HƯỚNG DẪN SỬ DỤNG ASYMPTOTE – 3D

Huỳnh Văn Thơ

Ngày 26 tháng 7 năm 2019

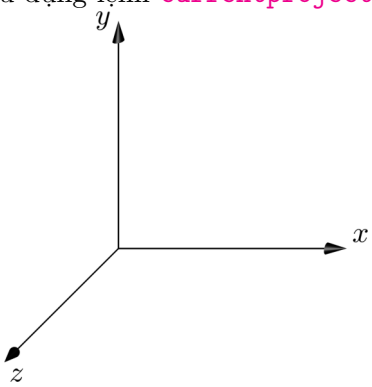
1. Hướng trục – Hướng ánh sáng

1.1. Hướng trục

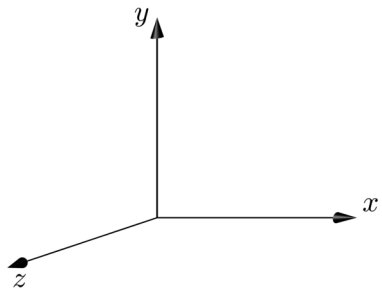
Hướng trục gồm các tùy chọn: `oblique`, `obliqueX`, `obliqueY`, `orthographic`, `perspective`.

`oblique(real goc)` hoặc `oblique` mặc định

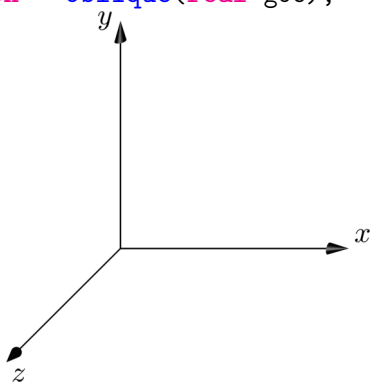
Sử dụng lệnh `currentprojection = oblique;` hoặc `currentprojection = oblique(real goc);`



Code 1: `oblique;`



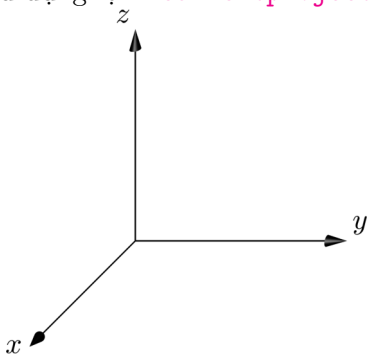
Code 2: `oblique(30);`



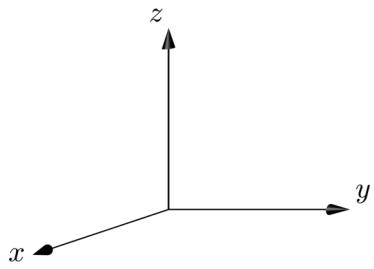
Code 3: `oblique(45);`

`obliqueX(real goc)` hoặc `obliqueX` mặc định

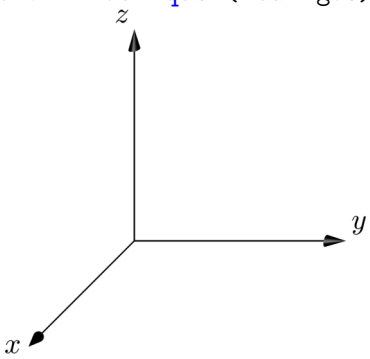
Sử dụng lệnh `currentprojection = obliqueX;` hoặc `currentprojection = obliqueX(real goc);`



Code 4: `obliqueX;`

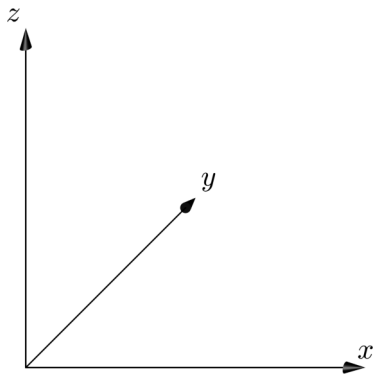


Code 5: `obliqueX(30);`

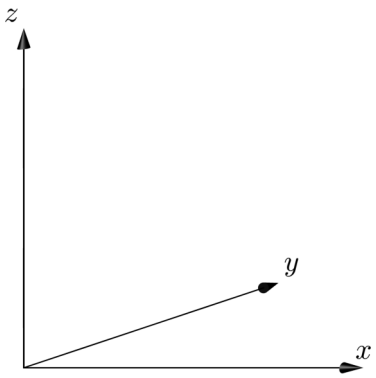


Code 6: `obliqueX(45);`

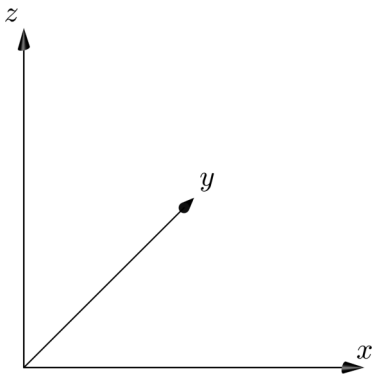
obliqueY(real goc) hoặc obliqueY mặc định



Code 7: obliqueY;



Code 8: obliqueY(30);



Code 9: obliqueY(45);

orthographic

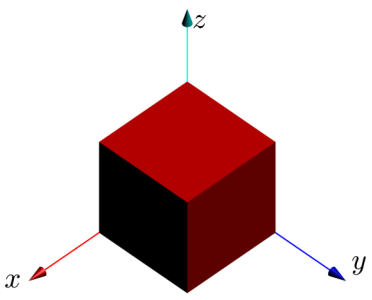
```
orthographic(
        triple camera, // Huong camera
        triple up=Z, // Quay truc Oz len tren
        triple target=0,
        real zoom=1,
        pair viewportshift=0,
        bool showtarget=true,
        bool center=false
);
```

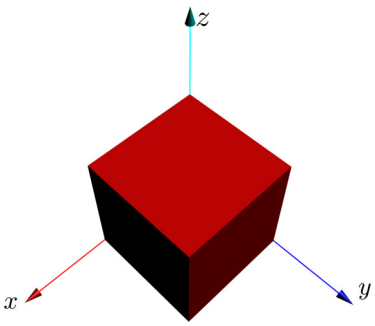
Dùng lệnh currentprojection=orthographic(<Cac tuy chon>);

perspective

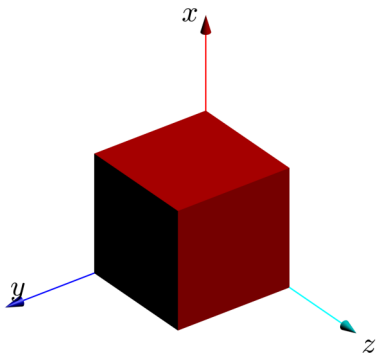
```
perspective(
        triple camera, // Huong camera
        triple up=Z, // Quay truc Oz len tren
        triple target=0,
        real zoom=1,
        real angle=0,
        pair viewportshift=0,
        bool showtarget=true,
        bool autoadjust=true,
        bool center=false
);
```

Dùng lệnh currentprojection=perspective(<Cac tuy chon>);

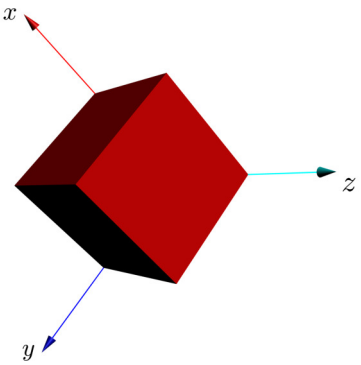




Code 10: *currentprojection = orthographic((1.5,1.5,2));*



Code 11: *currentprojection = perspective((1.5,1.5,2));*



Code 12: *orthographic((1.5,1.5,2),up=X);*

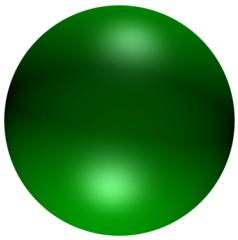
Code 13: *perspective((1.5,1.5,2),up=X+Z);*

1.2. Hướng sáng

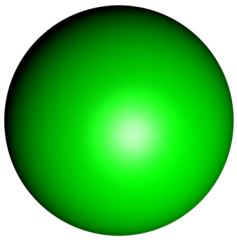
Các hướng sáng có sẵn các tùy chọn: `Headlamp`, `White`, `Viewport`

Dùng lệnh `currentlight=<Cac tuy chon>`

Code 14: *currentlight=Headlamp;*

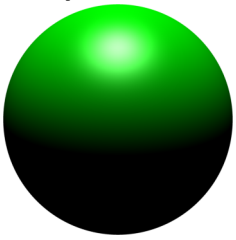


Code 15: *currentlight=White;*

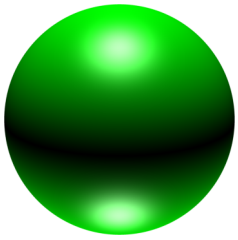


Code 16: *currentlight=Viewport;*

Ngoài ra còn có các tùy chọn khác.



Code 17: *currentlight = light((0,0,3));*



Code 18: *currentlight = light((0,0,3),(0,0,-3));*

2. Export hình ra các định dạng

2.1. Export ra hình 3D trong trong tập tin pdf

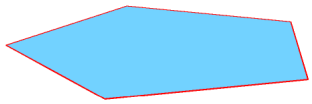
Ở đây, Asy sử dụng editor Texmaker. Trong Texmaker ta nhấn F1 hoặc Tools –> Quick Build thì trong thư mục chứa tập tin vẽ hình có đuôi mở rộng .asy sẽ tự động sinh ra một file pdf, trong file này chứa hình vẽ có thể xoay, phóng to, thu nhỏ trong không gian.

2.2. Export ra hình 2D(png, pdf)

Thêm dòng lệnh sau đây ở đầu tập tin của hình vẽ –> F1 hoặc Quick Build thì trong thư mục chứa tập tin của hình vẽ sẽ tự động sinh ra một file ảnh có đuôi mởi rộng .png hoặc (.pdf).

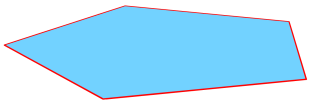
```
settings.outformat="png"; // hay "pdf"
settings.prc=false; //
settings.render=12;
```

trong đó `settings.render=12` là độ phân giả của ảnh png export ra. Thông thường Mando lấy từ 8 đến 24.



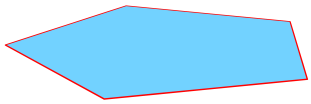
***Code** 19: settings.render=4;*

```
settings.outformat="png";
settings.prc=false;
settings.render=4;
import graph3;
size(4cm);
currentlight=light
    ((0.5,0.5,0.1),
        (-0.5,-0.4,0.5));
path3 p=path3(polygon(5));
draw(p,red);
draw(surface(p),cyan+blue);
```



***Code** 20: settings.render=12;*

```
settings.outformat="png";
settings.prc=false;
settings.render=8;
import graph3;
size(4cm);
currentlight=light
    ((0.5,0.5,0.1),
        (-0.5,-0.4,0.5));
path3 p=path3(polygon(5));
draw(p,red);
draw(surface(p),cyan+blue);
```



***Code** 21: settings.render=20;*

```
settings.outformat="png";
settings.prc=false;
settings.render=4;
import graph3;
size(4cm);
currentlight=light
    ((0.5,0.5,0.1),
        (-0.5,-0.4,0.5));
path3 p=path3(polygon(5));
draw(p,red);
draw(surface(p),cyan+blue);
```

Khi phóng to tài liệu này lên khoảng 1200%, Mando sẽ thấy độ mịn tăng dần của đường viền màu đỏ ở 3 hình.

3. Gói `Three`

3.1. triple, guide3, path3, surface và size3

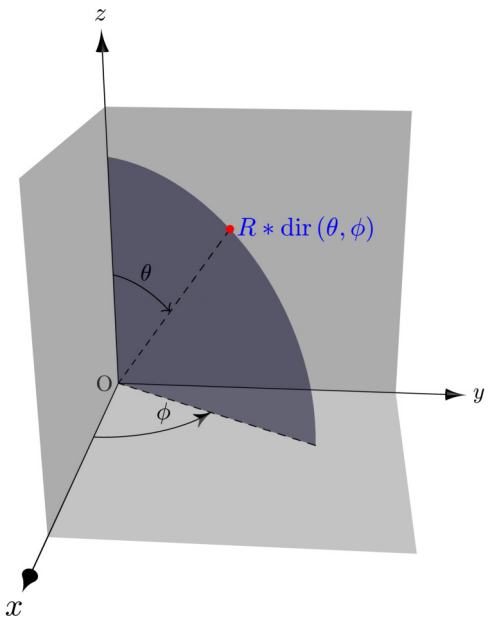
Đó các đối tượng chính trong moduls `three` tương ứng như trong mặt phẳng: `guide3 -- guide`, `path3 -- path`, `triple -- pair` và `size3 -- size`

Hàm triple

Hàm lấy điểm trong không gian $Oxyz$ là bộ ba `(x,y,z)`. Ví dụ `triple A = (1,2,-3);` định nghĩa điểm M có tọa độ $(1; 2; -3)$.

Tọa độ cầu của điểm

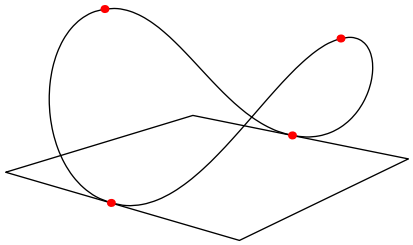
Mando có thể định nghĩa điểm bằng tọa độ cầu gồm 3 tham số: bán kính R, góc quét θ từ trục Oz xuống mặt phẳng Oxy và góc ϕ quét từ trục Ox sang mặt phẳng Oyz theo cú pháp sau: `triple M=R*dir(theta, phi)`.



