IGL Projects

---Curves: A Course for AIMS

Group Member

Professor: Steven Bradlow
Graduate Student: Nathan Orlow
Undergraduate Students:
Hiroshi Fujii
Xudong Wang

Short Description

In January 2013 Professor Bradlow will teach a three-week course on `Curves' at the African Institute of Mathematical Sciences (AIMS). The course is meant to be very interactive, with lots of `hands on' exercises and demonstrations. The project is to help him develop material for this course.

Project 1:
Goal:

- Explore the Geometric property of the conic sections.
- Make a 3D model that shows how conics are created by intersecting planes with different slopes.
- Make a 3D model that shows how foci in conics can also be represented geometrically by utilizing dandelion spheres.

Images:
Calculations:
To get a desired conic section, we must figure out what slope of a plane will produce what kind of conic section with the cone.

To find the equation, we will rotate the cone in different angles and intersect it with the XY plane. Also we will shift it in the positive z direction 4 units.

Circle:
Angle of rotation: 0
Equation of cone:
\[ Z = -\sqrt{x^2 + y^2} + 4 \]
Set it equal to zero.
\[ 0 = -\sqrt{x^2 + y^2} + 4 \]
Then we get the equation of a circle: \[ 16 = x^2 + y^2 \]

Ellipse:
Angle of rotation: \( \pi / 6 \)
Equation of cone:
\[(x \sin(\pi/6) + \cos(\pi/6) \cdot (z - 4))^2 - (x \cos(\pi/6) - \sin(\pi/6) \cdot (z - 4))^2 - y^2 = 0\]
We will set \( z \) equal to zero, and as a result we get an equation of an ellipse.

Parabola:
Angle of rotation: \( \pi / 4 \)
Equation of cone:
\[
\left( x \cdot \sin \left( \frac{\pi}{4} \right) + \cos \left( \frac{\pi}{4} \right) \cdot (z - 4) \right)^2 - \left( x \cdot \cos \left( \frac{\pi}{4} \right) - \sin \left( \frac{\pi}{4} \right) \cdot (z - 4) \right)^2 - y^2 = 0
\]

We will set \( z \) equal to zero, and as a result we get an equation of a parabola.

**Hyperbola:**
- Angle of Rotation: \( \pi/3 \)
- Equation of cone:

\[
\left( x \cdot \sin \left( \frac{\pi}{4} \right) + \cos \left( \frac{\pi}{4} \right) \cdot (z - 4) \right)^2 - \left( x \cdot \cos \left( \frac{\pi}{4} \right) - \sin \left( \frac{\pi}{4} \right) \cdot (z - 4) \right)^2 - y^2 = 0
\]

Setting \( z \) to 0, we get an equation of a hyperbola.

**Dandelin Spheres:**
Dandelin spheres are described as spheres that are tangent to the cone, and the plane of intersection. The tangent points of the plane will become the focus of the conic section made. First, we need to find what the slope of the cone will be, and the slope of the plane that we are going to cut it with. To figure these out, we can start from two dimensions, then extend it to the three dimensions.

First, we will make a circle of radius 2 on the origin.

\[
y^2 + x^2 = 4
\]

Then create two lines that will represent the cone in 2D. They must be both tangent, and one must be the reflection of the other in the x-axis.

\[
y = - \cot \left( \frac{\pi}{6} \right) + 4
\]

\[
y = \cot \left( \frac{\pi}{6} \right) + 4
\]

Next, we will add the plane that cuts the cone. Since we are making an ellipse, the slope must
be less than the sides of the cone. Additionally, it has to be tangent to the circle we had in the beginning.

\[ y = \tan\left(\frac{\pi}{6}\right) + 4/\sqrt{3} \]

Now we will create another dandelin circle. This must be tangent to \( y = \tan\left(\frac{\pi}{6}\right) + 4/\sqrt{3} \) and \( y = \cot\left(\pi/6\right) + 4 \), and also have its center on the y-axis. Thus, we can find the intersection of the y-axis and the angle bisector of these two lines, and that will be the center.

\[
\left( y - \left( 4 + \frac{4}{\sqrt{3}} + \tan\left(\frac{\pi}{12}\right) \cdot \frac{4-4\sqrt{3}}{\sqrt{3}+\sqrt{3}} \right) \right)^2 + x^2 = (\sqrt{3} - 2)^2 + (2\sqrt{3} - 3)^2
\]

Thus, we can see this diagram

Now we can just change this 2D model into 3D by changing circles into spheres, and lines into planes, and the sides into cones.

