

Elipses y elipsoides: ejemplos Matlab

© 2024, Antonio Sala. Universitat Politècnica de València. Todos los derechos reservados.

Código ejecutado en Matlab R2023a

Objetivos: apoyar con código la comprensión de conceptos teóricos sobre elipses y elipsoides.

Tabla de Contenidos

Circunferencia.....	1
Elipse alineada con ejes.....	2
Elipse genérica (rotación).....	2
Diagonalización.....	3
Cortes con ejes.....	4
Radio mínimo (circ. inscrita) y máximo (circunferencia circunscrita), semiejes.....	4
Imagen del círculo unidad por matriz L.....	5
SVD de L.....	6
Inclusión.....	7
Transformación lineal (ejemplo proyección).....	8
3D elipsoide (o hiperboloide).....	9

```
syms x [2 1] real
x
```

x =

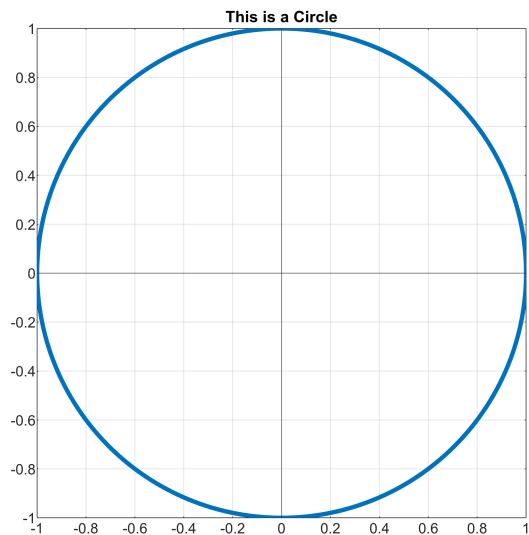
$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Circunferencia

```
M=eye(2);
x'*M*x
```

ans = $x_1^2 + x_2^2$

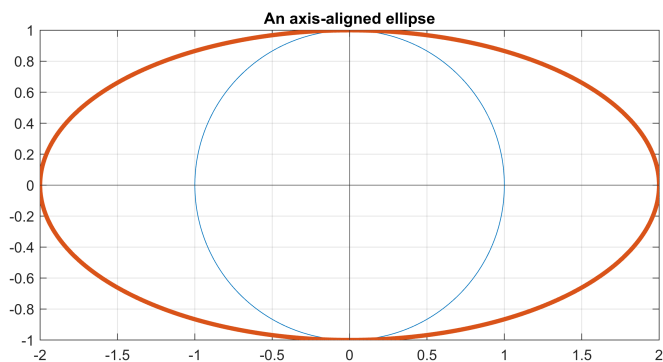
```
fimplicit(x'*M*x-1,LineWidth=3), grid on, axis equal
xline(0), yline(0), title("This is a Circle")
```



```
fimplicit(x'*M*x-1), grid on, axis equal %repeat unit circle, to compare
```

Elipse alineada con ejes

```
M=[1/2^2 0;0 1];
hold on
fimplicit(x'*M*x-1,LineWidth=3)
xline(0), yline(0)
hold off, title("An axis-aligned ellipse")
```



Elipse genérica (rotación)

```
fimplicit(x'*eye(2)*x-1,'-.'), grid on, axis equal %repeat unit sphere
M=[1/2^2 0;0 1];
hold on
```

```
fimplicit(x'*M*x-1,'-.') %aligned with axis
alpha=30*pi/180;
R=[cos(alpha) -sin(alpha); sin(alpha) cos(alpha)] %Rotate alpha radians
```

```
R = 2x2
    0.8660    -0.5000
    0.5000     0.8660
```

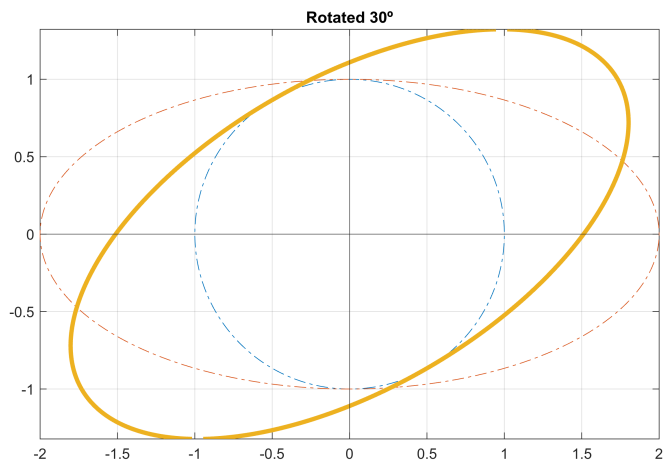
```
R'*R
```

```
ans = 2x2
     1     0
     0     1
```

```
P=R*M*R' %arbitrary positive-definite matrix
```

```
P = 2x2
    0.4375    -0.3248
   -0.3248     0.8125
```

```
fimplicit(x'*P*x-1,LineWidth=3)
hold off, title("Rotated 30°")
xline(0), yline(0)
```