3.2. Hàm surface – size3

Hàm tạo bề mặt của một path3 kín.

```
surface surface(
        path3 p, // path3
        triple[] internal=new triple[],
        triple[] normals=new triple[],
        pen[] colors=new pen[], // mau sac
        bool3 planar=default
);
```

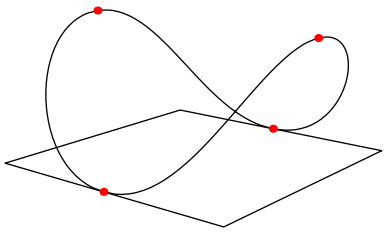


***Code** 22: Đường cong nối các điểm*

```
settings.render=0;
import three;
size3(10cm, 10cm, 2cm);
path3 g =(1,0,0)..(0,1,1)..(-1,0,0)
    ..(0,-1,1)..cycle; // Day la path cong noi
    cac diem
draw(g); // Ve path g
draw(((-1,-1,0)--(1,-1,0)--(1,1,0)--(-1,1,0)
    --cycle)); // Ve hinh vuong o duoi
dot(g,red); // Ve cac diem tren duong cong
```

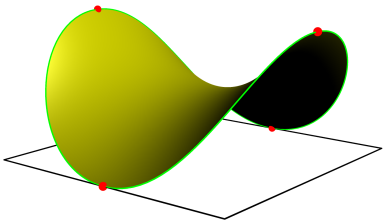
Hàm size3 lấy đơn vị trên 3 trục. Có thể dùng size.

```
void size3(
        picture pic=currentpicture,
        real x,
        real y=x,
        real z=y,
        bool keepAspect=pic.keepAspect
);
```



***Code** 23: Đường cong nối các điểm 2*

```
settings.render=0;
import three; //De khong phuc tap
size(5cm); // ta dung kieu don vi size, thong
    la 5cm
path3 g=(1,0,0)..(0,1,1)..(-1,0,0)..(0,-1,1)
    ..cycle;
draw(g);
draw(((-1,-1,0)--(1,-1,0)--(1,1,0)--(-1,1,0)
    --cycle));
dot(g,red);
```



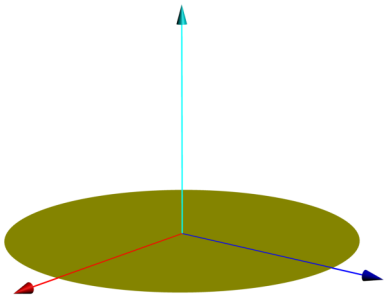
Code 24: *Mặt tạo bởi đường cong khép kín*

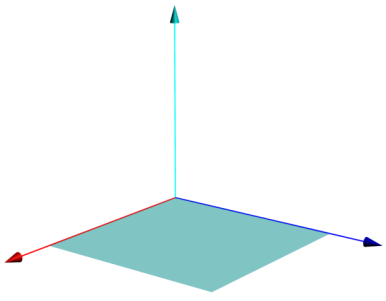
```
settings.render=16;
import three; //
size(5cm); // ta dung kieu don vi size, thong la 5cm
path3 g=(1,0,0)..(0,1,1)..(-1,0,0)..(0,-1,1)..cycle;
draw(g,green); // Ve path3 g
dot(g,red); // Cac diem g di qua
draw(((-1,-1,0)--(1,-1,0)--(1,1,0)--(-1,1,0)--cycle));
//------- Ve mat sinh boi g
surface Sur = surface(g);
draw(Sur,yellow); // Ve be mat se noi ky o phan sau...
```

3.3. Một số bề mặt – surface có sẵn trong modul `three`

Trong modul `three` đã định nghĩa sẵn một số đối tượng cơ bản thường dùng trong hình học không gian.

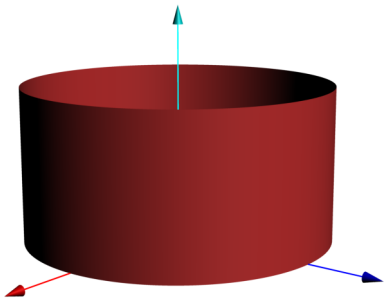
1) `unitdisk`; Đĩa tròn đơn vị trong mặt phẳng (Oxy).
2) `unitplane`; Hình vuông đơn vị trong mặt phẳng (Oxy).
3) `unitcylinder`; Măt trụ đơn vị có $R = 1$, $h = 1$.
4) `unitcube`; Khối lập phương đơn vị.
5) `unitcone`; Hình nón đơn vị.
6) `unitsolidcone`; Khối nón đơn vị.
7) `unitsphere`; Mặt cầu đơn vị.
8) `unithemisphere`; Nửa mặt cầu đơn vị.
9) `path3 path3(polygon(n))`; Đa giác đều n cạnh 1 đơn vị.
10) `surface surface(path3(polygon(n)))`; Bề mặt của đa giác đều n cạnh.





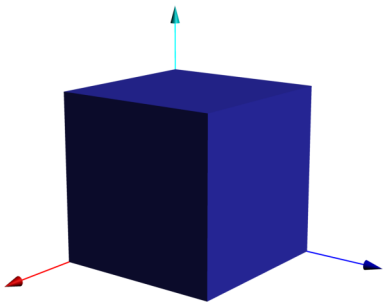
Code 25: Dĩa đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.3*X,red,Arrow3()); //-- Ox
draw(O--1.3*Y,blue,Arrow3()); //-- Oy
draw(O--1.3*Z,cyan,Arrow3()); //-- Oz
//------- Ve dia don vi
draw(unitdisk,yellow);
```



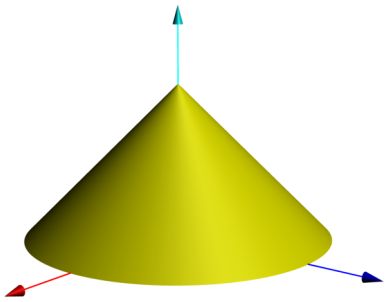
Code 26: Hình vuông đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.3*X,red,Arrow3());
draw(O--1.3*Y,blue,Arrow3());
draw(O--1.3*Z,cyan,Arrow3());
//------- Ve hinh vuon don vi
draw(unitplane,cyan+opacity(0.5));
```



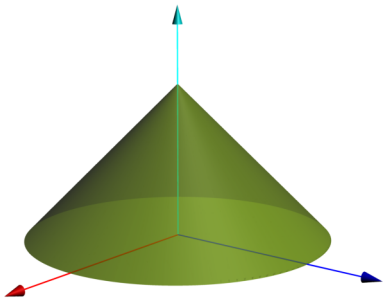
Code 27: Mặt trụ đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
//------- Ve mat tru don vi
draw(unitcylinder,mediumred);
```



Code 28: Khối lập phương đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
//------- Ve khoi lap phuong don vi
draw(unitcube,mediumblue);
```

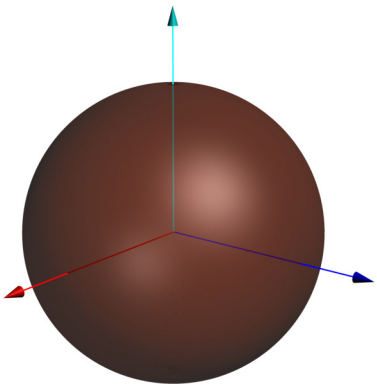


Code 29: Hình nón đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
//------- Ve hinh non don vi
draw(unitcone,yellow);
```

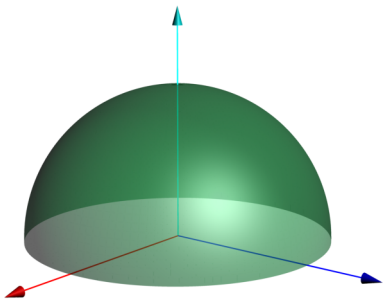
Code 30: Khối nón đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
pen p = rgb(0.5,0.7,0);
draw(unitsolidcone,p + opacity(0.6));
```



Code 31: Khối cầu đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
//------- Ve hinh non don vi
pen p = rgb(0.5,0.1,0);
draw(unitsphere,p+opacity(0.6));
```



Code 32: Nửa mặt cầu đơn vị

```
settings.render=12;
import three;
size(5cm);
draw(O--1.5*X,red,Arrow3());
draw(O--1.5*Y,blue,Arrow3());
draw(O--1.5*Z,cyan,Arrow3());
//------- Ve hinh non don vi
pen p = rgb(0.1,0.7,0.3);
draw(unithemisphere,p+opacity(0.6));
```

3.4. Các đối tượng khác: `circle`, `arc`, `plane`

Trong moduls `three` định nghĩa sẵn các điểm `O=(0,0,0)`, `X=(1,0,0)`, `Y=(0,1,0)` và `Z=(0,0,1)`. Sau này đặt tên cho điểm, Mando không nên đặt tên trùng với các điểm trên.

3.4.1. unitcircle3 – Đường tròn đơn vị trong mặt phẳng Oxy

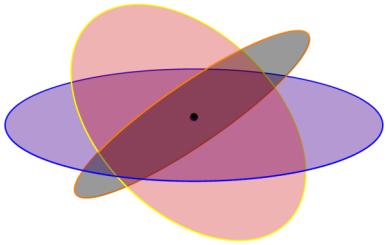
```
path3 unitcircle3 = X..Y..-X..-Y..cycle;
```

3.4.2. circle – Đường tròn có tâm, bán kính và trục

```
path3 circle(
       triple C, // Tam C
       real r, // ban kinh r
       triple normal = Z // truc Oz, co the thay bang truc khac nhu: X,Y, X+Y, (1,1,0.5)
);
```

Code 33: Hình tròn

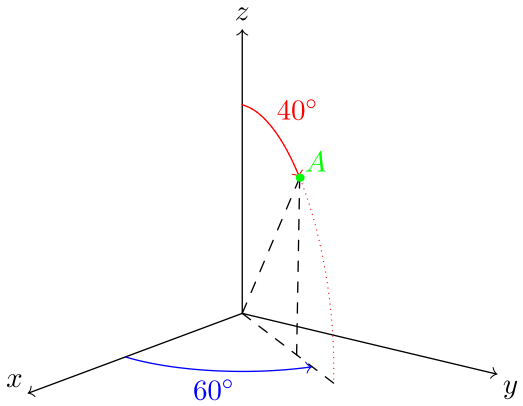
```
settings.render=12; // Dung che do opacity( )
import three;
size(5cm); dot(O); // goc toa do
path3 cirA = unitcircle3; //---- duong tron cirA
draw(cirA,blue); // Ve cirA mau blue
draw(surface(cirA),purple+opacity(0.4)); // To hinh tron cirA
    mau purple, do mo 0.
path3 cirB=circle(O,0.75,Z-Y);//----- duong tron cirB
draw(cirB,orange);
draw(surface(cirB),yellow+opacity(0.4));
path3 cirC=circle(O,0.75,Z+Y);//
draw(cirC,yellow);
draw(surface(cirC),red+opacity(0.3));
```



3.5. arc – Cung

3.5.1. Cung tạo bởi tọa độ cầu

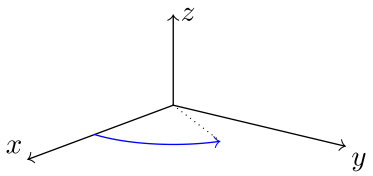
```
path3 arc(
        triple c, // Tam duong tron chua cung
        real r, // Ban kinh cung
        real alpha1,
        real beta1, // Diem r*dir(alpha1,beta1)
        real alpha2,
        real beta2, // Diem r*dir(alpha2,beta2)
        triple normal=O //
);
```



Code 34: Cung

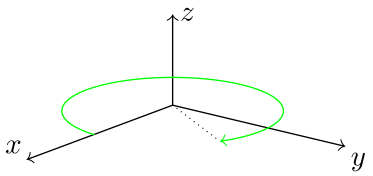
```
settings.render=0;
import three;
size(7cm);
DefaultHead3=TeXHead3;
//---------Truc toa do
draw(O--2*X,Arrow3()); label("$x$",2*X,NW);
draw(O--2*Y,Arrow3()); label("$y$",2*Y,SE);
draw(O--2*Z,Arrow3()); label("$z$",2*Z,N);
```

```
//----- Thong so lay cung
real alpha1 = 0, beta1 = 0;
real alpha2 = 40, beta2=60;
//------ Cung mau do
real R=1.5; real r=1.2;
path3 aD = arc(O,R,alpha1,beta1,alpha2,beta2);
draw(Label(format("$%f^\circ$",alpha2),align=N),aD,red,Arrow3());
//----- Diem A
triple A = R*dir(alpha2,beta2);
dot("$A$",A,NE,green);
//-----Cung cham do
path3 acD=arc(O,R,alpha1,beta1,90,beta2);
draw(acD,red+dotted);
//----- Cung xanh
path3 aX = arc(O,r,90,beta1,90,beta2);
draw(Label(format("$%f^\circ$",beta2),align=S),aX,blue,Arrow3());
//--- Hinh chieu cua A len (Oxy)
triple H = (A.x,A.y,0);
draw(O--R*dir(90,beta2),dashed);
draw(A--H^^O--A,dashed);
```



Code 35: Cung quét ngược chiều kim đồng hồ

```
settings.render=0;
import three;
DefaultHead3=TeXHead3;
size(5cm);
draw(O--2*X,Arrow3()); // Truc Ox
draw(O--2*Y,Arrow3()); // Truc Oy
draw(O--Z,Arrow3()); // Truc Oz
label("$x$", 2*X, NW); // Nhan truc
label("$y$", 2*Y, SE);
label("$z$", Z, E);
real R=1.2; // ban kinh cung
path3 ar=arc(O,R,90,0,90,60);
draw(ar,blue,Arrow3());
draw(O--R*dir(90,60),dotted);
```

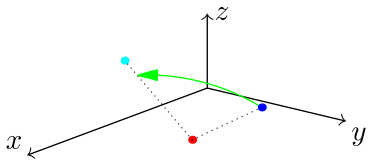


Code 36: Cung quét cùng chiều kim đồng hồ

```
settings.render=0;
import three;
DefaultHead3=TeXHead3;
size(5cm);
draw(O--2*X,Arrow3()); // Truc Ox
draw(O--2*Y,Arrow3()); // Truc Oy
draw(O--Z,Arrow3()); // Truc Oz
label("$x$", 2*X, NW); // Nhan truc
label("$y$", 2*Y, SE); label("$z$", Z, E);
real R=1.2; // ban kinh cung
path3 ar=arc(O,R,90,0,90,60,CW); // CW cung
    quet cung chieu kim dong ho
draw(ar,green,Arrow3());
draw(O--R*dir(90,60),dotted);
```

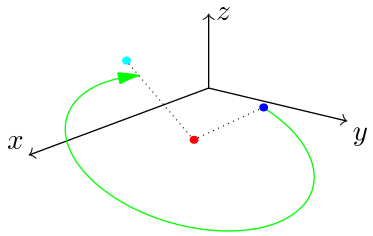
3.5.2. Cung quét

```
path3 arc(
       triple C, //Tam
       triple A, // Diem bat dau
       triple B, // Diem ket thuc
       triple normal=O, //
       bool direction=CCW // Mat dinh nguoc chieu kim dong ho
);
```



Code 37: *Cung quét ngược chiều kim đồng hồ*

```
settings.render=0;
import three;
DefaultHead3=TeXHead3;
size(5cm);
draw(O--2*X,Arrow3()); // Truc Ox
draw(O--2*Y,Arrow3()); // Truc Oy
draw(O--Z,Arrow3()); // Truc Oz
label("$x$", 2*X, NW); // Nhan truc
label("$y$", 2*Y, SE);
label("$z$", Z, E);
triple C, A, B;
C=(1.5,1,0); A=(0.2,1,0);
B=(1,-0.5,0.5);
path3 ar=arc(C, A, B);
draw(ar,green,Arrow3);
dot(C,red); dot(A,blue); dot(B,cyan);
draw(C--B^^C--A,dotted);
```



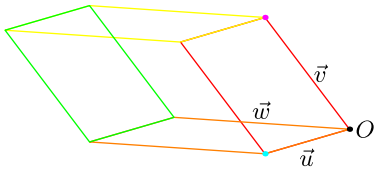
Code 38: *Cung quét cùng chiều kim đồng hồ*

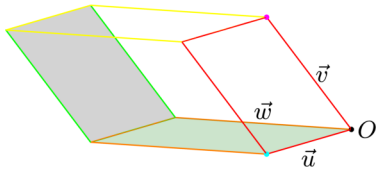
```
settings.render=0;
import three;
DefaultHead3=TeXHead3;
size(5cm);
draw(O--2.8*X,Arrow3()); // Truc Ox
draw(O--2*Y,Arrow3()); // Truc Oy
draw(O--Z,Arrow3()); // Truc Oz
label("$x$", 2.8*X, NW); // Nhan truc
label("$y$", 2*Y, SE);
label("$z$", Z, E);
triple C, A, B;
C=(1.5,1,0); A=(0.2,1,0);
B=(1,-0.5,0.5);
path3 ar=arc(C, A, B, CW); // quet cung chieu
draw(ar,green,Arrow3);
dot(C,red); dot(A,blue); dot(B,cyan);
draw(C--B^^C--A,dotted);
```

3.5.3. plane – Hình bình hành

Hình bình hành được tạo bởi hai tia OU và OV.

```
path3 plane(
       triple U, // Diem U
       triple V, // Diem V
       triple O = O // Mat dinh O goc toa do
);
```





Code 39: *Hình bình hành*

```
settings.render=0;
import three; size(5cm);
defaultpen(fontsize(9pt));
currentprojection = orthographic(.7,2,0.3);
//---- 3 diem
triple U , V, W;
       U = (1,1,0);
       V = (1,1,1);
       W = cross(U,V); // Tich co huong cua
           vecto OU, OV
path3 Pa, Pb, Pc, Pd;
       Pa = plane(U,V,O), // red
       Pb = plane(U,W,O), // orange
       Pc = plane(U,W,V), // tinh tien Pb theo
            vecto V
       Pd = plane(U,V,W); // tinh tien Pa theo
            vecto W
draw(Pa,red); // hinh binh hanh red
draw(Pb,orange); //hinh binh hanh orange
draw(Pc,yellow); //hinh binh hanh yellow
draw(Pd,green); //hinh binh hanh green
dot("$O$",O,E);
dot(U,cyan+2bp); // Diem U, cyan
dot(V,magenta+2bp); // Diem V, magenta
label("$\vec u$", O--U,S);
label("$\vec v$", O--V,E);
label("$\vec w$", O--W,0.6*N);
```

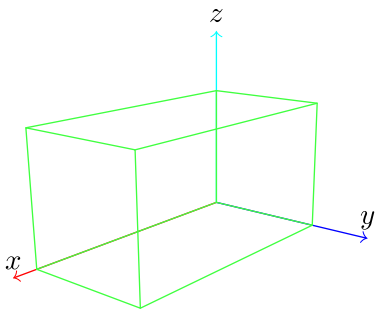
Code 40: *Tô bề mặt hình bình hành*

```
settings.render=12;
import three; size(5cm);
defaultpen(fontsize(9pt));
currentprojection = orthographic(.7,2,0.3);
//---- 3 diem
triple U , V, W;
       U = (1,1,0);
       V = (1,1,1);
       W = cross(U,V); //
path3 Pa, Pb, Pc, Pd;
       Pa = plane(U,V,O); // red
       Pb = plane(U,W,O); // orange
       Pc = plane(U,W,V); //
       Pd = plane(U,V,W); //
//--------------------
draw(Pa,red); draw(Pb,orange);
//--- To be mat Pb
draw(surface(Pb),green+opacity(0.2)); //
draw(Pc,yellow); draw(Pd,green);
//--- To be mat Pd
draw(surface(Pd),mediumgray+opacity(0.3));
dot("$O$",O,E);
dot(U,cyan+2bp); // Diem U, cyan
dot(V,magenta+2bp); // Diem V, magenta
label("$\vec u$", O--U,S);
label("$\vec v$", O--V,E);
label("$\vec w$", O--W,0.6*N);
```

3.5.4. Hình hộp

`path3[] box(triple U, triple V);` Hình hộp có 2 đỉnh đối diện U và V.

`path3[] box(O,(1,1,1));` Hình lập phương đơn vị.



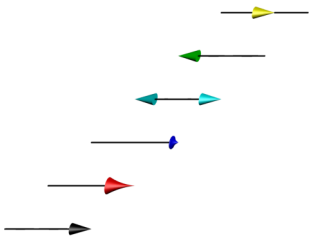
Code 41: *Hình hộp*

```
settings.render=0;
import three; size(5cm);
DefaultHead3=TeXHead3;
draw(O--2.2*X,red,Arrow3()); // Ve truc Ox
label("$x$",2.2*X,N); // Gan nhan x tre truc Ox
draw(O--(1.5*Y),blue,Arrow3());
label("$y$",1.5*Y,N);
draw(O--1.5*Z,cyan,Arrow3());
```

```
label("$z$",1.5*Z,N);
//---- Hinh hop
triple A = 2X+Y+Z; // A=(2;1;1)
path3[] h = box(O,A); // Hinh hop co 2 dinh doi dien O va A
draw(h,mediumgreen); // Ve hinh hop
```

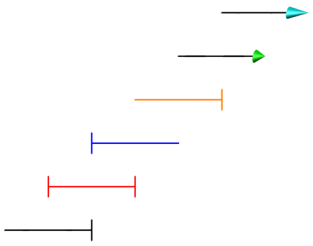
3.6. Mũi tên – Arrow3, Barr

Một số mũi tên và thanh ngang có sẵn trong modul `three`



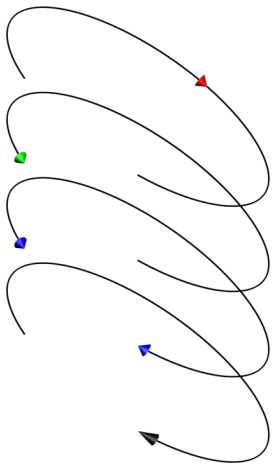
***Code** 42*: *Mũi tên*

```
settings.render=12;
import three;
currentprojection=obliqueX;
size(4cm);
draw((2,0,0)--(2,1,0),Arrow3(DefaultHead3));
draw((1,0,0)--(1,1,0),Arrow3(HookHead3,red));
draw((0,0,0)--(0,1,0),Arrow3(TeXHead3,blue));
draw((-1,0,0)--(-1,1,0),Arrows3(cyan));
draw((-2,0,0)--(-2,1,0),BeginArrow3(green));
draw((-3,0,0)--(-3,1,0),MidArrow3(yellow));
```



***Code** 43*: *Mũi tên, barr*

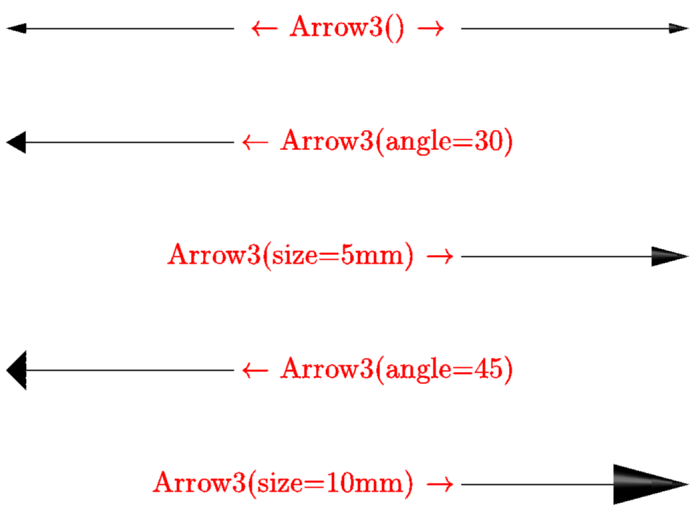
```
settings.render=12;
import three;
currentprojection=obliqueX;
size(4cm);
draw((2,0,0)--(2,1,0),Bar3);
draw((1,0,0)--(1,1,0),red,Bars3);
draw((0,0,0)--(0,1,0),blue,BeginBar3);
draw((-1,0,0)--(-1,1,0),orange,EndBar3);
draw((-2,0,0)--(-2,1,0),ArcArrow3(green));
draw((-3,0,0)--(-3,1,0),Arrow3(cyan));
```



***Code** 44*: *Cung chứa mũi tên*

```
settings.render=12;
import three;
currentprojection=obliqueX;
size(7cm);
draw(arc(O,(1,-0.5,0.75),(1,0.5,0),CW),Arrow3(DefaultHead3));
draw(arc((0,0,0.75),(1,-0.5,1.5),(1,0.5,0.75),CW),ArcArrows3(blue));
draw(arc((0,0,1.5),(1,-0.5,2.25),(1,0.5,1.5),CW),BeginArcArrow3(green));
draw(arc((0,0,2.25),(1,-0.5,3),(1,0.5,2.25),CW),MidArcArrow3(red));
```

3.6.1. Tùy biến mũi tên



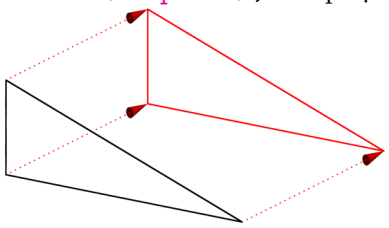
Code 45: Tùy biến mũi tên

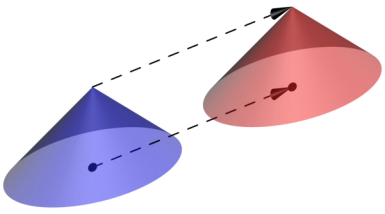
```
import three;
currentprojection=orthographic(0,5,0);
unitsize(3cm);
draw(X--O, Arrow3());
draw(X-Z--O-Z, Arrow3(size=5mm));
draw(X-2*Z--O-2*Z, Arrow3(size=10mm));
draw(2*X--3*X, Arrow3());
draw(2*X-0.5*Z--3*X-0.5*Z, Arrow3(angle=30));
draw(2*X-1.5*Z--3*X-1.5*Z, Arrow3(angle=45));
label("$\leftarrow$ Arrow3() $\rightarrow$", (1.5,0,0), red);
label("Arrow3(size=5mm) $\rightarrow$", (1,0,-1),W, red);
label("Arrow3(size=10mm) $\rightarrow$", (1,0,-2),W, red);
label("$\leftarrow$ Arrow3(angle=30)", (2,0,-0.5), E, red);
label("$\leftarrow$ Arrow3(angle=45)", (2,0,-1.5), E, red);
```

3.7. các phép biến hình

3.7.1. Phép tịnh tiến

`transform3 shift(triple u);` Phép tịnh tiến theo $\vec{u}$.





Code 46: Tịnh tiến tam giác ABC

```
settings.render=12; import three;
currentprojection=obliqueX; size(5cm);
//--- Hai vec to tinh tien
triple u=(-1.5,0.75,0), v=(0,-0.5,-1);
//--- Phep tinh tien
transform3 Ta=shift(u);
//--- 3 diem
triple A=(0,0,0), B=(0,0,1), C=(1,3,0);
path3 p=A--B--C--cycle, //-- tam giac ABC
    g=Ta*p; // p qua phep tinh tien u
draw(p); draw(g,red); // Ve p ,g
draw(A--Ta*A,red+dotted,Arrow3);
draw(B--Ta*B,red+dotted,Arrow3);
draw(C--Ta*C,red+dotted,Arrow3);
```

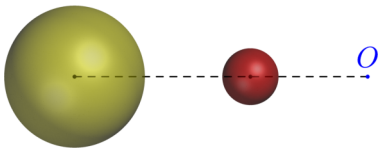
Code 47: Hình nón

```
settings.render=12;
import three;
currentprojection = obliqueX;
size(5cm);
triple N=(0,0,1),S=(0,0,0);
//--- Vecto u
triple u=(0,2.5,1);
//---- Phep tinh tien theo u
transform3 Tt=shift(u);
draw(unitcone,blue+opacity(0.5));
draw(Tt*unitcone,red+opacity(0.5));
draw(N--Tt*N,dashed,Arrow3);
draw(S--Tt*S,dashed,Arrow3);
dot(S);dot(Tt*S);
```

3.7.2. Phép vị tự

a) `transform scale3(real k);` Phép vị tự tâm O, tỷ số k.

b) `transform scale(real x, real y, real z);` Phép co – giãn tỷ số x, y, z theo 3 trục Ox, Oy, Oz.



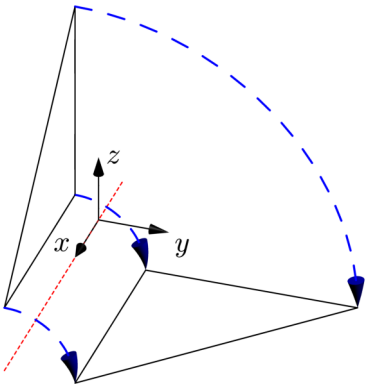
Code 48: Mặt cầu qua phép vị tự

```
settings.render=12;
import three;
import solids; // goi nay chua ham lay mat cau
currentprojection orthographic(0,1,0.5);;
size(5cm);
dotfactor = 3; // Tang kich thuoc dau cham
triple I = 5*X; // Tam duong tron mau yellow
real R = 1.2; // Ban kinh duong tron yellow
revolution Sp = sphere(I,R); // Hinh cau yellow
draw(surface(Sp),yellow+opacity(0.5)); // Ve mat cau yellow
dot("$O$",O,N,blue); // Tam vi tu O
draw(scale3(0.4)*surface(Sp),red+opacity(.6)); // Mat cau red
draw(O--I,linetype("5 5"));
dot(I); dot(scale3(0.3)*I); // Tam mat cau red
```

3.8. Phép quay

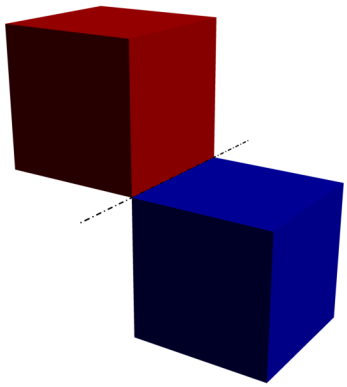
a) `tranform3 rotate(real angle, triple v);` Phép quay tâm O, góc angle, với trục là $\vec{v}$.

b) `tranform3 rotate(real angle, triple u, triple v);` Phép quay góc angle, trục là đường thẳng uv.



Code *49*: *Quay tam giác*

```
settings.render=12;
import three;
size(5cm);
currentprojection=orthographic(3,1,2);
triple I=(1,0,0);
transform3 r=rotate(-90,I);
triple A=(1,0,1), B=(4,0,1), C=(1,0,4);
path3 tri=A--B--C--cycle; // tam giac
path3 trQ=r*tri; // Quay tam giac
draw(tri^^trQ); // Ve hai tam giac
draw(O--X,Arrow3); // truc Ox
draw(O--Y,Arrow3); // Truc Oy
draw(O--Z,Arrow3); // truc Oz
label("$x$", X, NW); // Nhan
label("$y$", Y, SE);
label("$z$", Z, E);
pen dotteddash=linetype("4 4 4 4"),
p=.8bp+blue+dashed; // kieu, mau
draw((-1,0,0)--(4,0,0),red+dotteddash);
draw(arc((A.x,0,0),A,r*A),p,Arrow3);
draw(arc((B.x,0,0),B,r*B),p,Arrow3);
draw(arc((C.x,0,0),C,r*C),p,Arrow3);
```

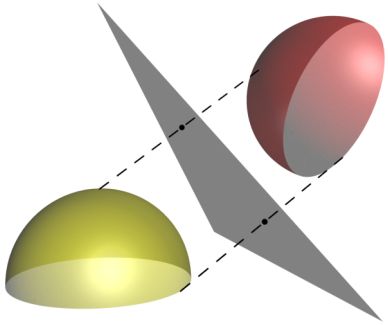


Code *50*: *Quay khối lập phương quanh trục AB*

```
settings.render=12;
import three;
size(5cm);
//currentprojection=obliqueX;
triple A=(0,1,0),
       B=(1,1,0);
transform3 r=rotate(180,A,B);
draw(unitcube,red);
draw(r*unitcube,blue); // unitcube qua
pen dotteddash=linetype("0 4 4 4");
draw((-0.5,1,0)--(1.5,1,0),dotteddash);
```

3.9. Phép đối xứng qua mặt phẳng xác định bởi 3 điểm

`transform3 reflect(triple A, triple B, triple C);` Phép đối xứng qua mặt phẳng xác định bởi 3 điểm A, B, C



Code *51*: *Đối xứng nửa mặt cầu qua mặt phẳng*

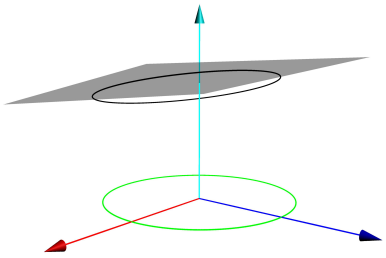
```
settings.render=12;
import three;
size(5cm); dotfactor =4;
triple A=(-3,0,0), B=(0,3,0), C=(0,0,3);
transform3 sym=reflect(A,B,C);
path3 tr=A--B--C--cycle; // Tam giac ABC
draw(surface(tr),opacity(0.2)); // To tam giac ABC
draw(unithemisphere,yellow+opacity(0.5)); // Nua mat cau vang
draw(sym*unithemisphere,red+opacity(0.5)); // Doi xung nua mat cau vang --> do
triple E=(0,0,1), F=(0,1,0); // Hai diem thuoc nua cau vang
draw(E--sym*E,dashed); // Duong noi anh va diem anh
draw(F--sym*F,dashed);
dot((E+sym*E)/2); // hai hai trung diem
dot((F+sym*F)/2);
```

3.10. Phép chiếu song song

1) `transform3 planeproject(path3 p, triple dir=normal(p));` Chiếu vuông góc lên mặt phẳng chứa path p. Nếu thay `normal(p)` bằng một `triple` (vectơ) thì ta có được phương chiếu song song.

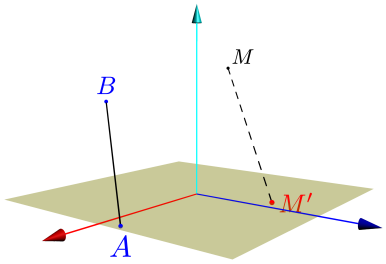
   `triple normal(path3 p);` Vectơ pháp tuyến của mặt phẳng chứa path p.

2) `transform3 planeproject(XY*unitsquare3);` Chiếu vuông góc lên mặt phẳng Oxy, tương tự: `YZ*unitsquare3` và `ZX*unitsquare3`.



***Code** 52: Chiếu vuông góc*

```
import three; size(5cm);
//------ Cac truc
draw(O--2*X,red, Arrow3);
draw(O--2*Y,blue,Arrow3);
draw(O--2*Z,cyan,Arrow3);
triple u,v,w;
       u=(2,-1,0);
       v=(0,2,0.5);
       w=(-0.7,-0.5,1);
path3 p = plane(u,v,w); // Mat phang chieu
draw(surface(p),yellow+opacity(0.4));
//-- Chieu vuong goc duong tron don vi len p
draw(unitcircle3,green); // duong tron don vi
transform3 Prj = planeproject(p, dir=normal(
    p));
path3 Fi=Prj*unitcircle3; // Chieu vuong goc
    len p
draw(Fi); // anh duong tron
```

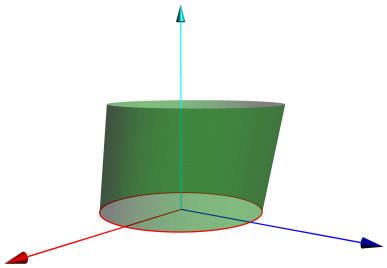


***Code** 53: Chiếu song song*

```
import three; size(5cm); dotfactor =3;
defaultpen(fontsize(10pt));
draw(O--3*X,red, Arrow3); // Cac truc
draw(O--3*Y,blue,Arrow3);
draw(O--3*Z,cyan,Arrow3);
path3 hv=path3(polygon(4)); //hv don vi
draw(surface(scale3(3)*hv),yellow+opacity
    (0.4));
triple A=(2,0.3,0),B=(1,-1,1.5),M
    =(-1.6,-0.5,2);
dot("$A$",A,S,blue); dot("$B$",B,N,blue);
dot("$M$",M,NE); draw(A--B);
triple vector=B-A; // vecto AB
//-- Phep chieu ssong theo phuong AB
transform3 Pj=planeproject(hv, dir=vector);
triple Mp=Pj*M; //-- Chieu diem M
draw(M--Mp,dashed); dot("$M'$",Mp,red+
    linewidth(2bp));
```

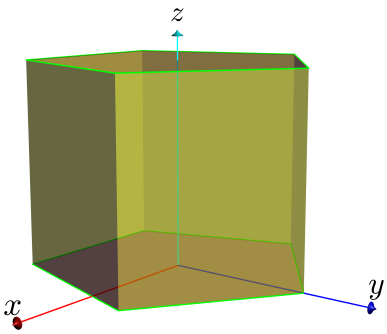
3.11. Phép kéo theo trục

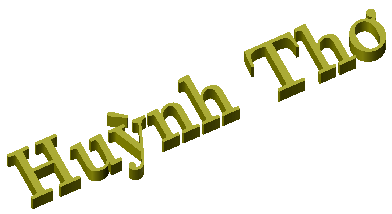
1) `surface extrude(path p, triple axis=Z);` Tạo bề mặt bằng cách kéo path p theo trục Oz, Mando cũng có thể thay `Z` bằng một `triple` khác.
2) `surface extrude(Label L, triple axis=Z);` Tạo bề mặt bằng cách kéo `Label L` theo trục Oz.



Code 54: *Mặt trụ xuyên*

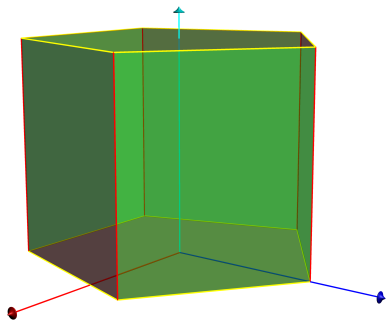
```
import three; size(5cm);
//--- Cac truc
draw(O--3*X,red, Arrow3);
draw(O--3*Y,blue,Arrow3);
draw(O--3*Z,cyan,Arrow3);
//--- Duong tron don vi*1.2
path3 cir =scale3(1.2)*unitcircle3;
draw(cir,red); // Ve cir
surface Sur=extrude(cir,axis=(0.3,0.5,1.6));
draw(Sur,green+opacity(0.5));
```





Code 55: *Text 3D*

```
import three; size(5cm);
//--- font tieng Viet
texpreamble("\usepackage[utf8]{vietnam}");
currentprojection=orthographic(-1,-2,4);
currentlight=White;
//-- Text,trong moi truong lstlising ko choi
    duoc chu co dau
Label L=Label("Huynh Tho",(0,0));
surface s=extrude(L,3*Z);
draw(s,red+green);
```



Code 56: Ngũ giác đều

```
import three; size(5cm);
DefaultHead3=TeXHead3;
//----- Cac truc
draw(O--1.5*X,red,Arrow3());
label("$x$",1.5*X,N);
draw(O--(1.5*Y),blue,Arrow3());
label("$y$",1.5*Y,N);
draw(O--1.6*Z,cyan,Arrow3());
label("$z$",1.6*Z,N);
//-- Giac giac deu don vi trong (Oxy)
path3 Ngiac = path3(polygon(5)),
         NGiac = shift(1.4*Z)*Ngiac;// tinh
             tien 1.4*Z
draw(Ngiac^^NGiac,green);
//-- Ve 2 be mat cua 2 ngu giac
draw(surface(Ngiac),0.6*red+opacity(0.5));
draw(surface(NGiac),0.6*red+opacity(0.5));
//-- Khoi lang tru tao boi ngu giac don vi
surface MatNgiac=extrude(Ngiac,1.4*Z);
draw(MatNgiac,yellow+opacity(0.5));
```

Code 57: Ngũ giác đều 2

```
import three;size(5cm);DefaultHead3=TeXHead3;
//----- Cac truc
draw(O--1.5*X,red,Arrow3());
draw(O--(1.5*Y),blue,Arrow3());
draw(O--1.6*Z,cyan,Arrow3());
//-- Giac giac deu don vi trong (Oxy)
path3 Ngiac = path3(polygon(5)),
         NGiac = shift(1.4*Z)*Ngiac;// tinh
             tien 1.4*Z
draw(Ngiac^^NGiac,yellow);
//-- Ve 2 be mat cua 2 ngu giac
draw(surface(Ngiac),0.6*red+opacity(0.5));
draw(surface(NGiac),0.6*red+opacity(0.5));
//-- Khoi lang tru tao boi ngu giac don vi
surface MatNgiac=extrude(Ngiac,1.4*Z);
draw(MatNgiac,green+opacity(0.5));
//-- Noi cac canh ben
for(int i =0; i<=4; ++i){
 draw(point(Ngiac,i)--point(NGiac,i),red);}
//-- point(Ngiac,i) dinh thu i cua Ngiac
```

3.12. Một số tiện ích của hàm draw và path3

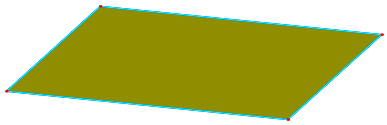
3.12.1. Đối với hàm draw

```
void draw(
       picture pic=currentpicture,
       surface s, // Be mat tao boi path3
       int nu=1, // So duong ngang cua luoi
       int nv=1, // So duong doc cua luoi
       material surfacepen=currentpen, // Mau cua be mat
       pen meshpen=nullpen,
       light light=currentlight, // Huong sang
       light meshlight=light, // Huong sang cho luoi
       string name="", // Ten cho hinh
       render render=defaultrender //
);
```

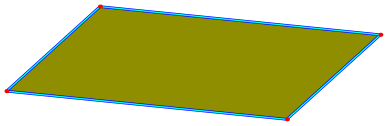
```
void draw(
       picture pic=currentpicture,
       surface s,
       int nu=1,
       int nv=1,
       material[] surfacepen, // Tao kieu, phoi mau cho be mat
       pen meshpen,
       light light=currentlight,
       light meshlight=light,
       string name="",
       render render=defaultrender
);
```

```
void draw(
       picture pic=currentpicture,
       surface s,
```

```
        int nu=1,
        int nv=1,
        material[] surfacepen,
        pen[] meshpen=nullpens, // Phoi mau cho luoi
        light light=currentlight,
        light meshlight=light,
        string name="",
        render render=defaultrender
);
```



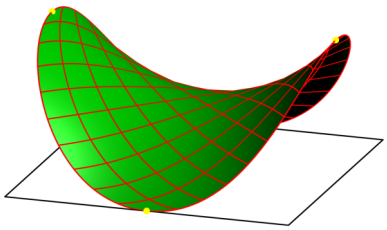
Code 58: Mặt tạo bởi path3 phẳng



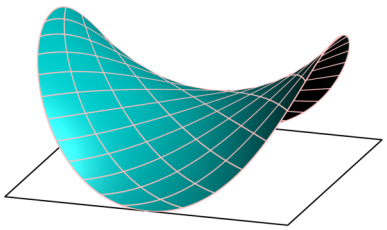
Code 59: Mặt tạo bởi path3 phẳng 2

```
import three;
currentprojection=orthographic(3,1,1);
size(5cm); dotfactor = 2;
path3 g=(-1,-1,0)--(1,-1,0)--(1,1,0)
    --(-1,1,0)--cycle;
draw(
        surface(g), // Be mat tao boi path3 kin
        nu=4,
        nv=6,
        surfacepen=yellow, // Mau cua be mat
        meshpen=blue, // Mau cua luoi
        name="Hvuong" // Ten cua be mat
);
draw(g,cyan); //Ve path3 g mau cyan
dot(g,red); // Ve 4 dinh cua g
```

```
import three;
currentprojection=orthographic(3,1,1);
size(5cm); dotfactor = 3;
path3 g=(-1,-1,0)--(1,-1,0)--(1,1,0)
    --(-1,1,0)--cycle;
draw(
        surface(g), // Be mat tao boi path3 kin
        nu=6, // So duong cua luoi
        nv=9,
        yellow, // Mau cua be mat
        blue + thick(), //co the thay thick()
            bang thin(), linewidth(0.6pt)
        name="Hvuong" // Ten cua be mat
);
draw(g,cyan); dot(g,red); // Ve 4 dinh cua g
```



Code 60: Bề mặt cong



Code 61: Bề mặt cong 2

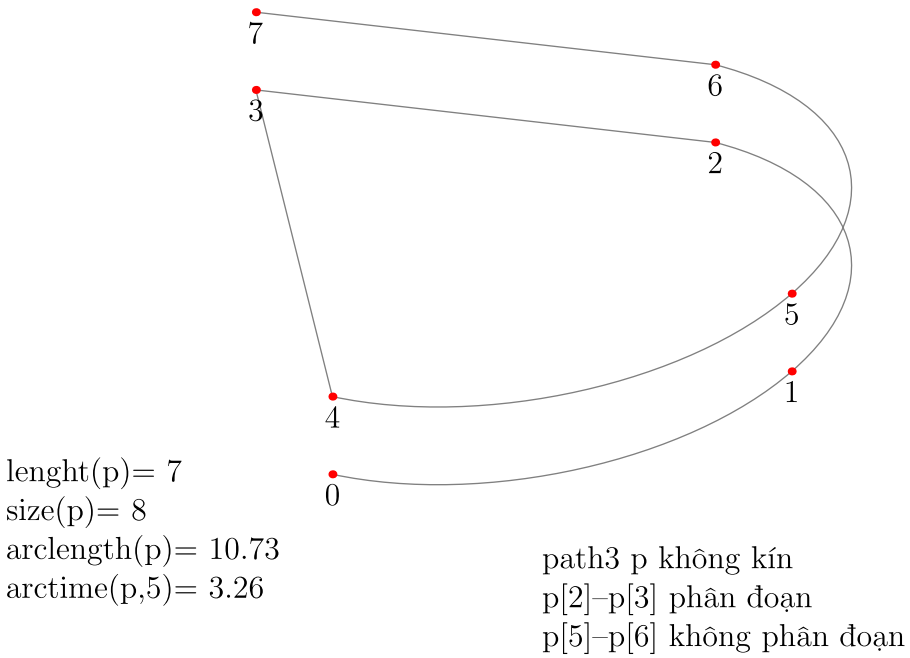
```
import three;
currentprojection=orthographic(3,1,1);
size(5cm); dotfactor =5;
//-- path3 cong noi cac diem
path3 p=(1,0,0)..(0,1,1)..(-1,0,0)..(0,-1,1)
    ..cycle;
draw(surface(p),nu=10, nv=10, green, red+
    thick());
dot(p,yellow); // Cac diem cua p
//---// Hinh vuong don vi tam O
path3 g =scale3(sqrt(2))*path3(polygon(4));
draw(g);
```

```
import three;
currentprojection=orthographic(3,1,1);
size(5cm);
pen[] pSur={cyan,magenta}, // Mau cua mat
 pMesh={palered, red, brown, darkbrown}; //
     Mau luoi
//-- Path3 cong noi cac diem
path3 g=(1,0,0)..(0,1,1)..(-1,0,0)..(0,-1,1)
    ..cycle;
draw(surface(g),nu=8, nv=12, pSur,pMesh+thick
    ());
draw(scale3(sqrt(2))*path3(polygon(4)));
```

3.12.2. Đối với path3

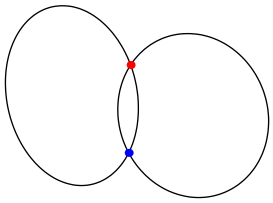
Mando cần `import math;` gói hỗ trợ tính toán để sử dụng các hàm sau.

1) `int length(path3 p);` Số điểm của `path3 p`.
2) `int size(path3 p);`
3) `triple point(path3 p, int t);` Điểm trên `path3 p` theo tỷ số t.
4) `triple point(path3 p, real t);` Điểm trên `path3 p` theo tỷ số t.
5) `triple dir(path3 p, int t, int sign=0, bool normalize=true);`
6) `triple dir(path3 p, real t, bool normalize=true);`
7) `triple accel(path3 p, int t, int sign=0);` If $sign < 0$, return the acceleration of the incoming path p at node t; if $sign > 0$, return the acceleration of the outgoing path. If sign=0, the mean of these two accelerations is returned. returns the acceleration of the path p at the point t.
8) `triple accel(path3 p, real t);`
9) `real radius(path3 p, real t);` returns the radius of curvature of the path p at the point t.
10) `triple precontrol(path3 p, int t);` returns the precontrol point of p at node t.
11) `triple precontrol(path3 p, real t);` returns the precontrol point of p at node t.
12) `triple postcontrol(path3 p, int t);` returns the postcontrol point of p at node t.
13) `triple postcontrol(path3 p, real t);` returns the postcontrol point of p at node t.
14) `real arclength(path3 p);` returns the length (in user coordinates) of the piecewise linear or cubic curve that path p represents.
15) `real arctime(path3 p, real L);` returns the path "time", a real number between 0 and the length of the path in the sense of point(path p, real t), at which the
16) `path reverse(path3 p);` returns a path running backwards along p.
17) `path subpath(path3 p, int a, int b);` returns the subpath of p running from node a to node b. If $a < b$, the direction of the subpath is reversed.
18) `path subpath(path3 p, real a, real b);`
19) `triple min(path3 p);`
20) `triple max(path3 p);`
21) `bool cyclic(path3 p);` // Hàm bool về tính khép kín của `path3 p`.
22) `bool straight(path3 p, int i);` Hàm bool về tính thẳng hàng của `path3 p`.
23) `real[] intersect(path3 p, path3 q, real fuzz=-1);` Mảng số thực chứa các chỉ số giao điểm của `path3 p` và `path3 g`.
24) `triple intersectionpoint(path3 p, path3 q, real fuzz=-1);`
25) `real[][] intersections(path3 p, path3 q, real fuzz=-1);`
26) `triple[] intersectionpoints(path3 p, path3 q, real fuzz=-1);` Giao điểm của 2 path3.
27) `real intersect(triple P, triple Q, triple n, triple Z);`
28) `triple intersectionpoint(triple n0, triple P0, triple n1, triple P1);`
29) `triple[] intersectionpoints(path3 p, surface s, real fuzz=-1);` Giao diểm của path3 với bề mặt.



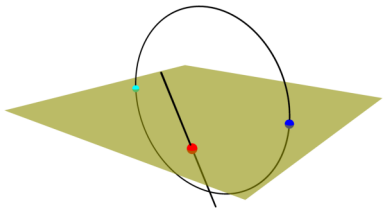
Code 62: *Giải thích một số hàm thường dùng*

```
import three;
texpreamble("\usepackage[utf8]{vietnam}");
currentprojection = orthographic(.7,0.5,0.3);
size(12cm);
//------ 4 diem dau
triple[] z;
        z[0]=(1,0,0); z[1]=(0,1,0.25); z[2]=(-1,0,0.5); z[3]=(0,-1,0.75);
//------ 4 diem sau
for(int n=4; n <= 7; ++n) z[n]=z[n-4]+0.25*Z;
//---- path3 noi cac diem tren
path3 p=z[0]..z[1]..z[2]--z[3]--z[4]..z[5]..z[6]--z[7];
draw(p,grey); dot(z,red); //
//---- Nhan cac diem
for(int n=0; n<=7; ++n) label(format("$%i$",n),z[n],S);
real h=-0.125;
//---- Nhan cac ham
label(format("length(p)= %i",length(p)), (1,-1.25,2*h),E);
label(format("size(p)= %i",size(p)), (1,-1.25,3*h),E);
label(format("arclength(p)= %.2f",arclength(p)), (1,-1.25,4*h),E);
label(format("arctime(p,5)= %.2f",arctime(p,5)), (1,-1.25,5*h),E);
//-- Neu p kin thi gan nhan path3 p kin va nguoc lai
if(cyclic(p)==true)
label("path3 p kin",(1,0.75,h),E);
else
label("path3 p khong kin",(1,0.75,h),E);
//----!!!!!!!!!!!!!!!!!!!
void segment(path3 p, int i, int n) {
if(straight(p,i)==true)
label(format("p[%i]",i)+format("--p[%i] phan doan",i+1),(1,0.75,n*h),E);
else
label(format("p[%i]",i)+format("--p[%i] khong phan doan",i+1),(1,0.75,n*h),E);
}
segment(p,2,2); segment(p,5,3);
```



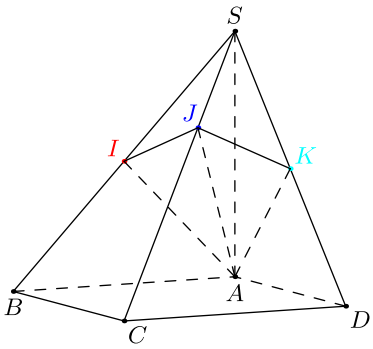
Code 63: *giao điểm của 2 path3*

```
import three;
size(3.5cm);
//-------------
path3 DtronA =circle((2Y+Z),1.5,normal=X),
          DtronB =circle((-Y+Z),2,normal=X);
draw(DtronA^^DtronB);
//----- Giao diem cua hai duong tron
triple[] I = intersectionpoints(DtronA,DtronB
    );
dot(I[0],red); // diem mau red
dot(I[1],blue); // diem mau blue
```



Code 64: *Giao điểm của path3 và surface*

```
import three; size(5cm);
//----- path3 hinh vuong
path3 Hvuong=scale3(3)*path3(polygon(4));
draw(surface(Hvuong),yellow+opacity(.6));
//------ path3 Duong tron, duong thang
path3 Dtron =circle(O,1.5,normal=X),
          Dthang = (1,1,-1)--(2,1,1);
draw(Dtron^^Dthang);
//----- Giao diem cua duong thang va mat
triple[] I = intersectionpoints(Dthang,
    surface(Hvuong));
dot(I,red); // Mau do
//----- Giao diem cua duong tron va mat
triple[] J = intersectionpoints(Dtron,surface
    (Hvuong));
dot(J[0],blue); dot(J[1],cyan);
```



Code 65: *Hình chóp tứ giác*

```
import three; import graph3; import math;
size(5cm); dotfactor =3.5;
defaultpen(fontsize(9pt));
currentprojection = orthographic(1,2,0.3);
//-------- Cac diem
triple A, B, C, D, S;
       A = O; B = X; C = X+Y; D = Y; S=Z;
//-------- Nhan diem
dot("$A$",A,dir(-90)); dot("$B$",B,dir(-90));
dot("$C$",C,SE); dot("$D$",D,SE); dot("$S$",S,N);
//-------- Cac doan thang
draw(B--C--D);
draw(S--B^^S--C^^S--D);
draw(S--A^^B--A--D,dashed);
//-- ty so giao diem cua SB voi mat phang qua A,vtpt SC
real kI = intersect(S,B,(C-S),A);
triple I=(1-kI)*S+kI*B; // Giao diem I, SB=kI*SI
```

```
dot("$I$",I,NW,red); draw(A--I,dashed);
//-- ty so giao diem cua SC voi mat phang qua A,vtpt SC
real kJ = intersect(S,C,(C-S),A);
triple J=(1-kJ)*S+kJ*C; // Giao diem J, SC=kJ*SJ
dot("$J$",J,dir(120),blue); draw(A--J,dashed);
//-- ty so giao diem cua SD voi mat phang qua A,vtpt SC
real kK=intersect(S,D,(C-S),A);
triple K=(1-kK)*S+kK*D; // Giao diem K, SD=kK*SK
dot("$K$",K,NE,cyan); draw(A--K,dashed);
draw(I--J--K); //--- Noi 3 diem I,J,K
```

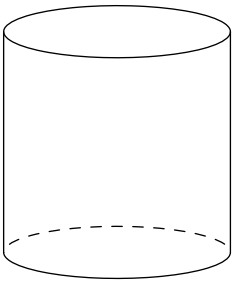
4. Gói Solids

Để sử dụng các hàm đã có trong gói `solids`, Mando thêm dòng lệnh `import solids;` ở đầu tập tin.

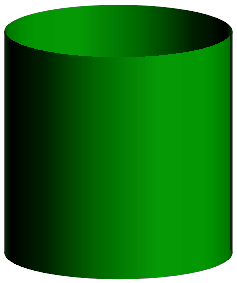
4.1. Hình trụ

Định nghĩa hình trụ

```
revolution cylinder(
        triple c = O, // Tam cua day, mac dinh la goc toa do O
        real r,  // Ban kinh day
        real h, // Chieu cao
        triple axis = Z // Huong truc, mac dinh la truc Oz
){
        triple C = c + r*perp(axis);
        axis = h*unit(axis);
        return revolution(c,C--C + axis, axis);
}
```



Code 66: Hình Trụ



Code 67: Mặt Trụ

```
settings.render=0;
import solids; // Ve mat tru chi co "xuong"
unitsize(1.5cm);
currentprojection =orthographic(.7,2,0.5);
    // Huong truc
revolution Cy = cylinder(O, 1, 2, Z);
draw(Cy); // Draw cylinder no surface
```

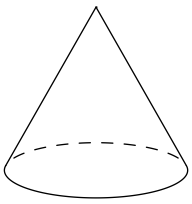
```
settings.render=0;
import solids; // Ve mat tru co "xung" va "da"
unitsize(1.5cm);
currentprojection =orthographic(.7,2,0.5);
revolution Cy = cylinder(O, 1, 2, Z);
draw(Cy); // Chi ve cac duong vien "xuong"
draw(surface(Cy), green); // Mat tru
```

4.2. Mặt nón

Định nghĩa hình nón.

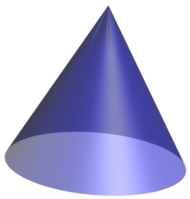
```
revolution cone(
        triple c = O, // Tam cua day
```

```
        real r, // ban kinh day
        real h,  // Chieu cao
        triple axis = Z, // Truc, mac dinh la truc Oz
        int n = nslice // So duong vi tuyen, mac dinh la 6
){
        axis = unit(axis);
        return revolution(c, approach(c + r*perp(axis)--c + h*axis, n), axis);
}
```



Code 68: Hình nón

```
settings.render=0;
import solids; // Ve mat non chi co "xuong"
unitsize(1cm);
revolution Co = cone(O, 1, 1.7, n = 1);
draw(Co);
```



Code 69: Mặt nón

```
settings.render=12;
import solids; unitsize(1.2cm);
revolution Co=cone(O,1,1.7,axis=0.2*Y+2*Z,n
    =1);
draw(surface(Co),blue+opacity(0.5));
```

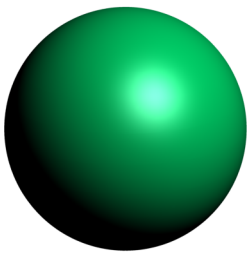
4.3. Mặt cầu

Định nghĩa mặt cầu

```
revolution sphere(
        triple c = O, // Tam mat cau
        real r, // Ban kinh
        int n = nslice // so lat cat cua mat cau, mac dinh la 14
){
        return revolution(c, Arc(c, r, 180, 0, 0, 0, Y, N),Z)
}
```

Code 70: Mặt cầu

```
settings.render=12
import solids;
unitsize(1cm);
currentprojection=orthographic(5,4,2);
real R=2; // ban kinh
revolution s=sphere(O,R);
draw(surface(s),green+cyan); //
```



4.4. Mặt tròn xoay

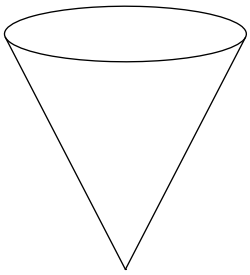
Mặt tròn xoay được tạo bởi một path3 (đường cong, đường thẳng hoặc đồ thị hàm số trên một đoạn) khi quanh tâm – điểm và vuông góc với trục từ góc ang1 đến góc ang2. Ở đây chỉ có khung "xương", không có bề mặt "da".

