MATH 2270-2
Symmetric matrices, conics and quadrics
December 3, 2001

Conic sections:

```
> restart:
> with(linalg):with(plots):#for computations and pictures
  with(student):#to do algebra computations like completing the
  square
```

Let's do the problem we did in class this past Wednesday, using Maple:

```
> A:=matrix(2,2,[2,5/2,5/2,2]);
```

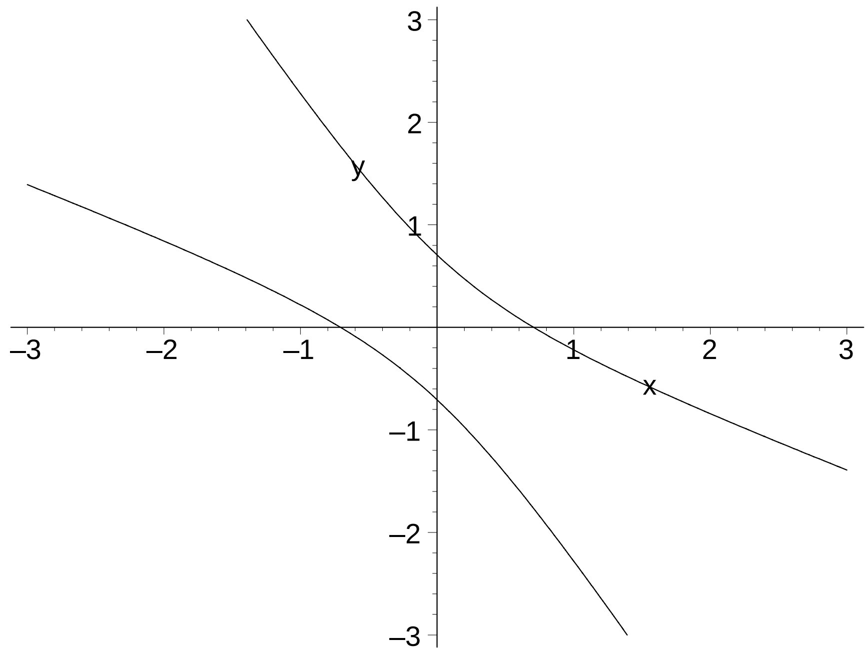
$$A := \begin{bmatrix} 2 & \frac{5}{2} \\ \frac{5}{2} & 2 \end{bmatrix}$$

```
> eigenvalues(A); #compute eigenvalues
```

$$\frac{9}{2}, \frac{-1}{2}$$

Since the eigenvalues have the opposite sign, the conic must be a hyperbola (or two lines crossing) . We can verify this immediately by using the implicitplot command:

```
> implicitplot(2*x^2+2*y^2+5*x*y= 1,
         x=-3..3,y=-3..3, grid=[100,100],
         color='black');
```



Yes, it was a hyperbola.

Now, what about all that work we did to explicitly pick new coordinates in which there was no cross term? If we really need to do that, we can try to let Maple help, and although the commands seem a little cumbersome at first, when we're done we'll have a template that will work for any quadratic equation with only minor modifictations.

Let's do #30 page 501 of Kolman: We will write our quadratic as x(^T)Ax + Bx + C=0:

```
> A:=matrix(2,2,[8,-8,-8,8]);
      #matrix for quadratic form
  B:=matrix(1,2,[33*sqrt(2),-31*sqrt(2)]);
      #row matrix for linear term
  C:=70;
      # constant term
```

$$A := \begin{bmatrix} 8 & -8 \\ -8 & 8 \end{bmatrix}$$

$$B := [33\sqrt{2} \quad -31\sqrt{2}]$$

$$C := 70$$

```
> fmat:=v->evalm(transpose(v)&*A&*v
                 + B&*v +C) ;
       #this function takes a vector v and computes
       # transpose(v)Av + Bv + C ,
       #which will be a one by one matrix
  f:=v->simplify(fmat(v)[1]); #this extracts the entry of the one
       #by one matrix fmat(v)
```

$$fmat := v \to \text{evalm}(\text{`\&*`}(\text{`\&*`}(\text{transpose}(v), A), v) + \text{`\&*`}(B, v) + C)$$

$$f := v \to \text{simplify}(\text{fmat}(v)_1)$$

```
> fmat(vector([x,y])); #should be a one by one matrix
  f(vector([x,y]));  #should get its entry, which is what we want.
     #should agree with our example
```

