The R package *Conics*

Version 0.3

Bernard Desgraupes
University Paris Ouest
Modal'X

Last update: 2013-11-16

Contents

---

1 Algebraic background

This section is a survey of the main results concerning the algebraic representation of a plane conic.

1.1 Notation

A conic $\mathcal{C}$ is a plane algebraic curve of degree 2. It is the set of zeroes of a polynomial of degree 2 in 2 variables, that is to say the set of points (x_1, x_2) satisfying an equation of the form

$$P(x_1, x_2) = a_1\,x_1^2 + a_2\,x_1x_2 + a_3\,x_2^2 + a_4\,x_1 + a_5\,x_2 + a_6 = 0. \tag{1}$$

Consider the following change of variables which introduces homogeneous coordinates (X_1, X_2, X_3):

$$x_1 = \frac{X_1}{X_3} \qquad x_2 = \frac{X_2}{X_3} \tag{2}$$

Provided $X_3 \neq 0$, the previous equation can be simplified :

$$Q(X_1, X_2, X_3) = a_1\,X_1^2 + a_2\,X_1X_2 + a_3\,X_2^2 + a_4\,X_1X_3 + a_5\,X_2X_3 + a_6\,X_3^2 = 0. \tag{3}$$

It appears that $Q(X_1, X_2, X_3)$ is quadratic form. This form can be represented by a 3×3 matrix A such that

$$Q(X) = {}^tXAX \tag{4}$$

The matrix is defined like this :

$$A = \begin{pmatrix} a_1 & \frac{1}{2}a_2 & \frac{1}{2}a_4 \\ \frac{1}{2}a_2 & a_3 & \frac{1}{2}a_5 \\ \frac{1}{2}a_4 & \frac{1}{2}a_5 & a_6 \end{pmatrix} \tag{5}$$

It is a symmetric matrix. Let $\Delta = \det(A)$. If $\Delta \neq 0$, the conic is said to be *regular* (or *non-degenerate*), otherwise it is *degenerate*. The same terminology applies to the quadratic form itself.

When a quadratic form is degenerate, it splits into the product of two polynomials of degree 1. Geometrically, it means that the conic is a pair of lines. On the contrary, if the quadratic form is non-degenerate, the conic is an ellipse, a hyperbola, or a parabola.

1.2 Classification

In order to decide which kind of conic is represented by the matrix A, one must consider the 2×2 top left submatrix, i-e the matrix B obtained by deleting the last row and the last column of A:

$$B = \begin{pmatrix} a_1 & \frac{1}{2}a_2 \\ \frac{1}{2}a_2 & a_3 \end{pmatrix} \tag{6}$$

The determinant of B is denoted δ. Its value is

$$\delta = a_1 a_3 - \frac{1}{4} a_2^2. \tag{7}$$

In the non-degenerate case, the matrix A has rank 3 and one has the following classification based on the value of δ:

- if $\delta > 0$, $\mathcal{C}$ is an ellipse
- if $\delta = 0$, $\mathcal{C}$ is a parabola
- if $\delta < 0$, $\mathcal{C}$ is a hyperbola

If the conic is degenerate, A has rank less than 3 and one has the following classification:

- if $\delta > 0$, $\mathcal{C}$ is empty
- if $\delta = 0$, $\mathcal{C}$ is a pair of parallel lines (possibly coincident)
- if $\delta < 0$, $\mathcal{C}$ is a pair of intersecting lines

In particular, the case of a double line (coincident parallel lines) occurs when A is of rank 1.

1.3 Points at infinity

Except in the case of an ellipse, all the conics have points at infinity. These points can be found by letting $X_3 \longrightarrow 0$ in equation (3). One obtains the following equation:

$$a_1 X_1^2 + a_2 X_1 X_2 + a_3 X_2^2 = 0$$

which can be rewritten in variables x_1 and x_2 like this

$$a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 = 0. \tag{8}$$

Let $t = \dfrac{x_2}{x_1}$. The variable t can be interpreted as the slope of the directions to infinity. The previous equation becomes, after division by x_1^2:

$$a_1 + a_2 t + a_3 t^2 = 0 \tag{9}$$

It is an ordinary equation of degree 2 which will have real solutions if its discriminant is non-negative :

$$D = a_2^2 - 4a_1 a_3 = -4\delta \geq 0 \tag{10}$$

So, if $\delta > 0$ (case of an ellipse), the discriminant is negative and there are no solutions: this is normal since an ellipse does not have points at infinity. If $\delta < 0$ (case of a hyperbola), one finds two distinct solutions which correspond to the slope of the asymptotes of the hyperbola. Finally, if $\delta = 0$ (case of a parabola), one finds a unique solution which is the asymptotic direction of the branches of the parabola.

1.4 Center

Some conics have a center C. In the center, the gradient of the quadratic polynomial P is null. This leads to the following equations:

$$\begin{cases} \dfrac{\partial P}{\partial x_1} = 0 \\ \\ \dfrac{\partial P}{\partial x_2} = 0 \end{cases} \tag{11}$$

The partial derivatives yield the following equations:

$$\begin{cases} a_1\, x_1 + \frac{1}{2} a_2\, x_2 + a_4 & = & 0 \\ \frac{1}{2} a_2\, x_1 + \frac{1}{2} a_2\, x_3 + a_5 & = & 0 \end{cases} \tag{12}$$

This is a system of two linear equations in two variables. