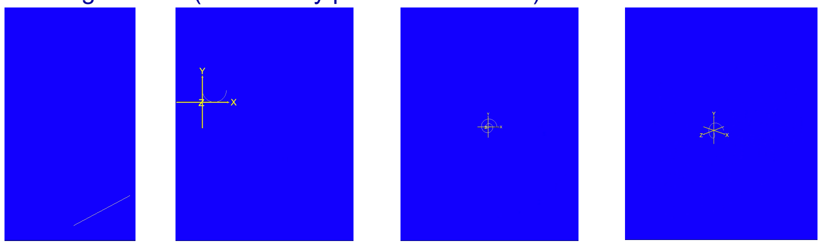
link

Lab1: Parametric Curves



Lab1: Parametric Curves

Assignment 1 (elementary parametric curves)



(a) (b) (c) (d)

1. Define parametrically in 4 separate files using functions $x(u), y(u), u \in [0,1]$ and display:

a. Straight line segment spanning from the point with coordinates (-***N***,-***M***) to the point with coordinates (***M***,***N***).

b. A circular arc with radius ***N***, centered at point with coordinates (***N***,***M***) with the angles $[\frac{\pi}{N}, 2\pi]$.

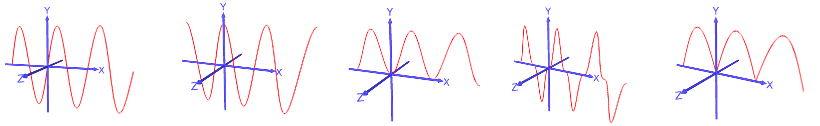
c. Origin-centered 2D spiral curve which starts at the origin, makes ***N***+***M*** revolutions clockwise and reaches eventually the radius 2****M***.

d. 3D cylindrical helix with radius ***N*** which is aligned with axis *Z*, makes ***M*** counterclockwise revolutions about axis *Z* while spanning from $z_1 = -\boldsymbol{N}$ to $z_1 = \boldsymbol{M}$.

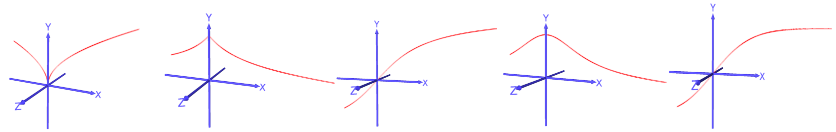
(4 marks)

Lab1: Parametric Curves

Assignment 2 (conversion from explicit to parametric representations)



(1) (2) (3) (4) (5)



(6) (7) (8) (9) (10)

Lab1: Parametric Curves

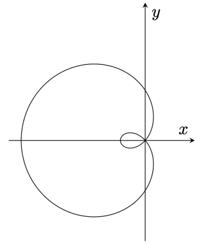
Assignment 3 (conversion from polar to Cartesian coordinates)

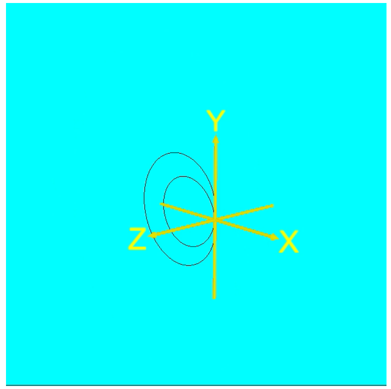
With reference to the figure, a curve is defined in polar coordinates by:

$$r = \boldsymbol{N} - (\boldsymbol{M} + 5)\cos\alpha \quad \alpha \in [0, 2\pi]$$

Define the curve parametrically as $x(u), y(u), u \in [0,1]$ and display it.

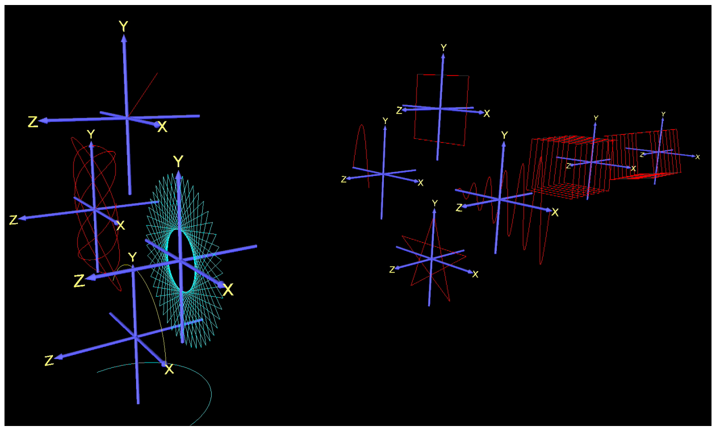
(4 marks)