And in the end, we can get the dandelin sphere models.
Project 2

Images for Brochure

Conic Sections and 2*2 Matrix

The curve of the conic section is $ax^2+2hxy+by^2+2fx+2gy+c=0$

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>e</td>
</tr>
<tr>
<td>f</td>
<td>g</td>
</tr>
</tbody>
</table>

The curve of the conic section is $x^2 + 4xy + 3y^2 + 2x - 3y = 2$

The curve of the conic section in the new coordinate system is $1/\sqrt{103}x^2 + 1/\sqrt{72}y^2 = 1$

**NOTE:**
- If $b^2 > ad$, then the curve is a hyperbola. E.g. $a=2$, $b=3$, $d=3$
- If $b^2 < ad$, then the curve is an ellipse. E.g. $a=2$, $b=1$, $d=3$
- If $b^2 = ad$, then the curve is a parabola. E.g. $a=1$, $b=2$, $d=4$, $f=4$

Interactive Module

http://www.sagenb.org/home/pub/5037

Project Report

1. **Goal:**
   - Explore the relationship between symmetric 2*2 matrices, quadratic forms, and conic sections.
   - Given a coefficient matrix, identify the conic.
   - Show how diagonalizing the matrix corresponds to rotating the conic section into a standard position.
Show how the addition of the linear terms corresponds to shifting the conic section into a standard position
Show that for any degree two polynomials in two variables, the zero set is a conic section
Explore how to identify hyperbolas, ellipses, and parabolas in terms of the coefficients of the polynomial
Transform the conic section in the general form into standard form with a new coordinate system

2. Background

If

\[ Q(x, y) = ax^2 + 2bxy + dy^2 + fx + gy + e \]

\( Q(x, y) = 0 \) is a conic section and it is the general form.

To begin with set \( f=0 \) and \( g=0 \)

\[ ax^2 + 2bxy + dy^2 = -e \]

i.e.

\[ \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} a & b \\ b & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = -e \]

Let \( A = \begin{pmatrix} a & b \\ b & d \end{pmatrix} \)

Diagonalizing \( A \) we get

\[ D = Q^T A Q \]

We get two eigenvalues \( \lambda_1, \lambda_2 \) and eigenvectors \( \mathbf{v}_1, \mathbf{v}_2 \)

Now the conic can be represent in a new coordinate system

\[ \lambda_1 x'^2 + \lambda_2 y'^2 = -e \]

Now we add the linear terms

The new equation is

\[ \lambda_1 x'^2 + \lambda_2 y'^2 + f'x' + g'y' = -e \]

i.e.

\[ \lambda_1 \left( x' + \frac{f'}{2} \right)^2 + \lambda_2 \left( y' + \frac{g'}{2} \right)^2 = \frac{f'^2}{4} + \frac{g'^2}{4} - e \]

So we can represent the conic in a new coordinate system in standard form
\[ X'^2 + Y'^2 = 1 \]

3. CODE Description

\[
\text{html}('\text{Conic Sections and 2*2 Matrix} <center> Conic Sections and 2*2 Matrix </center> \text{title} \n\text{html}('The curve of the conic section is ax^2+2bxy+dy^2+fx+gy=e \n') \n\text{var}(x,y,X,Y) \n\text{html}('The curve of the conic section is $%sx^2+%sxy+%sy^2+%sx+%sy=%se$%%(a,2*b,d,f,g,e)) \n\text{html}('') \n\text{A} = \text{matrix}(2,2,[a,b,b,d]) \n\text{k}(x,y)=a*x^2+2*b*x*y+d*y^2+f*x+g*y \n\text{h}(x,y)=y \n\text{v}(x,y)=x \n\text{plot1} = \text{implicit_plot}(k-e,(x,-viewsize,viewsize),(y,-viewsize,viewsize)) \n\text{plot2} = \text{implicit_plot}(h,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='black') \n\text{plot3} = \text{implicit_plot}(v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='black') \n\text{V}=\text{A.eigenvectors_right()} \n\text{la1=V[0][0]} \ #eigenvalue1 \n\text{la2=V[1][0]} \ #eigenvalue2
lam1=numerical_approx(la1, digits=3)
lam2=numerical_approx(la2, digits=3)

ev0=V[0][1][0] #eigenvector1(new x axis)
ev1=V[1][1][0] #eigenvector2(new y axis)

v=vector(ev0)
v1=v.norm()
v2=v/v1 #normalization of eigenvector1

o=vector(ev1)
o1=o.norm()
o2=o/o1 #normalization of eigenvector2

r=v2[0]
s=o2[0]
t=v2[1]
u=o2[1]

