CS 322: Assignment P1

Due: Wednesday, February 9, 2002 (In Lecture)

You may work with at most one other person. Follow the course rules for the submission of assignments. Do not submit work unless you have adhered to the principles of academic integrity as descibed on the course website. Points will be deducted for poorly commented code, redundant computation that seriously effects efficiency, and failure to use features of MATLAB that are part of the course syllabus. In particular, use vector operations whenever possible.

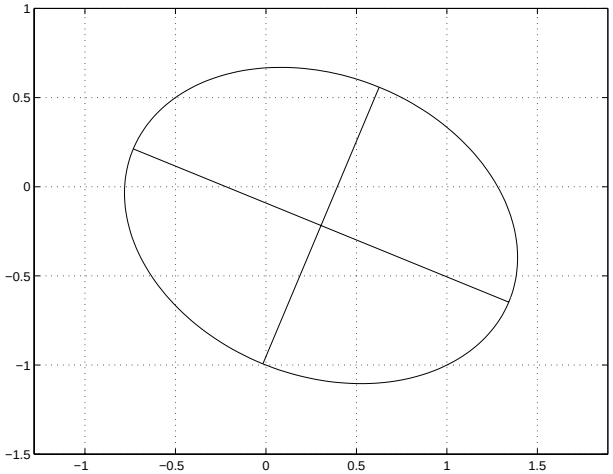
Representing an Ellipse

The general recipe for a conic section is given by

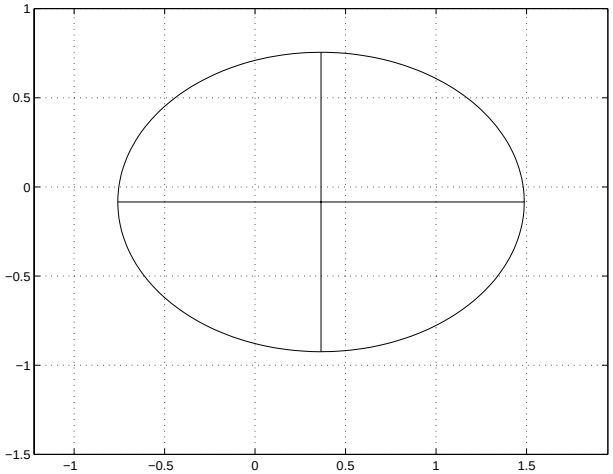
$$\mathcal{E} = \left\{ (x, y) : Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \right\}. \tag{1}$$

Depending upon the values of the coefficients, $\mathcal{E}$ could define a single point, a pair of lines, a circle, an ellipse, an hyperbola, a parabola, or just the empty set. It describes a nondegenerate ellipse if $A > 0$, $C > 0$, $B^2 - 4AC < 0$, and some additional conditions on D, E, and F hold. The goal in Parts A, B, and C is to plot $\mathcal{E}$ if it is nonempty and specifies a nondegenerate ellipse.

Equation (1) is not a particularly handy representation for an ellipse when it comes to plotting because it does not explicitly define y in terms of x or vice versa. It turns out that life is made difficult by the xy term, whose presence means that the ellipse is "tilted". For example, here is what $2x^2 + xy + 3y^2 - x + y - 2 = 0$ and its axes look like:



However, we can rotate this ellipse counterclockwise about the origin so that its axes are parallel to the coordinate axes:



An ellipse like this has a specification of the following form

$$\mathcal{E}' \;=\; \left\{\, (x,y) : A'x^2 + C'y^2 + D'x + E'y + F' = 0 \right\}$$

With a little algebraic manipulation we can specify $\mathcal{E}'$ in terms of the ellipse center (h,k) and semiaxes a and b:

$$\mathcal{E}' \;=\; \left\{ (x,y) : \left(\frac{x-h}{a}\right)^2 \;+\; \left(\frac{y-k}{b}\right)^2 \;=\; 1 \right\}.$$

The parametric representation

$$\mathcal{E}' \;=\; \left\{\, (x,y) : x = h + a\cos(\theta),\; y = k + b\sin(\theta)),\; 0 \le \theta \le 2\pi \right\}.$$

is particularly convenient for plotting in MATLAB:

```
theta = linspace(0,2*pi); x = h + a*cos(theta); y = k + b*sin(theta);
plot(x,y)
axis equal
```

Of course, we want to plot $\mathcal{E}$, not $\mathcal{E}'$. But once we know (h,k), a, b, and the rotation angle ϕ that was used to obtain $\mathcal{E}'$ from $\mathcal{E}$, we simply rotate the display points defined by `x` and `y` by $-\phi$ in order to get display points on $\mathcal{E}$. In Parts A, B, and C you will write functions that implement these ideas.