Lab1: Parametric Curves

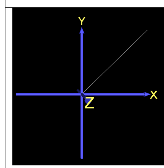
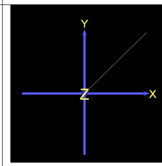
Extras (assignments 1, 2, 3: "meet expectations",
extras may "exceed expectations" and "far exceed expectations")



Lab1: Parametric Curves

Report:

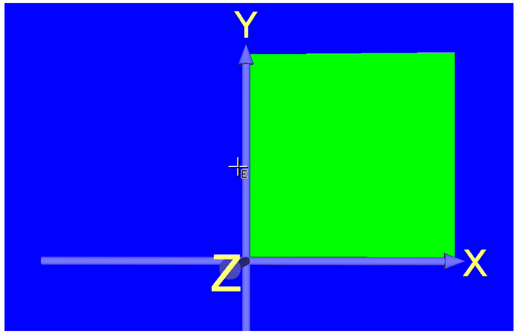
- Copy screenshots of the displayed shapes;
- Write their function definitions, domains, resolutions, other parameters;
- Write the names of the respective FVRML or Shape Explorer files;
- Write brief descriptions of your experiments with the shapes and observations made.

Curve 1	Curve 2	Notes
 Above is the snapshot of "straight line 1.wrl" which define a straight line by parametric equations x=u, y=u, and z=0 with parameter domain [0,1]. The sampling resolution is 100	 Above is the snapshot of "straight line 2.wrl" which define the same curve as the picture on the left with the same equation. The sampling resolution is 10	**Note 1:** The sampling resolution for straight line can be set as minimum as 1 and nothing will change because it basically only requires one straight line to create a straight line.
		Note 2:

Lab2: Parametric Surfaces

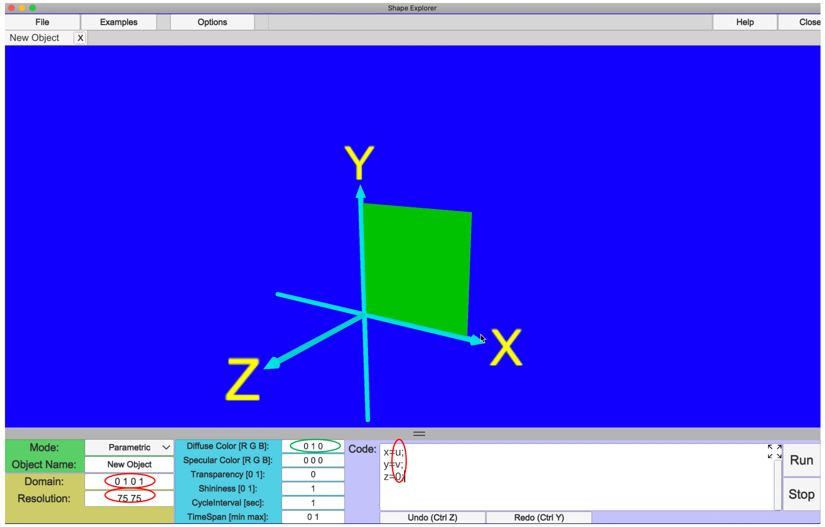
```
FShape {

geometry FGeometry {
definition "       x=u;
                   y=v;
                   z=0;"
parameters [0 1 0 1]
resolution [75 75]
 }
appearance FAppearance {
material FMaterial {
diffuseColor "r=0; g=1; b=0;"   }    }
}
```



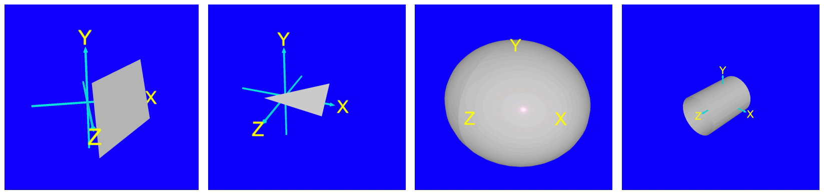
link

Lab2: Parametric Surfaces



Lab2: Parametric Surfaces

Assignment 1 (elementary parametric surfaces)



(a) (b) (c) (d)

1. In 4 separate files, define parametrically using functions $x(u,v), y(u,v), z(u,v), u,v \in [0,1]$ and display:

a. A plane passing through the points with coordinates (***N***, ***M***, 0), (0, ***M***, ***N***), (***N***, 0, ***M***).

b. A triangular polygon with the vertices at the points with coordinates (***N***, ***M***, 0), (0, ***M***, ***N***), (***N***, 0, ***M***).