```
revolution(
        triple c, // Tam quay
        path3 p, // path3 p
        triple axis, // Truc quay, thuong dung: Z -- truc Oz
```

```
        real ang1, //Goc 1
        real ang2 //Goc 2
)
```

Code 71: Hình nón và mặt nón

```
settings.render=0;
import solids;
unitsize(1.4cm);
currentprojection = orthographic(.7,2,0.5);
path3 l = O--(0,1,2); // doan noi O voi diem (0,1,2)
revolution Co=revolution(O,l,axis = Z,0,360);
draw(Co); // Ve hinh non
path3 ll = O --(0,0.75,1.5);
revolution Coo=revolution(O,ll,axis = Z,0,360);
surface s = surface(Coo); // Tao mat non
draw(s,cyan+opacity(0.5)); // Ve mat non.
```



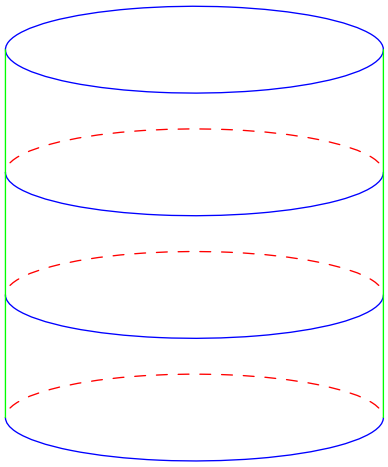
4.5. Lệnh draw thường dùng với gói solids

Các cú pháp này thường dùng để vẽ hình ("xương") của khối tròn xoay revolution.

```
void draw(
        picture pic=currentpictur, //
        revolution r, // Mat tron xoay: cone, cylinder, sphere
        int m=0, // So vi tuyen
        int n=nslice, // So lat cat, cai nay it dung
        pen frontpen=currentpen, // Mau cho khung "xuong" phia truoc
        pen backpen=frontpen, // Mau cho khung "xuong" phia sau
        pen longitudinalpen=frontpen, // Kinh tuyen truoc(tren)
        pen longitudinalbackpen=backpen, // Kinh tuyen sau(duoi)
        light light=currentlight, // Huong sang
        string name=" ", // Ten
        render render=defaultrender,
        projection P=currentprojection
)
```

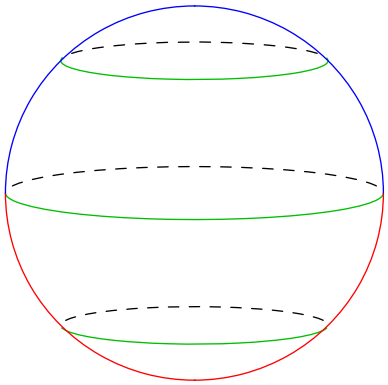
Code 72: Hình trụ – các kinh, vĩ tuyến

```
settings.render=0;
import solids;
currentprojection = orthographic(.7,2,0.5); // Huong truc
unitsize(1.2cm);
real R = 2.5;
real h = 2*R;
revolution r = cylinder(O, R, h);
draw(
        r, // revolution -- tru
        m=4, // 4 vi tuyen (m>=3)
        frontpen=blue, // mau vi tuyen truoc (nhin thay)
        backpen=red, // mau vi tuyen sau (khong nhin thay)
        longitudinalpen=green // mau kinh tuyen
);
```



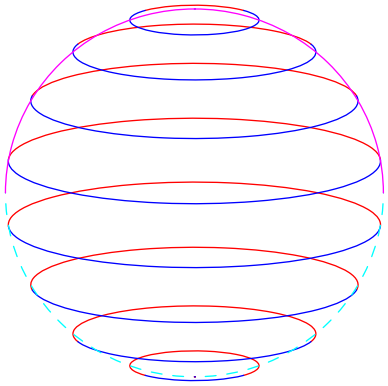
Code 73: *Hình cầu – các kinh, vĩ tuyến*

```
settings.render=0;
import solids;
currentprojection = orthographic(.7,2,0.3);
unitsize(1cm);
real R = 2.5;
revolution r = sphere(O, R);
draw( r, // revolution -- sphere: cau
       m=5, // Gom 3 vi tuyen, 1 kinh tuyen tren, 1 kinh tuyen duoi
       frontpen = heavygreen,
       backpen = black, // Muon vi tuyen sau duong lien --> +
           solid
       longitudinalpen = blue, // Kinh tuyen tren
       longitudinalbackpen = red + solid // Muon kinh tuyen duoi
            dut --> bo +solid
);
```



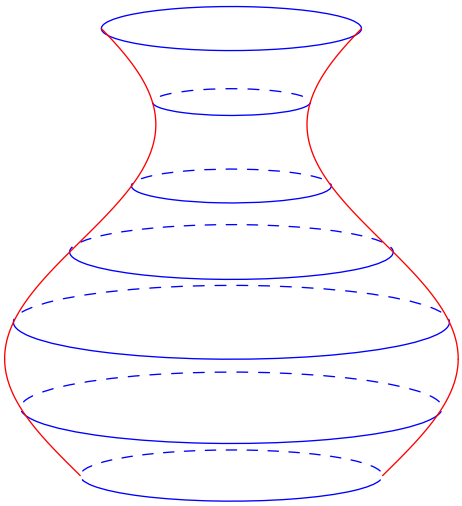
Code 74: *Hình cầu – điều khiển kinh, vĩ tuyến*

```
settings.render=0;
import solids;
currentprojection = orthographic(.7,2,0.5);
unitsize(1cm);
real R = 2.5; / Ban kinh mat cau
revolution sp = sphere(O, R);
draw( sp,
       m=10, // 8 vi tuyen, 1 kinh tuyen tren, 1 kinh tuyen duoi
       frontpen=blue, // vi tuyen truoc
       backpen=red+solid, // vi tuyen sau de che do lien net--> +
           solid
       longitudinalpen=magenta, // kinh tuyen tren
       longitudinalbackpen=cyan // Ko ve kinh tuyen duoi --> thay
            cyan boi nullpen
);
```



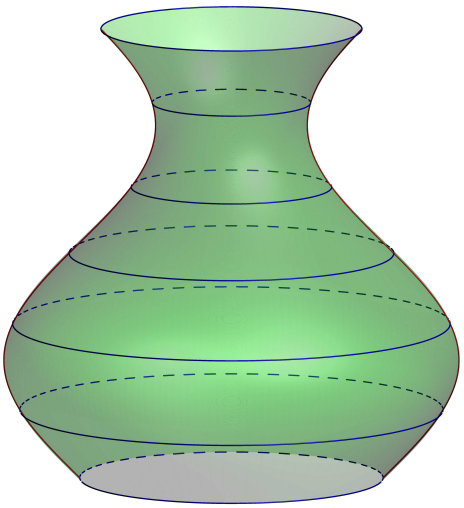
Code 75: Lọ hoa

```
settings.render=0;
import graph3; // De tao path3 sinh boi ham so f(t)
import solids;
unitsize(1cm);
currentprojection=orthographic(.7,1.6,0.3);
//====== Hai can
real a = 0, b = 6;
//====== Tao duong cong sinh boi ham so
real f(real t){return sin(t)+2;}
real x(real t){return f(t);}
real y(real t){return 0;}
real z(real t){return t;}
path3 p=graph(x,y,z,a,b,operator..);
//==== Tao revolution
revolution LoHoa = revolution(O,p,axis=Z,0,360);
draw(LoHoa,
        m = 7,
        frontpen=blue,
        longitudinalpen=red,
        longitudinalbackpen= red+solid
);
```



Code 76: Lọ hoa 2

```
settings.render=0;
import graph3; // De tao path3 sinh boi ham so f(t)
import solids;
unitsize(1cm);
currentprojection=orthographic(.7,1.6,0.3);
//====== Hai can
real a = 0; real b = 6;
real f(real t){return sin(t)+2;}
real x(real t){return f(t);}
real y(real t){return 0;}
real z(real t){return t;}
path3 p=graph(x,y,z,a,b,operator..);
//==== Tao revolution
revolution LoHoa = revolution(O,p,axis=Z,0,360);
draw(LoHoa,
        m = 7,
        frontpen=blue,
        longitudinalpen=red,
        longitudinalbackpen= red+solid
);
draw(surface(LoHoa),green+opacity(0.3)); // mat tron xoay
```



4.6. Điều khiển kinh tuyến – vĩ tuyến bằng cấu trúc skeleton

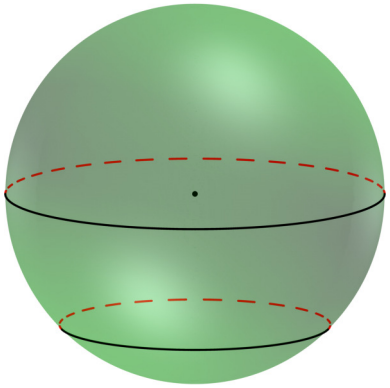
Mục đích của cấu trúc này là dùng để vẽ và điều khiển tính ẩn hiện (đứt – liền) đối với đường kinh tuyến và vĩ tuyến của mặt tròn xoay.

```
struct skeleton{
       struct curve{
              path3[] front;
              path3[] back;
```

```
        }
        //transverse skeleton: Vuong goc voi truc cua revolution
        curve transverse;
        //longitudinal skeleton: Song song voi truc cua revolution
        curve longitudinal;
}
```

Code 77: Mặt cầu – các vĩ tuyến.

```
settings.render=12;
import solids; unitsize(3cm);
currentprojection = orthographic(.7,2,0.4);
currentlight=White; //Mau anh sang
dotfactor*=0.7; // Do day cua diem
real R=1; // ban kinh mat cau
revolution s=sphere(O,R);
draw(surface(s),green+opacity(0.3)); // Ve mat cau
dot(O); //Goc toa do Oxyz
skeleton sk; // Tao skeleton sk
//== Hai vi tuyen: Xich dao, 1 vi tuyen
s.transverse(sk,reltime(s.g,acos(0)/pi),currentprojection);
s.transverse(sk,reltime(s.g,acos(0.7)/pi),currentprojection);
//== kieu va mau cua vi tuyen sau
draw(sk.transverse.back,linetype("8 8")+0.8bp+red);
//== kieu va mau cua vi tuyen truoc
draw(sk.transverse.front,0.8bp+black);
```

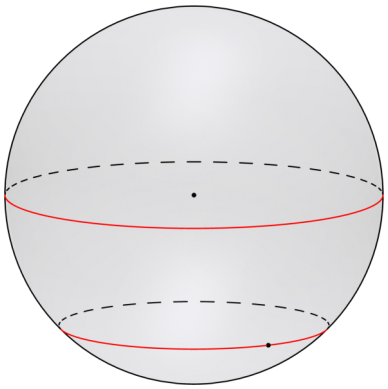


Cách lấy vĩ tuyến, điểm thuộc vĩ tuyến

- `s.transverse(sk,reltime(s.g,acos(0)/pi),currentprojection);` Lấy vĩ tuyến có tâm là đường tròn (xích đạo) có bán kính bằng bán kính của mặt cầu s.
- `s.transverse(sk,reltime(s.g,acos(0.7)/pi),currentprojection);` Lấy vĩ tuyến (ở phía dưới) có khoảng cách từ tâm của nó đến tâm mặt cầu s là 0.7R. Nếu là -0.7 thì vĩ tuyến đó ở phía trên.
- `s.transverse(sk,reltime(s.g,acos(-k)/pi),currentprojection);` Lấy vĩ tuyến có khoảng cách từ tâm của nó đế tâm mặt cầu s bằng $|k|$ với vĩ tuyến ở phía trên nếu $k > 0$, lấy vĩ tuyến phía dưới nếu $k \leq 0$. Lấy k thuộc $[-1; 1]$.
- Nếu đã chọn vĩ tuyến `s.transverse(sk,reltime(s.g,acos(-k)/pi),currentprojection);` thì lệnh lấy điểm thuộc vĩ tuyến đó là `triple M = R*dir(90+aSin(-k),GocQuay);`

Code 78: Mặt cầu, điểm thuộc vĩ tuyến

```
settings.render=0;
import solids; size(5cm);
currentprojection=orthographic(10,5,2);
currentlight=White; dotfactor=3.5; real R=3;
revolution s=sphere(O,R);
draw(surface(s),0.9white+opacity(0.1));
draw(s.silhouette(),black); // Ve duong tron vien cua mat cau
dot(O); skeleton sk; //
//== vi tuyen xich dao
s.transverse(sk,reltime(s.g,acos(0)/pi),currentprojection);
real k = -0.7; // Vi tuyen -0.7 -- phia duoi
s.transverse(sk,reltime(s.g,acos(-k)/pi),currentprojection);
//== Ve hai vi tuyen
draw(sk.transverse.back,linetype("10 10"));
draw(sk.transverse.front,red);
//== Lay diem M tren vi tuyen -0.7
triple M = R*dir(90+aSin(-k),60); dot(M);
```

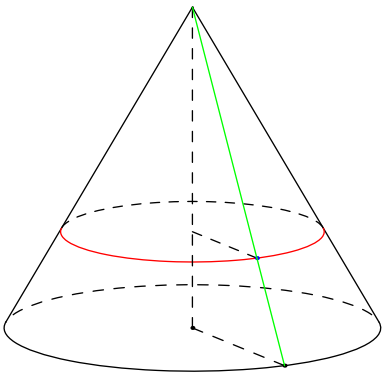


Chú ý.

- `s.silhouette()`: Đường tròn xung quanh của mặt cầu dùng để hiển thị rõ mặt cầu được vẽ trong mặt phẳng.
- `R*dir(90+aSin(0.7),60)`: Lấy điểm thuộc vĩ tuyến -0.7 ở vị trí quét từ tia Oz xuống một góc `90+aSin(0.7)` độ và quét một cung lượng giác vuông góc với trục Oz góc 60°.

Code 79: Hình nón, điểm thuộc vĩ tuyến

```
settings.render=0;
import solids; size(6cm);
currentprojection = orthographic(.7,2,0.5);
dotfactor =3;
real R=2; // Ban kinh day
real h=3.5; // Chieu cao
revolution Co = cone(O,R,h,axis=Z,n=1); // Hinh non
draw(Co);
real r= 0.7; // Ban kinh vi tuyen bang 0.7 bk day. r tu 0--> 1
skeleton sk;
Co.transverse(sk,reltime(Co.g,1-r),currentprojection); // Vi
    tuyen
draw(sk.transverse.front,red); // Mau do vi tuyen truoc
draw(sk.transverse.back,linetype("7 7")); // Vi tuyen sau
//=== Diem nam tren vi tuyen
real d = sqrt(R^2*(r)^2+(1-r)^2*h^2); // khoang cach tu O toi vi
    tuyen
real Goc = aSin(R*(r)/d); // Goc quet tu truc Oz xuong (do)
triple M = d*dir(Goc,100); // Diem M vi tri (Goc,100)
draw((0,0,M.z)--M,dashed); dot(O); dot(M,blue);
//====
draw(h*Z--O,dashed); draw(h*Z--M,green);
//=== Lay diem N tren duong tron day
triple N = R*dir(90,100);
dot(N); draw(O--N,dashed); draw(N--M,green);
```



Chú ý.

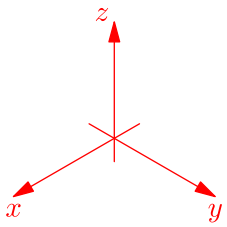
- `d = sqrt(R^2*(r)^2+(1-r)^2*h^2)`: Khoảng cách từ tâm O của hình nón (bk đáy R và chiều cao h) đến 1 điểm bất kỳ thuộc vĩ tuyến tỷ số r.
- `Goc = aSin(R*r/d)`: Góc quét từ trục Oz xuống mặt phẳng Oxy theo đơn vị độ, mục đích của góc quét này là dùng để lấy 1 điểm trên vĩ tuyến.
- `d*dir(Goc,100)`: Lấy điểm thuộc vĩ tuyến tỷ số k, hướng 100 độ.

5. Gói graph3

5.1. Hệ trục tọa độ $Oxyz$

Trong gói graph3 có một số hàm định nghĩa hệ trục $Oxyz$ trong không gian. Để nạp gói graph3 Mando thêm dòng lệnh `import graph3;` ở đầu tập tin asy.