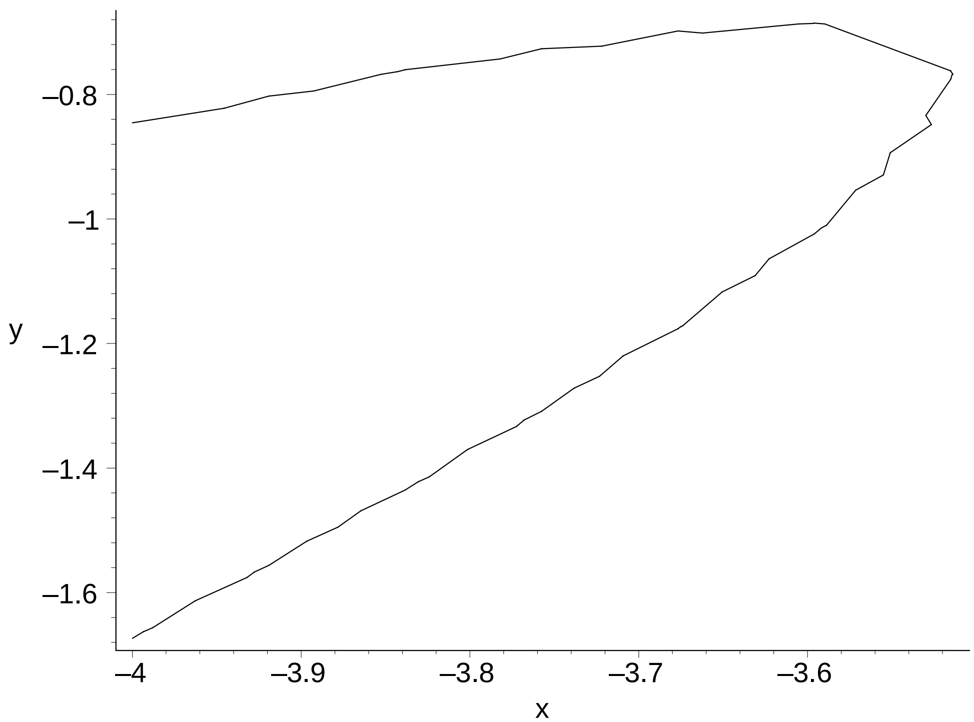
$$[33\sqrt{2}\,x - 31\sqrt{2}\,y + (8\,x - 8\,y)\,x + (-8\,x + 8\,y)\,y + 70]$$

$$33\sqrt{2}\,x - 31\sqrt{2}\,y + 8\,x^2 - 16\,x\,y + 8\,y^2 + 70$$

```
> eqtn:=f([x,y])=0;
     #should be our equation
```

$$eqtn := 33\sqrt{2}\,x - 31\sqrt{2}\,y + 8\,x^2 - 16\,x\,y + 8\,y^2 + 70 = 0$$

```
> implicitplot(eqtn,x=-4..4,y=-4..4, color=black,
      grid=[100,100]);
      #what are we in for?
>
```



Now let's go about the change of variables:

```
> data:=eigenvects(A);#get eigenvectors
```

$$data := [16, 1, \{[-1, 1]\}], [0, 1, \{[1, 1]\}]$$

You pick things out of the object above systematically, using inidices to work through the nesting of brackets:

```
> data[1];#first piece of data
```

$$[16, 1, \{[-1, 1]\}]$$

```
> data[1][1];#eigenvalue
```

```
> data[1][2];#algebraic multiplicity
```

$$1$$

```
> data[1][3];#basis for eigenspace
```

$$\{[-1, 1]\}$$

```
> data[1][3][1];#actual eigenvector
```

$$[-1, 1]$$

So that's how to extract the eigenvectors:

```
> v1:=data[1][3][1];#first eigenvector
  v2:=data[2][3][1];#second eigenvector
  w1:=v1/norm(v1,2);#normalized
  w2:=v2/norm(v2,2);#normalized
```

$$v1 := [-1, 1]$$

$$v2 := [1, 1]$$

$$w1 := \frac{1}{2}\, v1 \sqrt{2}$$

$$w2 := \frac{1}{2}\, v2 \sqrt{2}$$

```
> P:=augment(w1,w2):#our orthogonal matrix
    if det(P)<0 then P:=augment(w2,w1):
    fi:
  evalm(P);
```