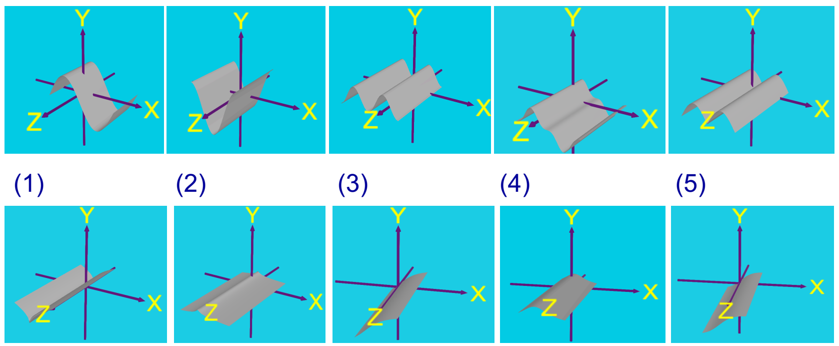
c. An origin-centered ellipsoid with the semi-axes ***N***, ***M***, (***N***+***M***)/2

d. A cylindrical surface with radius *N* which is aligned with axis Z, and spans from $z_1 = -\boldsymbol{N}$ to $z_1 = \boldsymbol{M}$.

(4 marks)

Lab2: Parametric Surfaces

Assignment 2 (making surfaces by translations sweeping of curves)

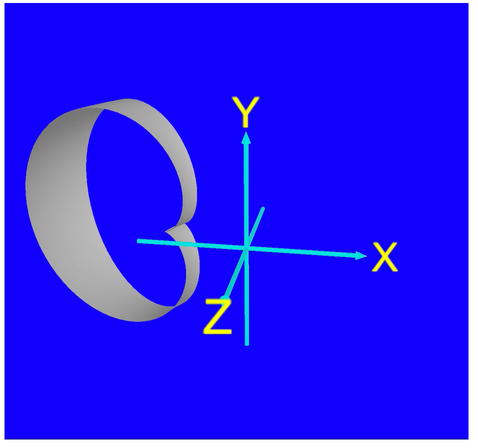


(1) (2) (3) (4) (5)

(6) (7) (8) (9) (10)

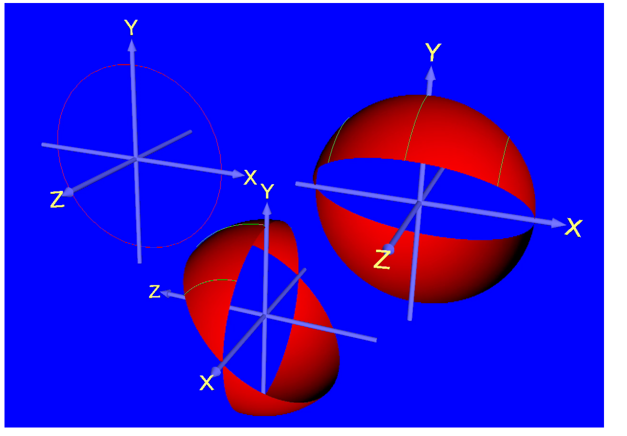
Lab2: Parametric Surfaces

Assignment 3 (making surfaces by rotational sweeping of curves)