Diagonalización

```
[V,D]=eig(P) %recuperamos la "rotación" y la "forma"
```

```
V = 2x2
   -0.8660    -0.5000
   -0.5000     0.8660
D = 2x2
    0.2500     0
     0     1.0000
```

```
P-V*D*V'
```

```
ans = 2x2
```

```
10-15 ×
      0      -0.1110
    -0.1110      0
```

R

```
R = 2×2
    0.8660    -0.5000
    0.5000     0.8660
```

```
atan(V(2,1)/V(1,1))*180/pi %ángulo de rotación
```

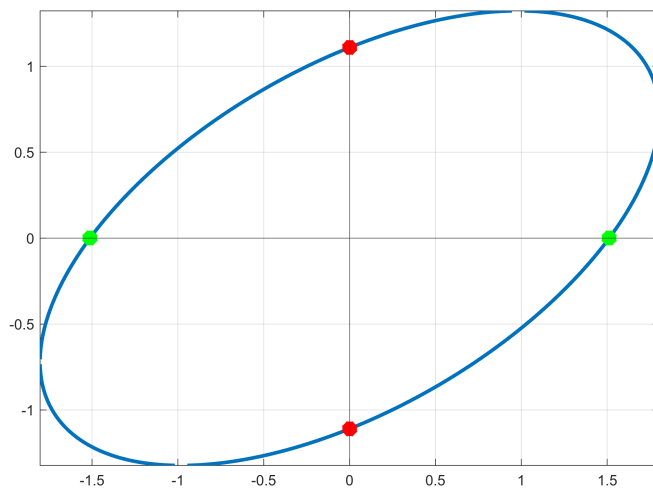
```
ans = 30.0000
```

Cortes con ejes

P

```
P = 2×2
    0.4375    -0.3248
   -0.3248     0.8125
```

```
fimplicit (x'*P*x-1,LineWidth=2.5), grid on, axis equal
xline(0), yline(0)
hold on
plot([1 -1]*1/sqrt(P(1,1)), [0 0], '*g', MarkerSize=10, LineWidth=5)
plot([0 0], [1 -1]*1/sqrt(P(2,2)), '*r', MarkerSize=10, LineWidth=5)
hold off
```



Radio mínimo (circ. inscrita) y máximo (circunferencia circunscrita), semiejes

P

```
P = 2x2
    0.4375    -0.3248
   -0.3248     0.8125
```

```
eig(P)
```

```
ans = 2x1
    0.2500
    1.0000
```

```
fimplicit (x'*P*x-1,LineWidth=3), hold on, grid on, axis equal
1./sqrt(diag(D))' %radio: mínimo y máximo
```

```
ans = 1x2
     2     1
```

```
semiej=V/sqrt(D);
plot([0;semiej(1,1)], [0;semiej(2,1)])
plot([0;semiej(1,2)], [0;semiej(2,2)])
fimplicit (x'*D(1,1)*eye(2)*x-1,'-.',LineWidth=1.5)
fimplicit (x'*D(2,2)*eye(2)*x-1,'-.',LineWidth=1.5)
hold off
legend("", "semiaxis 1", "semiaxis 2", "circunscrita D11", "inscrita D22", Location="bestout")
```

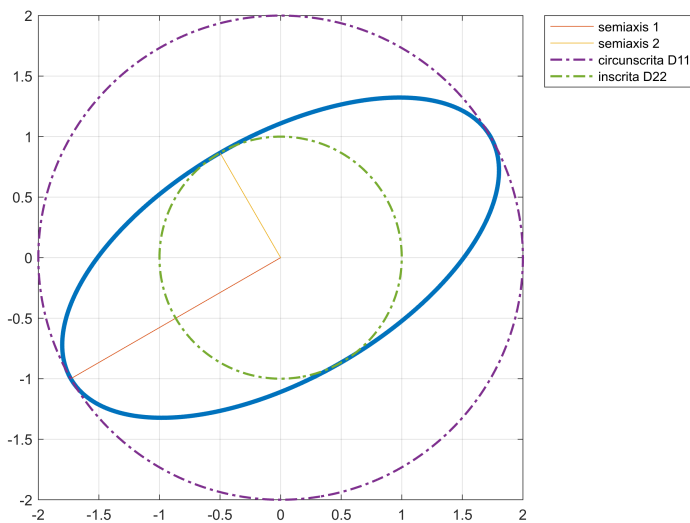


Imagen del círculo unidad por matriz L

```
theta=(0:6:360);
idxaxis=[1 16 31 46];
theta(idxaxis)
```