```
void axes3(
       picture pic=currentpicture,
       Label xlabel="x", // Ky hieu x o dau truc x
       Label ylabel="y", // ky hieu y o dau truc Oy
       Label zlabel="z", // ky hieu z o dau truc Oz
       triple min=(Xmin,Ymin,Zmin), // Do dai truc tu min toi max
       triple max=(Xmax,Ymax,Zmax), //
       pen p=currentpen, // Mau truc
       arrowbar3 arrow=None // Mui ten mac dinh khong co
);
```



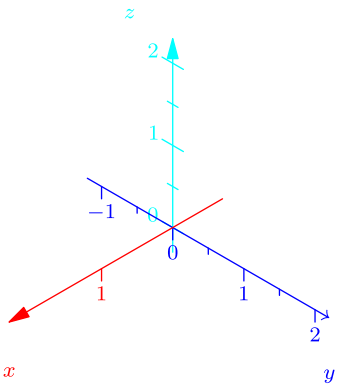
***Code** 80: Hệ trục đồng bộ 1 màu*

```
import graph3;
size(3cm);
currentprojection=orthographic(1,1,1);
axes3("$x$", "$y$", "$z$",
       (-0.5,-0.5,-0.4), //-- min
       (2,2,2), // -- max
       red, // --color
       Arrow3()
);
```

```
void xaxis3(
       picture pic=currentpicture,
       Label L="x", // Nhan truc Ox
       axis axis=YZZero,
       real xmin=-infinity, // xmin
       real xmax=infinity, // xmax
       pen p=currentpen, // color
       ticks3 ticks=NoTicks3,// gom co NoTicks3,InTicks,OutTicks, InOutTicks
       arrowbar3 arrow=None, // mui ten, mac dinh None
       bool above=false //
); // Tuong tu co truc Oy, Oz
```

Code 81: Hệ trục

```
import graph3;
size(8cm);
defaultpen(fontsize(8pt));
currentprojection=orthographic(1,1,1);
xaxis3("$x$",-0.7,2.3,red,OutTicks(),Arrow3());
yaxis3("$y$",-1.2,2.2,blue,OutTicks(),Arrow3(TeXHead3));
zaxis3("$z$",-0.3,2.3,cyan,InOutTicks(),Arrow3());
```

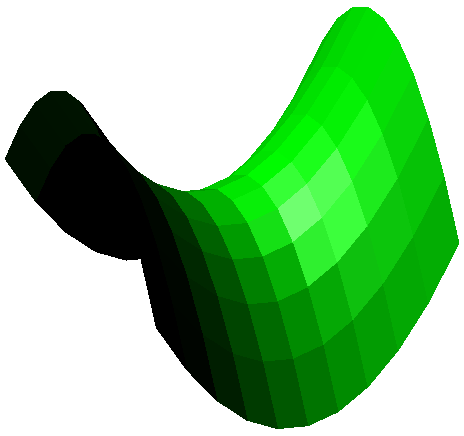


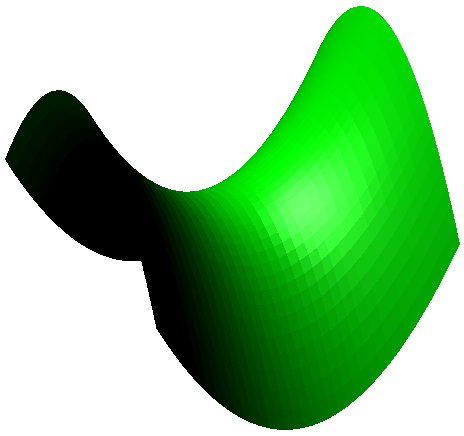
5.2. Đường – curve và mặt – surface

5.2.1. Mặt xác định bởi phương trình $z = f(x, y)$

```
surface surface(
        real f(pair z), // Ham hai bien z.x, z.y
        pair a, // a.x = xmin, a.y=ymin
        pair b, // b.x = xmax, b.y=ymax
        int nx=nmesh, // so o luoi tren mat
        int ny=nx, //
        bool cond(pair z)=null
);
```

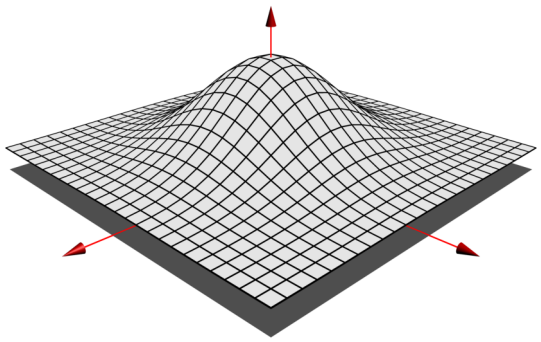
```
surface surface(
        real f(pair z),
        pair a,
        pair b,
        int nx=nmesh,
        int ny=nx,
        splinetype xsplinetype, // Do tron cua cac o luoi: Spline, notaknot, natural, periodic,
            clamped(real slopea, real slopeb)
        splinetype ysplinetype=xsplinetype,
        bool cond(pair z)=null
);
```





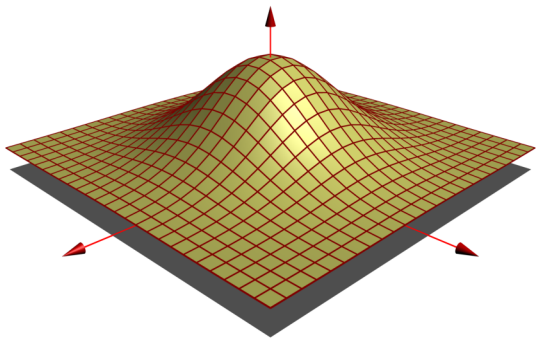
Code 82: Mặt yên ngựa – ít "mịn"

```
import graph3;
currentprojection=orthographic(2,4,3);
currentlight=(2,0,3);
size(6cm);
real f(pair z) {return z.x^2-z.y^2;}
surface s=surface(f, (-1,-1), (1,1), nx=10);
draw(s,green); // Ve mat yen ngua
```



Code 83: Mặt yên ngựa – rất "mịn"

```
import graph3;
currentprojection=orthographic(2,4,3);
currentlight=(2,0,3);
size(6cm);
real f(pair z) {return z.x^2-z.y^2;}
surface s=surface(f, (-1,-1), (1,1), nx=35);
draw(s,green); // Ve mat yen ngua
```



Code 84: Mặt không có hướng sáng

```
import graph3;
currentprojection=perspective(2,2,2);
size(7cm);
real a=2;
real f(pair z) {return exp(-abs(z)^2)+0.25;}
surface Mcong=surface(f, (-a,-a), (a,a), nx
    =25);
path3 Hvuong=plane(2*(a,0,0), 2*(0,a,0), (-a
    ,-a,0));
draw(surface(Hvuong), gray);
draw(Mcong,lightgray, meshpen=black+thick(),
    nolight);
xaxis3("",0,a+0.5,red,Arrow3); // 3 truc
yaxis3("",0,a+0.5,red,Arrow3);
zaxis3("",0,1.75,red,Arrow3);
```

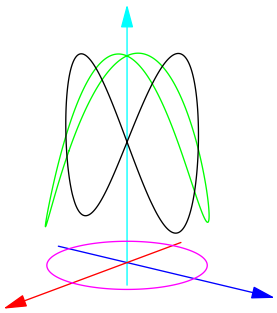
Code 85: Măt có hướng sáng

```
import graph3;
currentprojection=perspective(2,2,2);
size(7cm);
real a=2;
real f(pair z) {return exp(-abs(z)^2)+0.25;}
surface Mcong=surface(f, (-a,-a), (a,a), nx
    =25);
path3 Hvuong=plane(2*(a,0,0), 2*(0,a,0), (-a
    ,-a,0));
draw(surface(Hvuong), gray);
draw(Mcong,lightyellow, meshpen=brown+thick())
    ;
xaxis3("",0,a+0.5,red,Arrow3); // 3 truc
yaxis3("",0,a+0.5,red,Arrow3);
zaxis3("",0,1.75,red,Arrow3);
```

5.2.2. Đường – đường cong tham số

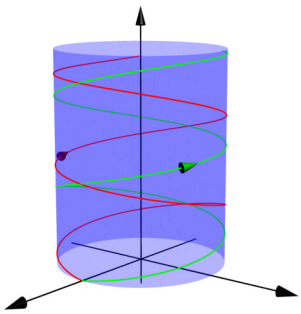
Đường – đường cong tham số, Mando có thể định nghĩa nó bởi `path3` như sau

```
real x(real t){return x(t);} // Phuong trinh hoanh do x theo bien t
real y(real t){return y(t);} // Phuong trinh tung do y theo bien t
real z(real t){return z(t);} // Phuong trinh cao do z theo bien t
path3 Cur = graph(x,y,z,a,b,operator..); // Duong tham so tu a->b, Neu la duong cong thi them
    operator.. de duong cong min hon.
```



Code 86: *Đường cong* $(\cos t, \sin t, \sin 2t + 1.5)$

```
import graph3;size(4cm);
//---------- truc toa do
xaxis3("",-1.2,2,red,Arrow3());
yaxis3("",-1.2,2,blue,Arrow3());
zaxis3("",-0.3,3,cyan,Arrow3());
//----- Duong cong Cur -- green
real x(real t){return cos(t);}
real y(real t){return sin(t);}
real z(real t){return sin(2*t)+1.5;}
path3 Cur = graph(x,y,z,0,2*pi,operator..);
draw(Cur,green);
//----- Chieu Cur len (Oxy)
transform3 Poxy=planeproject(XY*unitsquare3);
draw(Poxy*Cur,blue+red);
//----- Chieu Cur len (Oyz)
transform3 Poyz=planeproject(YZ*unitsquare3);
draw(Poyz*Cur);
```



Code 87: *Đường cong* $(\cos 6t, \sin 6t, t)$

```
import graph3;size(4cm);
xaxis3("",-1.2,2,Arrow3());
yaxis3("",-1.2,2,Arrow3());
zaxis3("",-0.3,3,Arrow3());
//----- Duong cong Cur -- green
real x(real t){return cos(6*t);}
real y(real t){return sin(6*t);}
real yy(real t){return -sin(6*t);}
real z(real t){return t;}
real a=0, b=2.5;
path3 Cur = graph(x,y,z,a,b,operator..);
draw(Cur,green,MidArrow3());
path3 Cury = graph(x,yy,z,a,b,operator..);
draw(Cury,red,MidArrow3());
//---------- Mat tru co chieu cao b
draw(scale(1,1,b)*unitcylinder,blue+opacity
    (.3),nolight);
```

6. Macros

Philippe Ivaldi đã tạo ra một số macros đánh dấu góc vuông, góc không vuông, khoảng cách để bổ sung thêm cho việc hoàn thiện một hình vẽ.

Để sử dụng các macros này, Mando coppy các macros vào một tập tin có tên macros3D.asy và lưu vào thư mục cài đặt asymptote.

6.1. Macro đánh dấu góc vuông

```
void drawrightangle(
       picture pic=currentpicture,
       triple M, // Danh dau goc vuong tai M voi hai canh goc vuong MA,MB
       triple A, //
       triple B, //
       real radius=0, // do dai canh goc vuong, dung do vi mm
       pen p=currentpen, // Mau goc vuong
       pen fillpen=nullpen, // To mau goc vuong
       projection P=currentprojection
){
       p=linejoin(0)+linecap(0)+p;
       if (radius==0) radius=arrowfactor*sqrt(2);
       transform3 T=shift(-M);
       triple OA=radius/sqrt(2)*unit(T*A),
       OB=radius/sqrt(2)*unit(T*B),
```

```
        OC=OA+OB;
        path3 _p=OA--OC--OB;
        picture pic_;
        draw(pic_, _p, p=p);
        if (fillpen!=nullpen) draw(pic_, surface(O--_p--cycle), fillpen);
        add(pic,pic_,M);
}
```

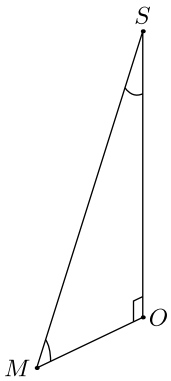
6.2. Macro đánh dấu góc không vuông

```
void markangle3D(
        picture pic=currentpicture,
        Label L="", // nhan cho goc
        triple M, // Danh dau goc tai M, hai tia MA, MB
        triple A,
        triple B,
        bool cc=true,
        real radius=0, // Ban kinh cung, do vi mm
        pen p=currentpen, // Mau cung
        pen fillpen=nullpen, // To mau
        arrowbar3 arrow=None, // Mui ten
        projection P=currentprojection
){
        p=linejoin(0)+linecap(0)+p;
        if (radius==0) radius=arrowfactor*sqrt(2);
        transform3 T=shift(-M);
        triple OA=radius/sqrt(2)*unit(T*A),
        OB=radius/sqrt(2)*unit(T*B);
        path3 pl=O--OA--OB;
        triple V=normal(pl);
        path3 _p; real k;
        if (cc) k=1;
        else k=-1;
        picture pic_;
        _p=arc(O,OA,OB,k*V);
        if (fillpen!=nullpen) draw(pic_, surface(O--_p--cycle), fillpen);
        draw(pic_, L, _p, p=p, arrow);
        add(pic,pic_,M);
}
```

6.3. Macro khoảng cách

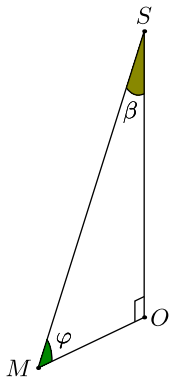
```
void cote3D(
        picture pic=currentpicture,
        Label L="", // Nhan
        triple A, // Hai dau mut A va B
        triple B,
        real d=5mm, // Khoang cach tu cac diem den ky hieu
        triple v, //
        bool cc=true,
        pen p=currentpen, // Mau
        pen joinpen=dotted, // kieu noi
        arrowbar3 arrow=Arrows3 // Kieu mui ten
){
        transform3 T=shift(d*unit(v));
```