Part A (5 pts) Rotating a Conic

Suppose ϕ is a radian measure and $c = \cos(\phi)$ and $s = \sin(\phi)$. If in the equation

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

we replace x with $xc + ys$ and y by $-sx + cy$ and simplify, then we obtain an equation of the form

$$A'x^2 + B'xy + C'y^2 + D'x + E'y + F' = 0.$$

Geometrically, the new conic section is obtained by rotating the original conic section ϕ radians counterclockwise about the origin. Work out recipes for the new coefficients (e.g., $D' = Dc - Es$) and implement the following function

```
  function New_Kappa = RotateConic(Kappa,phi)
% Kappa is a structure array with fields A,B,C,D,E, and F that
% represents the conic A*x^2 + B*xy + C*y^2 + D*x + E*x + F = 0.
% New_Kappa  is a structure array of the same form that represents a
% new conic obtained from the original by counterclockwise rotation about the
% origin by an angle whose measure is phi radians.
```

When "building" conics in this assignment, make use of the function

```
  function Z = MakeConic(A,B,C,D,E,F)
% Represents the conic defined by Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0
Z = struct('A',A,'B',B,'C',C,'D',D,'E',E,'F',F);
```

This is available on the course website. Test your implementation of `RotateConic` by running the script `P1A` which is also available on the website. It will produce two figures. Drag the legends so that they do not hide the graphs. Print the two figures and submit them both together with a listing of `RotateConic`.

Part B (5 pts) The Center-Axis-Tilt Representation

Equate to zero the expression for B' that you derived in Part A. By manipulating this equation obtain a simple recipe involving A, B, and C for

$$\tan(2\phi) = \frac{\sin(2\phi)}{\cos(2\phi)} = \frac{2\cos(\phi)\sin(\phi)}{\cos(\phi)^2 - \sin(\phi)^2}.$$

The built-in inverse tangent function `atan` can be used to solve for ϕ. This gives a formula for the rotation parameter ϕ. You should notice from `P1A` output that there are four different values of ϕ in the interval $[0, 2\pi)$ that work. We're not fussy about which one is utilized by your algorithm .

Using `RotateConic` we can obtain a representation $A'x^2 + C'y^2 + D'x + E'y + F' = 0$ for the untilted ellipse. Note that if $A' > 0$ and $C' > 0$ then

$$A'x^2 + C'y^2 + D'x + E'y + F' = A'\left(x^2 + \frac{D'}{A'} + \left(\frac{D'}{2A'}\right)^2\right) + C\left(y^2 + \frac{E'}{C'} + \left(\frac{E'}{2C'}\right)^2\right) - \tilde{F} = 0 \qquad (2)$$

where

$$\tilde{F} = -F' + \frac{D'^2}{4A'} + \frac{E'^2}{4C'}.$$

Assuming that $\tilde{F} > 0$, manipulate (2) so that it has the form

$$\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 = 1$$

thereby specifying the center (h, k) and axes a and b of the untilted ellipse. These four parameters together with ϕ make up the center-axis-tilt representation for the given ellipse that is specified by $\mathcal{E}$. Using these ideas, implement the following function:

```
  function Eta = MakeEllipse(Kappa)
% If the points (x,y) that satisfy
%
%       Kappa.A*x^2 + Kappa.B*xy + Kappa.C*y^2 + Kappa.D*x + Kappa.E*y + Kappa.F = 0
%
% define a nondegenerate ellipse, then that ellipse is obtained by rotating
% the ellipse ((x - Eta.h)/Eta.a)^2  + ((y - Eta.k)/Eta.b)^2 = 1
% clockwise about the origin by an angle that has measure Eta.phi radians.
% If Kappa does not define a nondegenerate ellipse, then Eta is the empty matrix.
```

Note that the empty matrix should be returned if $A' \leq 0$, $B' \leq 0$, or $\tilde{F} \leq 0$ where $\tilde{F}$ is defined above. Test your implementation by running the script `P1B` on the website. It prints h, k, a, b, and ϕ for every nondegenerate ellipse with $A \in \{0, 1\}$, $B \in \{0, 1, 2\}$, $C \in \{0, 1, 2\}$, $D \in \{-1, 0, 1\}$, $E \in \{-1, 0, 1\}$, and $F \in \{-1, 0, 1\}$. For your information, there are fewer than 30 nondegenerate ellipses with this form. Submit a copy of the output and a listing of `MakeEllipse`.