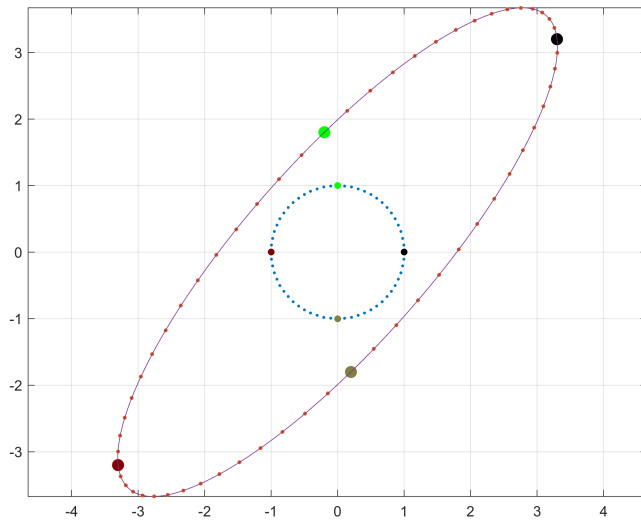
```
ans = 1x4
     0     90    180    270
```

```
theta_rad=theta/180*pi;
PuntosCircunfl=[cos(theta_rad); sin(theta_rad)];
```

```
L=[3.3 -0.2; 3.2 1.8]
```

```
L = 2x2
    3.3000    -0.2000
    3.2000     1.8000
```

```
PuntosEllipse=L*PuntosCircunfl;
plot(PuntosCircunfl(1,:),PuntosCircunfl(2,:),'.',LineWidth=2)
hold on,
plot(PuntosEllipse(1,:),PuntosEllipse(2,:),'.',MarkerSize=8,LineWidth=2)
Colors=[0 0 0;0 1 0;0.5 0 0;0.5 .5 0.25];
for i=1:4
    plot(PuntosCircunfl(1,idxaxis(i)),PuntosCircunfl(2,idxaxis(i)),'k.',MarkerSize=14,Color=Colors(i,:))
    plot(PuntosEllipse(1,idxaxis(i)),PuntosEllipse(2,idxaxis(i)),'k.',MarkerSize=25,Color=Colors(i,:))
end
fimplicit(x'*inv(L*L')*x-1)
hold off, grid on, axis equal
```



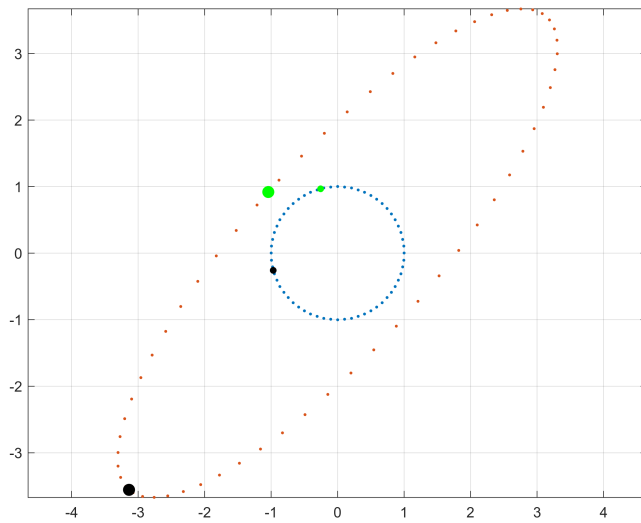
SVD de L

```
[U,S,V]=svd(L);
L=U*S*V'
```

```
ans = 2x2
10^-14 x
    -0.1776    -0.0389
    -0.1332    -0.0222
```

```
plot(PuntosCircunfl(1,:),PuntosCircunfl(2,:),'.',LineWidth=2)
hold on,
plot(PuntosEllipse(1,:),PuntosEllipse(2,:),'.',LineWidth=2)
Colors=[0 0 0;0 1 0;0.5 0 0;0.5 .5 0.25];
for i=1:2
    plot(V(1,i),V(2,i),'k.',MarkerSize=14,Color=Colors(i,:))
    plot(U(1,i)*S(i,i),U(2,i)*S(i,i),'k.',MarkerSize=25,Color=Colors(i,:))
end
```

```
end
hold off, grid on, axis equal
```



Inclusión

```
P1=[5 3;3 2];
eig(P1)'
```

```
ans = 1x2
    0.1459    6.8541
```