Lab2: Parametric Surfaces

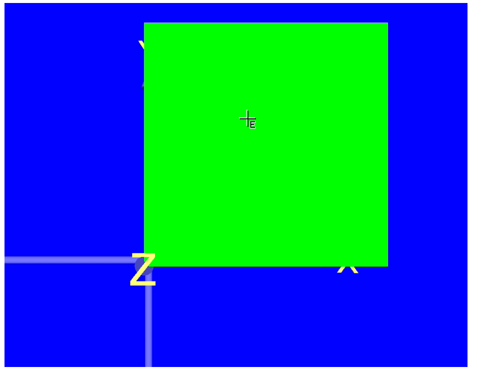
Extras



Lab 3: Parametric Solids

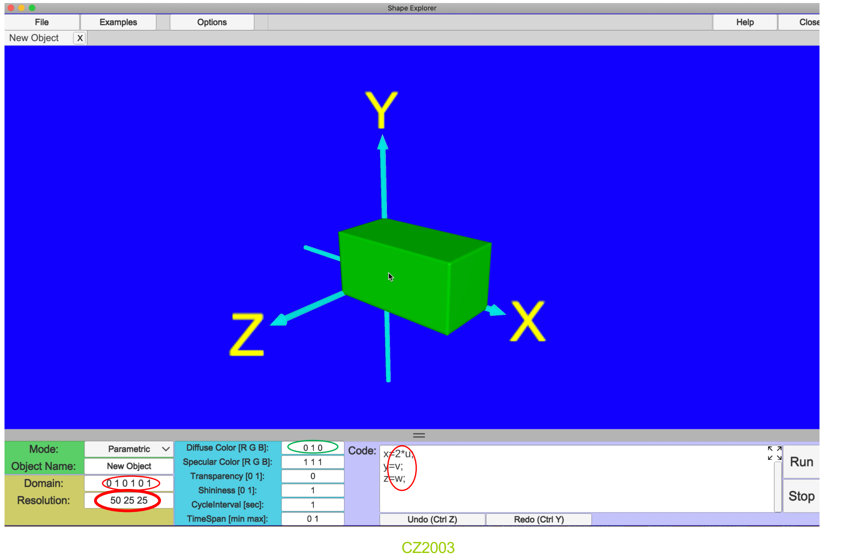
FShape {

geometry FGeometry {
definition " x=u;
y=v;
z=w;"
parameters [0 1 0 1 0 1]
resolution [75 75 75]
} **must be the same**
appearance FAppearance {
material FMaterial {
diffuseColor "r=0; g=1; b=0;" } }
}



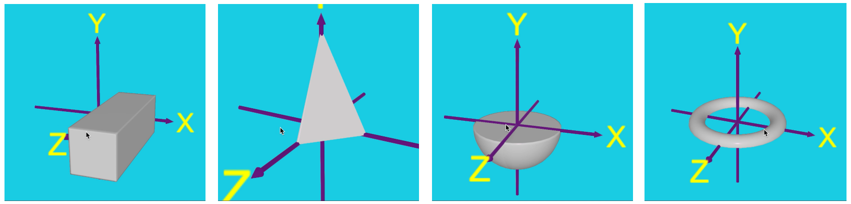
links

Lab 3: Parametric Solids



Lab 3: Parametric Solids

Experiment 1 (elementary parametric solids)



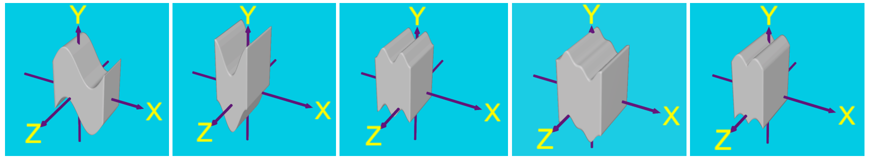
(a) (b) (c) (d)

1. Define parametrically using functions $x(u,v,w)$, $y(u,v,w)$, $z(u,v,w)$, $u,v,w \in [0,1]$ in 4 separate files and display:
   a. A solid box with the sides parallel to the coordinate planes and the coordinates of two opposite vertices (***N***, 0, ***M***), (***N+M***, ***M***, 2(***N+M***)).
   b. A solid three-sided pyramid with the vertices of the base with coordinates (0,0,0), (***N***,0,0), (0,0,***M***), and the apex at (0,***N+M***,0).
   c. A lower half of the origin-centered solid sphere with radius ***N***.
   d. An upper half of the origin-centered torus with the radius ***N*** from the center of the torus to the center of the tube and the tube radius $\frac{N}{5}$.