$$\begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix}$$

```
> f(evalm(P&*[u,v]));#do the change of variables
```

$$2\,u - 64\,v + 16\,v^2 + 70$$

```
> simplify(%);#simplify it!
```

$$2\,u - 64\,v + 16\,v^2 + 70$$

```
> completesquare(%,u); #complete the square in u
```

$$2\,u - 64\,v + 16\,v^2 + 70$$

```
> completesquare(%,v);
```

$$16\,(v-2)^2 + 6 + 2\,u$$

```
> Eqtn:=%=0;
    #this is the new equation
```

$$Eqtn := 16\,(v-2)^2 + 6 + 2\,u = 0$$

Let's collect everything into one template . You can see from where the colons and semicolons are that the output will be the eigenvalues, the transition matrix, a plot, and an equation just short of standard form. (You can improve it by picking a denser grid size.)

```
> restart;with(linalg):with(plots):with(student):
>
  A:=matrix(2,2,[8,-8,-8,8]);
  B:=matrix(1,2,[33*sqrt(2),-31*sqrt(2)]);
  C:=70;
  fmat:=v->evalm(transpose(v)&*A&*v
                 + B&*v + C):
  f:=v->simplify(fmat(v)[1]):
  eigenvals(A); #show the eigenvalues
  data:=eigenvectors(A):
  v1:=data[1][3][1]:#first eigenvector
  v2:=data[2][3][1]:#second eigenvector
  w1:=v1/norm(v1,2):#normalized
  w2:=v2/norm(v2,2):#normalized
  P:=augment(w1,w2):#our orthogonal matrix,
    #unless we want to switch columns
    if det(P)<0 then
       P:=augment(w2,w1):
    fi:
  evalm(P);
  implicitplot(f([x,y])=0,x=-5..5,y=-5..5,
       grid=[100,100],color=‘black‘);
       #increase grid size for better picture, but too big
       #takes too long
  f(evalm(P&*[u,v])):#do the change of variables
  simplify(%):#simplify it!
  completesquare(%,u): #complete the square in u
  completesquare(%,v):#and v
  Eqtn:=%=0;
```

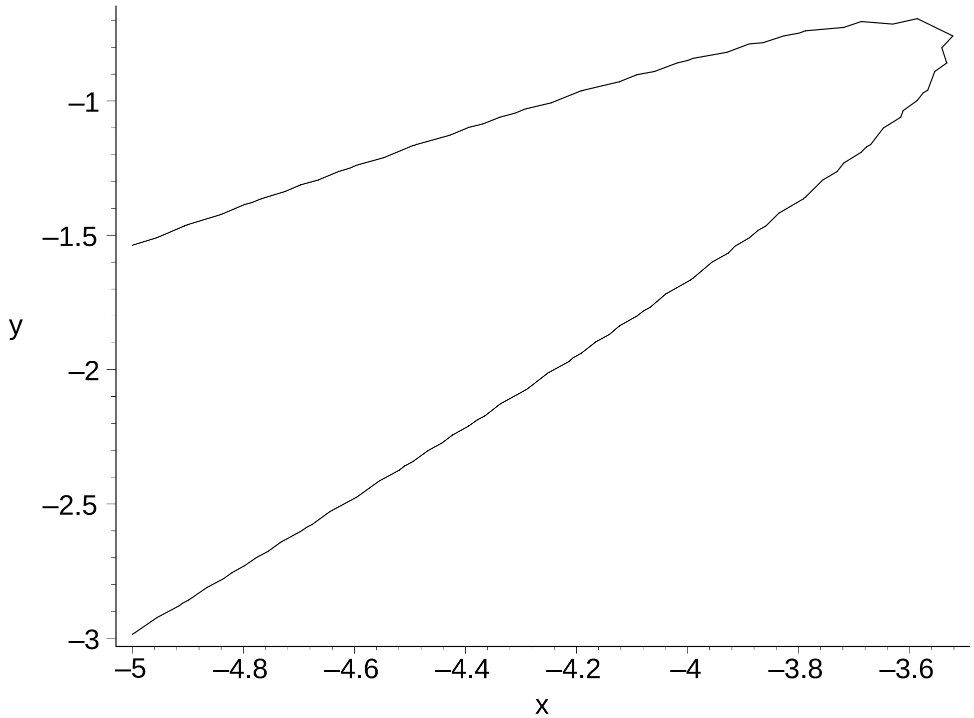
$$A := \begin{bmatrix} 8 & -8 \\ -8 & 8 \end{bmatrix}$$