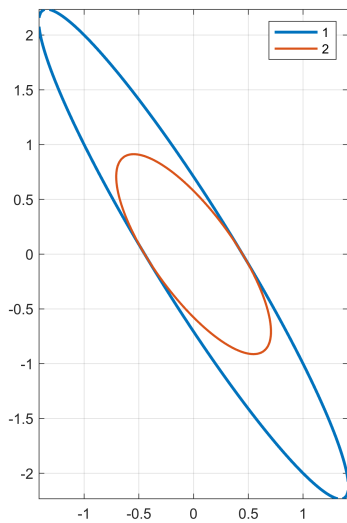
```
P2=[5 3;3 3];
eig(P2)'
```

```
ans = 1x2
    0.8377    7.1623
```

```
eig(P1-P2) %no tienen nada que ver con la suma o resta de los autovalores de P1 o P2, o
```

```
ans = 2x1
    -1
     0
```

```
fimplicit(x'*P1*x-1,LineWidth=2), hold on, grid on, axis equal
fimplicit(x'*P2*x-1,LineWidth=1.5),
hold off, legend("1","2")
```



Transformación lineal (ejemplo proyección)

Sobre eje horizontal:

P

```
P = 2x2
    0.4375    -0.3248
   -0.3248     0.8125
```

```
C=[1 0]; %C is a projection matrix, projecting on x1.
P_proj1=inv(C*inv(P)*C')
```

```
P_proj1 = 0.3077
```

```
P_proj1=P(1,1)-P(1,2)*P(2,1)/P(2,2) %Schur complement formula
```

```
P_proj1 = 0.3077
```

```
Proj1=1/sqrt(P_proj1); %semiaxis
```

Sobre eje vertical:

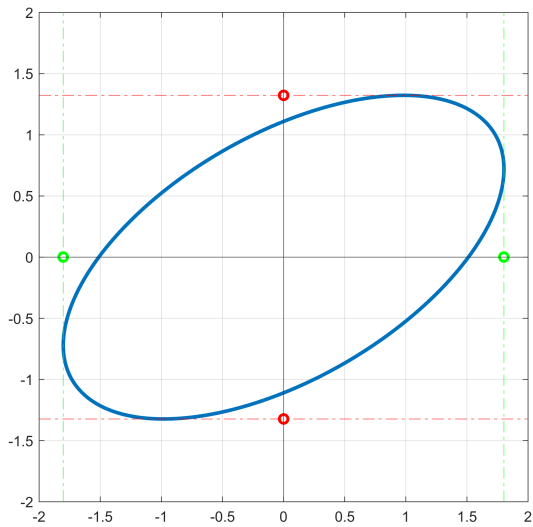
```
P_proj2=inv([0 1]*inv(P)*[0;1]) %[0 1] is a projection matrix
```

```
P_proj2 = 0.5714
```

```
Proj2=1/sqrt(P_proj2); %semiaxis
fimplicit(x'*P*x-1,LineWidth=2.5), axis equal %original ellipse
hold on
plot(Proj1,0,'og',LineWidth=2), plot(-Proj1,0,'og',LineWidth=2)
plot(0,Proj2,'or',LineWidth=2), plot(0,-Proj2,'or',LineWidth=2)
hold off
xline(0), yline(0)
xline(-Proj1,'-.g')
```



```
xline(Proj1,'-.g')
yline(Proj2,'-.r')
yline(-Proj2,'-.r')
axis([-2 2 -2 2]), grid on
```



3D elipsoide (o hiperboloide)

```
P=[3 2 1;2 0.5 2;1 2 5] %hyperboloid
```

```
P = 3x3
    3.0000    2.0000    1.0000
    2.0000    0.5000    2.0000
    1.0000    2.0000    5.0000
```

```
%P=[3 2 1;2 1.7143 2;1 2 5] %cylinder (degenerate ellipsoid)
%P=[3 2 1;2 1.9 2;1 2 5] %ellipsoid
pei=eig(P)'
```

```
pei = 1x3
   -0.9138    2.8222    6.5917
```

```
semiax=1./sqrt(pei)
```

```
semiax = 1x3 complex
    0.0000 - 1.0461i    0.5953 + 0.0000i    0.3895 + 0.0000i
```

```
syms z [3 1] real
fimplicit3(z'*P*z-1,'EdgeColor','none','FaceAlpha',.8), axis equal
C=[eye(2) zeros(2,1)]
```

```
C = 2x3
    1    0    0
    0    1    0
```

```
Mproj12=inv(C*inv(P)*C')
```

```
Mproj12 = 2x2  
    2.8000    1.6000  
    1.6000   -0.3000
```

```
hold on  
fimplicit(x'*P(1:2,1:2)*x-1,'r',LineWidth=3)  
fimplicit(x'*Mproj12*x-1,'b',LineWidth=2.5)  
hold off
```