Part C (5 pts) Plotting the Tilted Ellipse

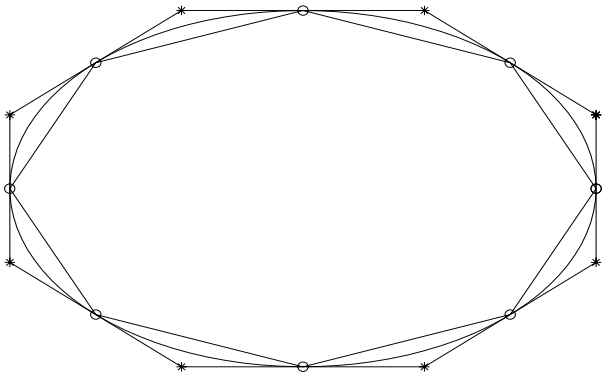
If $u' = \cos(\phi)u + \sin(\phi)v$ and $v' = -\sin(\phi)u + \cos(\phi)v$, then (u', v') is obtained by rotating the point (u, v) clockwise about the origin ϕ radians. Thus, if we have points on the untilted ellipse $((x-h)/a)^2+((y-k)/b)^2 = 1$ then we can obtain points on the tilted ellipse by using these rotation formulae. Implement the following function:

```
  function ShowEllipse(Eta)
% Let Eta represent the ellipse that is obtained by rotating the set of points
% that satisfy
%
%      ((x - Eta.h)/Eta.a)^2  + ((y - Eta.k)/Eta.b)^2 = 1
%
% clockwise about the origin by an angle that has measure Eta.phi radians.
% This function plots the ellipse together with its two axes.
```

Be sure to set `axis equal`. Test your implementation of `ShowEllipse` by running the script `P1C` which is available on the website. Submit output and a listing of `ShowEllipse`.

Part D (5 pts) The Perimeter of an Ellipse

Given positive real numbers a and b, define the ellipse $\mathcal{E} = \{ (a\cos(\theta), b\sin(\theta)) : 0 \leq \theta \leq 2\pi \}$ This problem is about finding the perimeter of $\mathcal{E}$. Unlike for the circle where $C = 2\pi r$ there is no closed formula and we have to resort to approximation. Our plan is to approximate the ellipse with a pair of polygons, one that is inscribed and one that is circumscribed. Their computable perimeters will then give us lower and upper bounds for $\mathcal{E}$'s perimeter. Here is a picture when these polygons have $n = 8$ sides:



To formally specify the vertices of these polygons assume that n is an integer that satisfies $n \geq 3$. For $i = 1{:}n{+}1$ define the angle $\theta_i = 2\pi(i-1)/n$. Let P_i be the point

$$P_i = (a\cos(\theta_i), b\sin(\theta_i))$$

for $i = 1{:}n+1$. Connecting these points in order renders the *inner polygon* and we denote its perimeter by $C_{inner}(n)$. Let L_i be the tangent to $\mathcal{E}$ at P_i. These tangent lines define intersection points $(x_1, y_1), \ldots, (x_n, y_n)$ where

$$x_i = a\frac{\sin(\theta_i) - \sin(\theta_{i+1})}{\sin(\theta_i)\cos(\theta_{i+1}) - \cos(\theta_i)\sin(\theta_{i+1})} \qquad \textit{and} \qquad y_i = b\frac{\cos(\theta_{i+1}) - \cos(\theta_i)}{\sin(\theta_i)\cos(\theta_{i+1}) - \cos(\theta_i)\sin(\theta_{i+1})}$$

Connecting these points (the asterisks in the figure) in order renders the *outer polygon* and we denote its perimeter by $C_{outer}(n)$. Thus, if C is the exact perimeter, then $C_{inner}(n) < C < C_{outer}(n)$ and so

$$|C - C_{outer}(n)| \;\leq\; C_{outer}(n) - C_{inner}(n) \;\equiv\; \epsilon(n)$$

Using these ideas implement a function `[C_inner,C_outer] = Eperimeter(a,b,n)` that returns the value of the $C_{inner}(n)$ and $C_{outer}(n)$. Your implementation should be fully vectorized–no loops!

Write a script `P1D` that uses `Eperimeter` and produces a `semilogy` plot of $\epsilon(n)/C_{inner}(10000)$ for the case $a = 3$ and $b = 2$. This quotient is a pretty good measure of the relative error associated with C_{outer}. Base the plot on the values $n = 10{:}10{:}1000$. Display the value of $C_{inner}(10000)$ to six decimal places in the title and appropriately label the x and y axes. Submit listings of `P1D` and `Eperimeter` and a copy of the output.