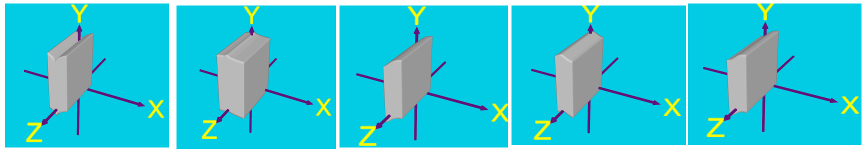
   (4 marks)

Lab 3: Parametric Solids

Experiment 2 (making parametric solids by translational sweeping)



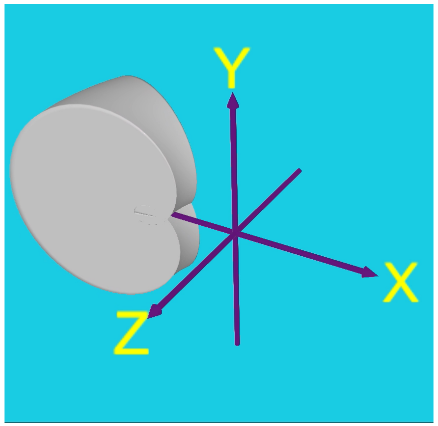
(1) (2) (3) (4) (5)



(6) (7) (8) (9) (10)

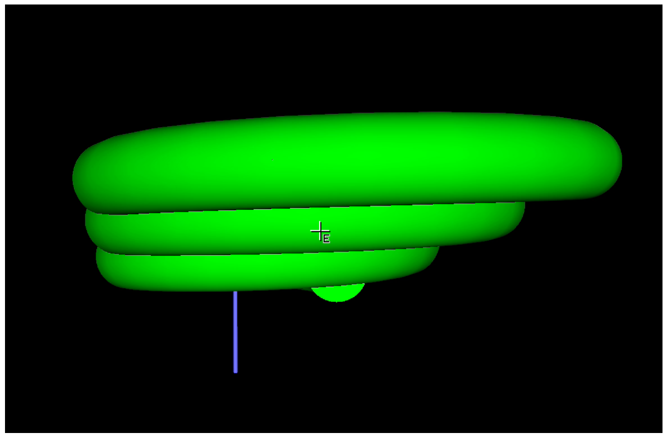
Lab 3: Parametric Solids

Experiment 3 (making parametric solids by rotational sweeping)



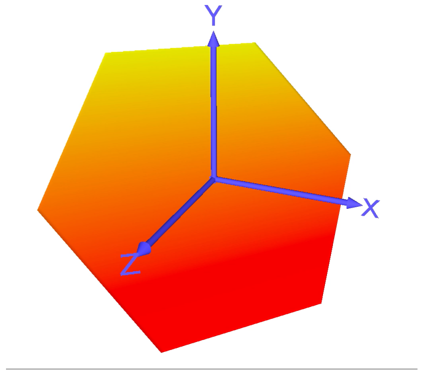
Lab 3: Parametric Solids

Extras



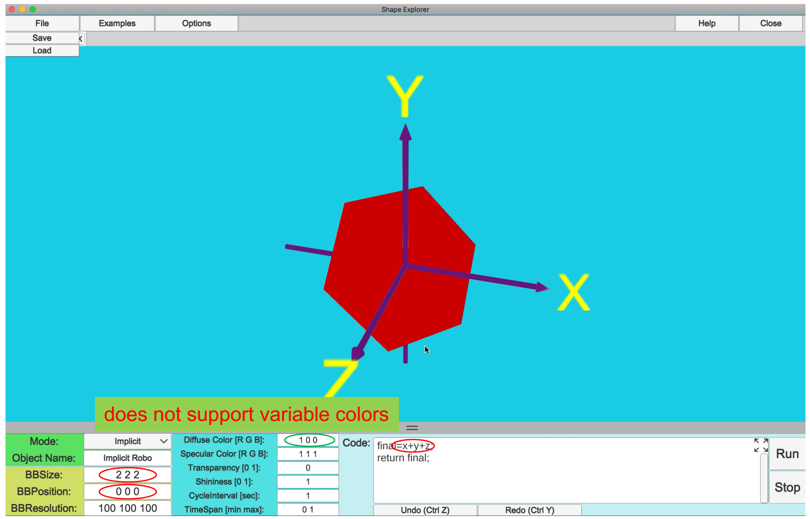
Lab 4: Implicit Surfaces and Solids

```
FShape {
geometry FGeometry {
        definition "x+y+z"
bboxCenter 0 0 0
bboxSize 2 2 2
}
appearance FAppearance {
material FMaterial {
diffuseColor "r=1; g=(v+1)/2; b=0;"
  }   }
}
```