$$B := [33\sqrt{2} \quad -31\sqrt{2}]$$

$$C := 70$$

$$0, 16$$

$$\begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix}$$



$$Eqtn := 16\,(v-2)^2 + 6 + 2\,u = 0$$

You could check your book problems about conic sections using such a template

Quadric Surfaces:

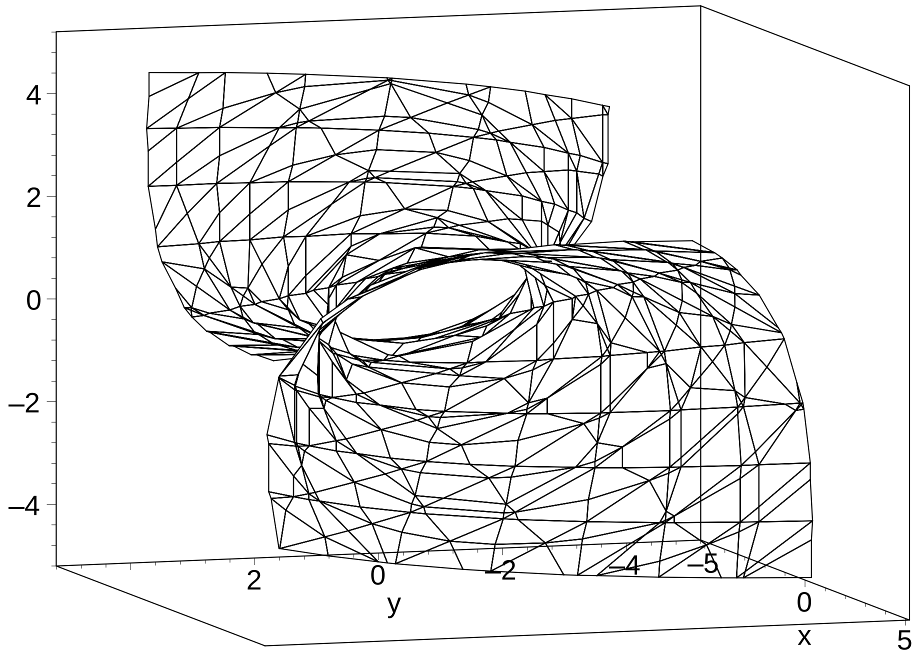
Bonus Question : **worth 10 midterm points**: Create a template which will diagonalize, graph, and put into standard form, the solution set to any quadratic equation in 3 variables - i.e. analogous to the template above which works for conic sections. You need only create a template which works when there are 3 distinct eigenvalues. (If you successfully tackle the case of higher geometric multiplicity y ou will need several conditionals to allow for the structure of the eigenvects array. Also, in that case you will need to use the Maple Gram-Schmidt command or your own, to get an orthonormal basis of the offending eigenspace.

To make 3-d plots you can use implicitplot3d. If you make your grid too fine the plot will take a long time to make, so try the default grid at first and adjust if necessary. You might also have to adjust your limits in the plot to get a better picture. You can manipulate 3d plots with your mouse. There are a lot of interesting plot options which you write into the command or access from the plotting toolbar. One that helps me see things is to use the ‘‘boxed’’ axes option.

The 3d plotting routines have been known to crash Maple, so save your file often. Sometimes what really happens is that a dialog box gets opened behind one of your windows, where you can’t see it, so that it will appear that Maple has frozen. If this happens, minimize your maple window (the black dot a the upper left corner), and then re-display it.

Here’s a plot to play with, #1 on page 511. You should be able to move it around to make it look like a one-sheeted hyperboloid.

```
> implicitplot3d(x^2 + y^2 +2*z^2 - 2*x*y -4*x*z -4*y*z + 4*x = 8,
                 x=-5..5,y=-5..5,z=-5..5, axes=‘boxed‘);
```


4
2
0
–2
–4
2
0
–2
–4
–5
y
0
x
5

>