```
        triple A=A, B=B;
        pic.add(new void(picture f, transform3 t) {
        picture opic;
        path3 dist;
        triple Ap=t*A, Bp=t*B;
        triple a=T*Ap, b=T*Bp;
        if (cc) {dist=a--b;}
        else {dist=b--a;}
        draw(opic,L,dist,p,arrow);
        draw(opic,a--Ap^^b--Bp,joinpen);
        add(f,opic);
        }, true);
}
```



Code 88: *Đánh dấu góc*

```
settings.render=0;
import macros3D; // Co chua modul three
size(5cm); dotfactor =3;
defaultpen(fontsize(9pt));
currentprojection = orthographic(.7,2,0.5);
//-- Cac diem
triple S = 1.7*Z, M = X+Y;
path3 p = S--O--M--cycle;
//-- Danh dau goc vuong SOM
drawrightangle(O,M,S,radius=4mm);
//-- Danh dau goc SMO
markangle3D(M,S,O,radius=6mm);
//-- Danh dau goc MSO
markangle3D(S,M,O,radius=12mm);
draw(p); // Tam giac SOM
dot("$O$",O,E); dot("$M$",M,W);
dot("$S$",S,N);
```

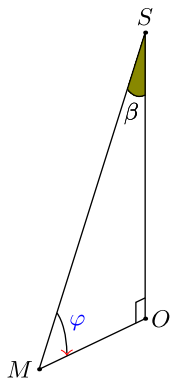


Code 89: *Đánh dấu, tô, ký hiệu góc*

```
settings.render=0; import macros3D;//
size(5cm); dotfactor =3;
defaultpen(fontsize(9pt));
currentprojection = orthographic(.7,2,0.5);
//-- Cac diem
triple S = 1.7*Z, M = X+Y;
path3 p = S--O--M--cycle;
//-- Danh dau goc vuong SOM
drawrightangle(O,M,S,radius = 4mm);
//-- Danh dau goc SMO
markangle3D(Label("$\varphi$",align=dir(30)),
     M,S,O,radius=6mm,fillpen=green);
//-- Danh dau goc MSO
markangle3D("$\beta$",S,M,O,radius=12mm,
    fillpen=yellow);
draw(p); dot("$O$",O,E); dot("$M$",M,W); dot
    ("$S$",S,N);
```

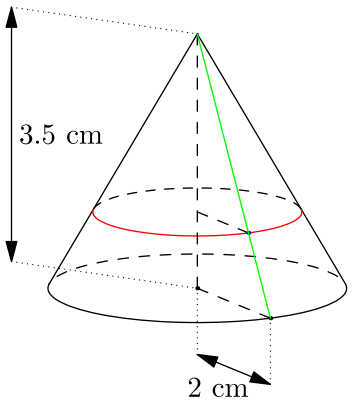
Code 90: Đánh dấu góc có mũi tên

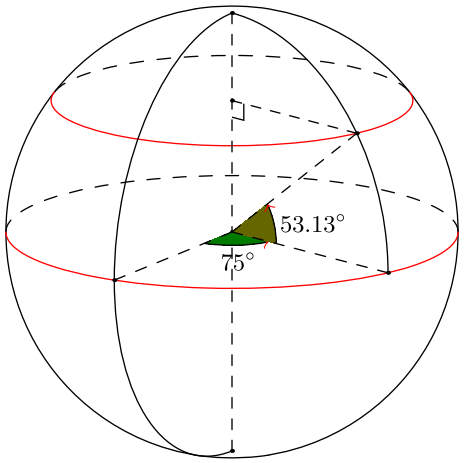
```
settings.render=0;
import macros3D; // chua modul three
DefaultHead3=TeXHead3;
size(5cm); dotfactor =3;
defaultpen(fontsize(9pt));
currentprojection = orthographic(.7,2,0.5);
//-- Cac diem
triple S = 1.7*Z, M = X+Y;
path3 p = S--O--M--cycle;
//-- Danh dau goc vuong SOM
drawrightangle(O,M,S,radius = 4mm);
//-- Danh dau goc SMO
markangle3D(Label("$\varphi$",align=dir(30),blue), M,S,O,radius
    =12mm,arrow=Arrow3(red));
//-- Danh dau goc SMO
markangle3D("$\beta$",S,M,O,radius=12mm,fillpen=yellow);
draw(p); // Tam giac SOM
dot("$O$",O,E); dot("$M$",M,W); dot("$S$",S,N);
```



Code 91: Hình nón – ký hiệu khoảng cách

```
settings.render=0;
import macros3D;
import solids; size(5cm);
currentprojection = orthographic(.7,2,0.5);
dotfactor =3;
real R=2,h=3.5; // Ban kinh day, Chieu cao
revolution Co = cone(O,R,h,axis=Z,n=1); // Hinh non
draw(Co);
real r= 0.7; // Ban kinh vi tuyen bang 0.7 bk day. r tu 0--> 1
skeleton sk;
Co.transverse(sk,reltime(Co.g,1-r),currentprojection); // Vi
    tuyen
draw(sk.transverse.front,red); // Mau do vi tuyen truoc
draw(sk.transverse.back,linetype("7 7")); // Vi tuyen sau
//=== Diem nam tren vi tuyen
real d = sqrt(R^2*(r)^2+(1-r)^2*h^2); // khoang cach tu O toi vi
    tuyen
real Goc = aSin(R*(r)/d); // Goc quet tu truc Oz xuong (do)
triple M = d*dir(Goc,100); // Diem M vi tri (Goc,100)
draw((0,0,M.z)--M,dashed); dot(O); dot(M,blue);
//====
draw(h*Z--O,dashed); draw(h*Z--M,green);
//=== Lay diem N tren duong tron day
triple N = R*dir(90,100);
dot(N); draw(O--N,dashed); draw(N--M,green);
//-------
cote3D(format("$%f$~cm",R),O,N,-Z, d=9mm);
cote3D(format("$%f$~cm",h),O,h*Z,(0.4,-0.5,0), d=2.9cm);
```

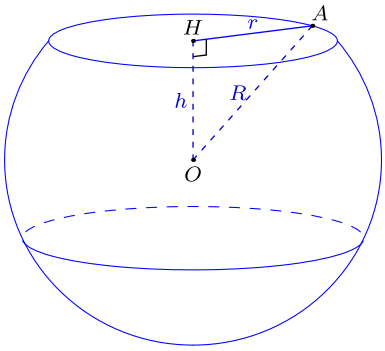




Code 92: Giả cầu

```
settings.render=0;
import macros3D;
import graph3; import solids;
size(6cm);
currentlight=White; // mau anh sang
currentprojection=orthographic(.7,1.6,0.45); // Huong truc
dotfactor =3;
DefaultHead3=TeXHead3; // Mat dinh kieu mui ten TeXHead3
defaultpen(fontsize(9pt)); // size chu 9pt
//------------Ban kinh
real R=1.2;
//------------ Diem cuc bac, cuc nam
triple Np = R*Z, Sp = -R*Z;
//-----------
revolution Sph = sphere(O,R); // Hinh cau tam O, bk R
draw(Sph.silhouette(),black); // Ve hinh tron
dot(O);
//-----------
skeleton sk; //
real k=0.6; //
Sph.transverse(sk,reltime(Sph.g,acos(0)/pi), currentprojection); // Xich dao
Sph.transverse(sk,reltime(Sph.g,acos(-k)/pi), currentprojection);// Vi tuyen ti so k
//-----------
draw(sk.transverse.back,linetype("10 10")); // kieu vi tuyen sau
draw(sk.transverse.front,red); // Kieu vi tuyen truoc
//----------
draw(Np--Sp,dashed); dot(Np^^Sp); //Duong kinh qua 2 cuc
//---------------------- Cung trai
real Qphai=35, Qdiem=110;
path3 ArcT=arc(O,R,0,Qphai,180,Qphai); // Cung trai
draw(ArcT);
//---------- Diem G, M, L
triple G = R*dir(90+aSin(0),Qphai),
       M = R*dir(90+aSin(-k),Qdiem),
       L = R*dir(90+aSin(0),Qdiem);
dot(G^^M^^L); // ve cac diem G,M,L
//------------ Danh dau goc
markangle3D(Label(format("$%f^\circ$",abs(Qdiem-Qphai)),align=S), O,G,L,radius=10mm,fillpen=
    green,arrow=Arrow3(red));
```

```
markangle3D(Label(format("$%0.2f^\circ$",abs(90-aCos(k))),align=E),O,L,M,radius=12mm,
    fillpen=yellow,
arrow=Arrow3(red));
//------------- Cung NM
path3 ArcP = arc(O,R,0,Qdiem,90,Qdiem); // Cung phai
draw(ArcP); draw(O--G^^O--M^^O--L,dashed);
triple H = M.z*Z; dot(H); draw(H--M,dashed);
```

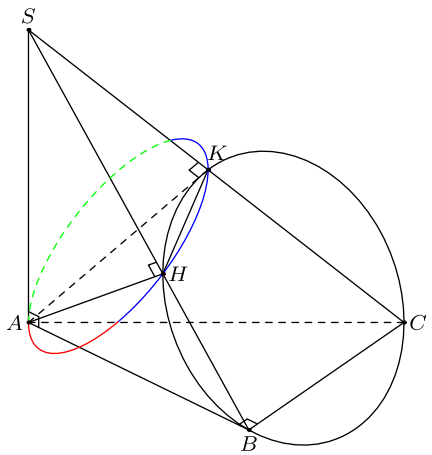


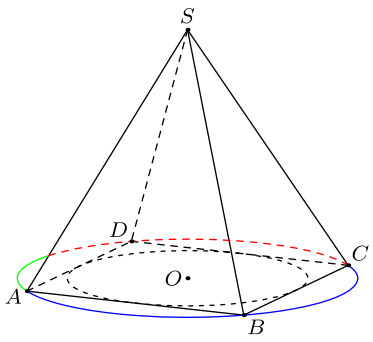
***Code** 93: Cắt mặt cầu bởi mặt phẳng*

```
settings.render=0;
import solids; import macros3D;
size(5cm);
defaultpen(fontsize(8pt)); dotfactor =3;
currentprojection=orthographic(.7,2,0.4);
// -- Ban kinh
real R=4, Goc=50;
//--------- Cac diem
triple I =-R*Z, A =R*dir(Goc,195), H=A.z*Z;
path3 ar= arc(O,A,I);
//--- mat xoay tao boi cung cung
revolution re=revolution(O,ar,axis=Z,0,360);
//------- Cat mat cau
pen pCau=blue+0.4*bp;
pen pAn=linetype("4 6")+0.8*blue+0.5*bp;
draw(re, m=3, frontpen=pCau,
       longitudinalpen=pCau,backpen=pCau,
       longitudinalbackpen=pCau+solid);
//---------
draw(Label("$r$",align=0.7*dir(90)),H--A,blue);
draw(Label("$h$",align=LeftSide),H--O,pAn);
draw(Label("$R$",align=LeftSide),A--O,pAn);
dot("$O$",O,dir(-90)); dot("$H$",H,dir(90));
dot("$A$",A,dir(60));
drawrightangle(H,O,A,radius=3mm);
```

Code 94: Ẩn – hiện trong 3D

```
settings.render=0;
import three; import math;
import solids; import macros3D;
size(5cm);
currentprojection=orthographic(.7,0,0.5);
defaultpen(fontsize(8pt));
dotfactor =3;
pen An=linetype("5 5");
//------ Diem va ca thon so
real r=2, ang=40, h=2.5;
triple S=h*Z, A=0*X,C=r/Cos(ang)*Y,
        B=(r*Sin(ang),r*Cos(ang),0);
//--- Noi cac canh
draw(S--A^^S--B^^S--C);
draw(A--C,An); draw(A--B--C);
//---- Dung hinh chieu cua A len SB
real kB = intersect(S,B,(B-S),A);
triple H = (1-kB)*S+kB*B; dot(H,red);
//---- Dung hinh chieu cua A len SC
real kC = intersect(S,C,(C-S),A);
triple K = (1-kC)*S+kC*C; dot(K,blue);
draw(A--K,An); draw(H--K); draw(A--H);
//--- Danh dau goc vuong
real bkGoc=2.5;
drawrightangle(H,A,S,radius=bkGoc*mm);
drawrightangle(K,A,S,radius=bkGoc*mm);
drawrightangle(B,C,A,radius=bkGoc*mm);
drawrightangle(A,B,S,radius=bkGoc*mm);
//---- Hai duong hai
triple I=C/2+H/2,J=A/2+K/2;
real ra=abs(C-H)/2,rb=abs(A-K)/2;
path3
       p=B--C--H--cycle,
       q=A--H--K--cycle,
       ca=circle(I,ra,normal(p)),
       cb=circle(J,rb,normal(q));
draw(ca);
//---- Tao an-hien cho duong tron tren
path
p1=cut(project(cb),project(A--S--K),1).before,
p2=cut(project(cb),project(A--S--K),2).after,
p=cut(cut(project(cb),project(A--S--K),1).after,
       project(A--S--K),1).before;
//----------------------------
draw(invert(p1,Z,O),blue);
draw(invert(p2,Z,O),red);
draw(invert(p,Z,O),An+green);
//---- Nhan diem
dot("$A$",A,dir(180));dot("$B$",B,dir(-90));
dot("$C$",C,dir(0)); dot("$S$",S,dir(90));
dot("$H$",H,dir(0)); dot("$K$",K,1.7*dir(64));
```





Code 95: Hình chóp tứ giác đều

```
settings.render=0;
import three; size(5cm);
//------------
currentprojection=orthographic(.7,2,0.5);
defaultpen(fontsize(8pt));
dotfactor = 3;
//------ ban kinh day, cac diem
real r=4,angle=30;
triple A[]; triple S=(0,0,6);
for(int i=0;i<=3;++i){
       A[i]=r*dir(90,i*90);}
//---- Duong tron ngoai, trong
path3 Co=scale3(r)*unitcircle3,
         ci=scale3(r*Sin(45))*unitcircle3;
pen Ana=linetype("3 5"), Anb=linetype("5 5");
draw(ci,Ana);
//-------------
draw(A[0]--A[1]--A[2]^^S--A[0]^^S--A[1]^^S--A[2]);
draw(S--A[3]--A[0]^^A[3]--A[2],Anb);
//-------Tao an - hien
path
p1=cut(project(Co),project(A[0]--S--A[2]),1).before,
p2=cut(project(Co),project(A[0]--S--A[2]),2).after,
p=cut(cut(project(Co),project(A[0]--S--A[2]),1).after,
       project(A[0]--S--A[2]),1).before;
//------- Ve cac duong an - hien
draw(invert(p1,Z,O),blue);
draw(invert(p2,Z,O),green);
draw(invert(p,Z,O),red+Anb);
//------- Nhan diem
dot("$A$",A[0],dir(200)); dot("$S$",S,N);
dot("$B$",A[1],SE); dot("$C$",A[2],NE);
dot("$D$",A[3],NW); dot("$O$",O,dir(180));
```