Link

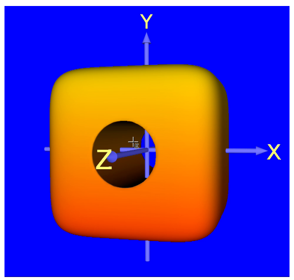
Lab 4: Implicit Surfaces and Solids



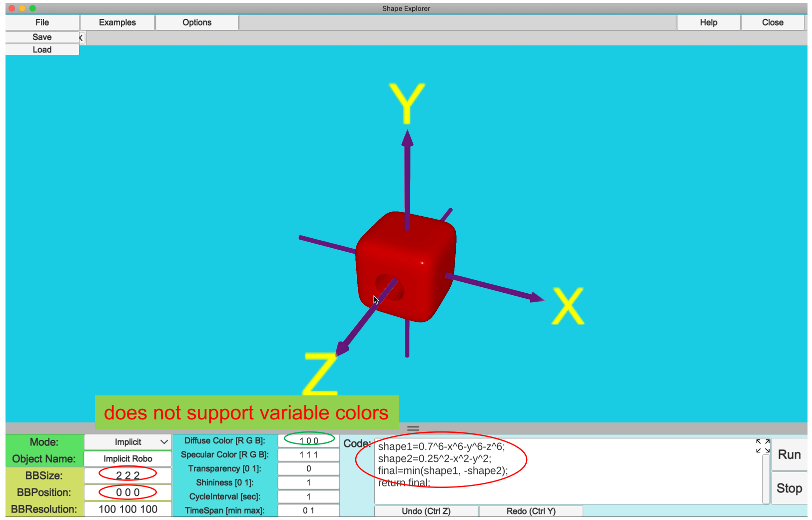
Lab 4: Implicit Surfaces and Solids

```
FShape {

geometry FGeometry {
definition "    function frep(x,y,z,t){
                shape1=0.7^6-x^6-y^6-z^6;
                shape2=0.25^2-x^2-y^2;
                final=min(shape1, -shape2);
                return final;}"
bboxCenter 0 0 0   bboxSize 2 2 2
resolution [100 100 100]
 }
appearance FAppearance {
material FMaterial { diffuseColor "r=1; g=(v+1)/2; b=0;"   }   }
}
```

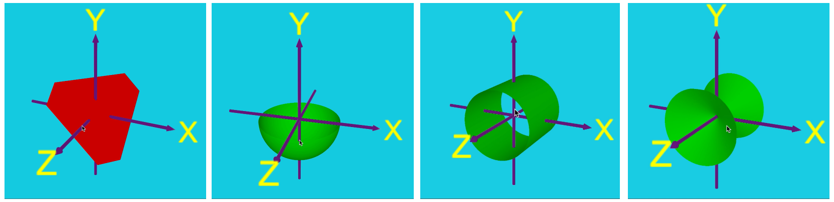


Lab 4: Implicit Surfaces and Solids



Lab 4: Implicit Surfaces and Solids

Experiment 1   (elementary implicit surfaces)



(a) (b) (c) (d)

In 4 separate files, define by implicit functions $f(x,y,z) = 0$ and by setting a proper bounding box:

a. A plane passing through the points with coordinates (***N***, ***M***, 0), (0, ***M***, ***N***), (***N***, 0, ***M***).

b. A lower half of the surface of the origin-centered sphere with radius ***M***.

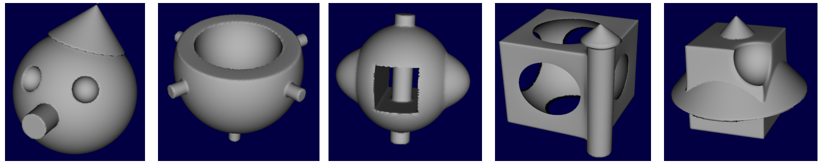
c. A cylindrical surface with radius ***M*** which is aligned with axis Z, and spans from $z_1 = -\boldsymbol{N}$ to $z_1 = \boldsymbol{M}$.

d. A two-side conical surface with radius ***M*** at distance 1 from its apex. The cone is aligned with axis Z, and spans from $z_1 = -1$ to $z_1 = 1$ with the cone apex located at the origin.