B = matrix(2,2,[r,s,t,u]) #Get the orthogonal matrix B
B1=B^-1

fg=vector([f,g])

fg1=B1*fg #get new f1 and g1

fga1=fg1[0] #f1
fga2=fg1[1] #g1

if b^2==a*d: #discuss
    html(" NOTE:
")
    html(" ")
    html(" ")
If $b^2$ is greater than $ad$, then the curve is a hyperbola. E.g. $a=2$ $b=3$ $d=3$

If $b^2$ is less than $ad$, then the curve is an ellipse. E.g. $a=2$ $b=1$ $d=3$

If $b^2$ is equal to $ad$, $f$ and $g$ not zero then the curve is a parabola. E.g. $a=1$ $b=2$ $d=4$ $f=4$.

```
a0=ev0[1]
b0=ev0[0]
a1=ev1[1]
b1=ev1[0]
k0=a0/b0
k1=a1/b1

kk0=numerical_approx(k0, digits=3)
kk1=numerical_approx(k1, digits=3) # get the slope of the line of new axes

k(x,y)=a*x^2+2*b*x*y+d*y^2+f*x+g*y # conic equation
h(x,y)=y
v(x,y)=x

plot6=
  implicit_plot(h-kk0*v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

plot7=
  implicit_plot(h-kk1*v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

show(plot1+plot2+plot3+plot6+plot7)
```

else:
  fgb1=fga1/lam1
  fgb2=fga2/lam2
e1=e+(fgb1/2)^2+(fgb2/2)^2

lama1=lam1/e1
lama2=lam2/e1 # get the new lama
The curve of the conic section in the new coordinate system is
\[ \frac{1}{S}x^2 + \frac{1}{S}y^2 = 1 \]

\[ \frac{1}{lama1} \]

\[ \frac{1}{lama2} \]

html("NOTE:")

html("If b^2 is greater than ad, then the curve is a hyperbola. E.g. a=2 b=3 d=3")

html("If b^2 is less than ad, then the curve is a ellipse. E.g. a=2 b=1 d=3")

html("If b^2 is equal to ad, then the curve is a parabola. E.g. a=1 b=2 d=4 f=4")

fg2=vector([fgb1,fgb2])/2

f1=fg2[0]
g1=fg2[1] #shift unit

fa1=f1*v2
ga1=g1*o2

mid=-1*(fa1+ga1)

midx=mid[0]
midy=mid[1]

a0=ev0[1]
b0=ev0[0]

a1=ev1[1]
b1=ev1[0]

k(x,y)=a*x^2+2*b*x*y+d*y^2+f*x+g*y

h(x,y)=y
v(x,y)=x
plot1 = implicit_plot(k-e,(x,-viewsize,viewsize),(y,-viewsize,viewsize))
plot2 = implicit_plot(h,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='black')
plot3 = implicit_plot(v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='black')

if b0==0:
    fb1=f1*ev1
    gb1=g1*ev0
    mid1=-1*(fb1+gb1)
    midx1=mid1[0]
    midy1=mid1[1]

    plot6= implicit_plot(h,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')
    plot7= implicit_plot(v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

    plot4= implicit_plot(h-midx1,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')
    plot5= implicit_plot(v-midy1,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')

    show(plot1+plot2+plot3+plot4+plot5+plot6+plot7)
elif b1==0:
    fc1=f1*ev1
    gc1=g1*ev0
    mid2=-1*(fc1+gc1)
    midx2=mid2[0]
    midy2=mid2[1]

    plot6= implicit_plot(h,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')
    plot7= implicit_plot(v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

    plot4= implicit_plot(h-midx2,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')
    plot5= implicit_plot(v-midy2,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')

    show(plot1+plot2+plot3+plot4+plot5+plot6+plot7)
else:
    k0=a0/b0
    k1=a1/b1

    kk0=numerical_approx(k0, digits=3)
    kk1=numerical_approx(k1, digits=3)
hh1=midy-kk0*midx #intercept1
hh2=midy-kk1*midx #intercept2

plot4= implicit_plot(h-kk0*v-hh1,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')

plot5= implicit_plot(h-kk1*v-hh2,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='red')

plot6= implicit_plot(h-kk0*v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

plot7= implicit_plot(h-kk1*v,(x,-viewsize,viewsize),(y,-viewsize,viewsize),color='green',linestyle='--')

show(plot1+plot2+plot3+plot4+plot5+plot6+plot7)