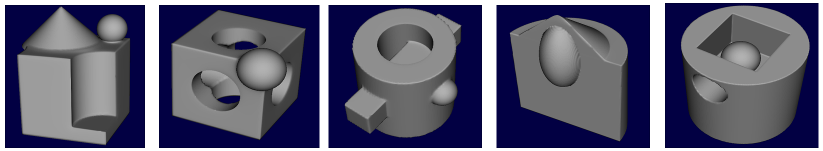
(4 marks)

Lab 4: Implicit Surfaces and Solids

Experiment 2   (Constructive-Solid Geometry)



(1) (2) (3) (4) (5)

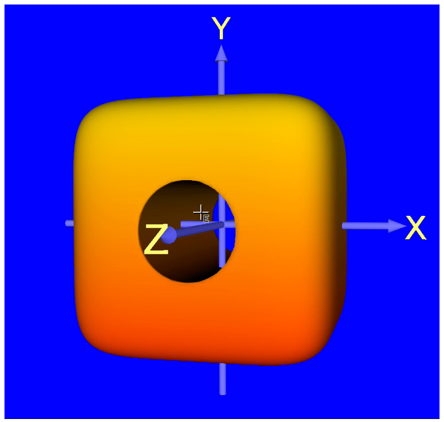


(6) (7) (8) (9) (10)

Lab 4: Implicit Surfaces and Solids

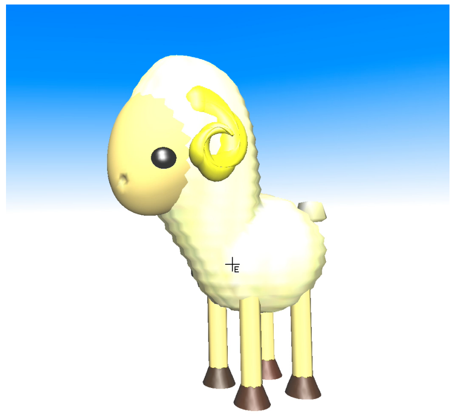
Experiment 2   (variable 3D colors – only FVRML)

This exercise can only be done using FVRML. Color the shape defined in exercise 2 with a variable color. To do it, define in **FMaterial** field a function-defined diffuse color for the whole shape by writing functions $r(u,v,w), g(u,v,w), b(u,v,w)$ where $u = x$, $v = y$, and $w = z$. Use function number ***M*** from Table 1 as a color profile but scale it so that the color values will be located within [0,1] on the visible surfaces of the shape.



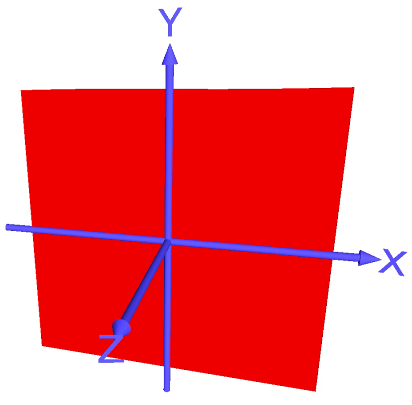
Lab 4: Implicit Surfaces and Solids

Extras



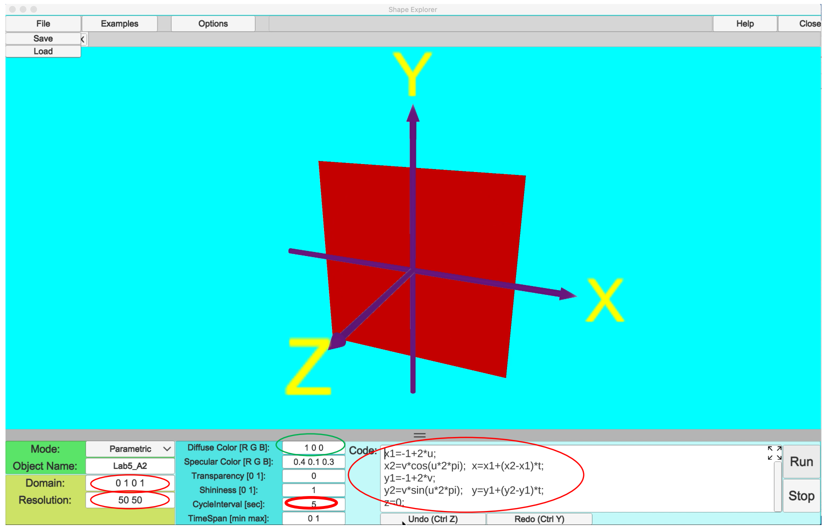
Lab5: Transformations and Motions

```
FShape {
loop TRUE    cycleInterval 5
geometry FGeometry {
definition "
function parametric_x(u,v,w,t)
{ x1=cos(u*pi/2)*cos(v*pi);
  x2=(cos(u*pi/2))^3*(cos(v*pi))^3;
  return x1+(x2-x1)*t;     }
function parametric_y(u,v,w,t)
{ y1=cos(u*pi/2)*sin(v*pi);
  y2=(cos(u*pi/2))*(sin(v*pi))^3;
  return y1+(y2-y1)*t;      }
function parametric_z(u,v,w,t)
{ z1=sin(u*pi/2);
  z2=0.6*(sin(u*pi/2))^5;
  return z1+(z2-z1)*t;       }"
parameters [0 1 0 1]
resolution [30 30]}
appearance FAppearance {
material FMaterial { diffuseColor "r=1; b=0; g=0;"  }   }   }
```



link

Lab5: Transformations and Motions



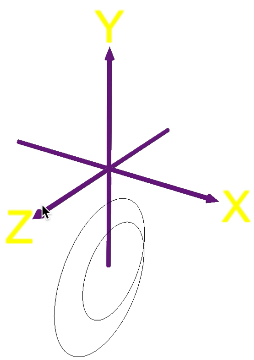
Lab5: Transformations and Motions

Experiment 1 (rotation transformation)

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).

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.

(4 marks)

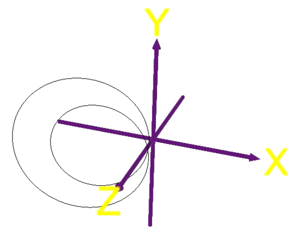


Lab5: Transformations and Motions

Experiment 2 (rotation motion)

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.

(4 marks)



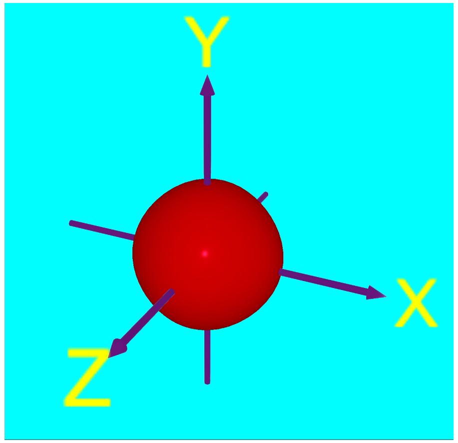
Lab5: Transformations and Motions

Experiment 3 (surface morphing swing animation)

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.

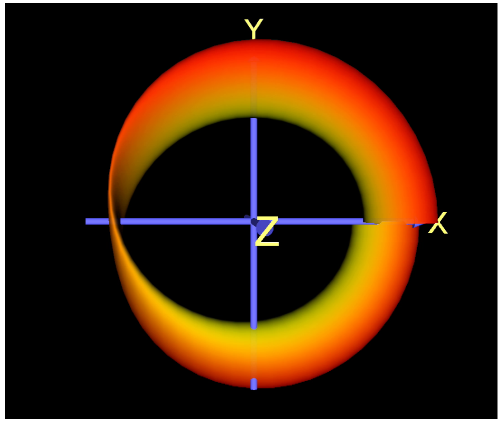
b. Define parametrically using (u,v,t), $y(u,v,t)$, $z(u,v,t)$, $u,v,t \in [0,1]$ a **swing (back and forth)** morphing transformation between surfaces ***M*** and (***N+M***). The morphing animation has to take 5 seconds and has be done with a uniform speed.

(4 marks)



Lab5: Transformations and Motions

Extras



Summary

- VRML will be used as a teaching aid to illustrate theoretical principles and rather as a container for mathematical definitions
- No prior er knowledge of VRML is required
- The same principles of separation of geometry and visual appearance are used in other computer graphics libraries and tools, e.g., OpenGL, WebGL, Java 3D, X3D, POV-Ray, etc.
- FVRML allows for defining content using mathematcal formulas
- Shape Explorer can be used for labs 2, 3, 4 (partially) and 5.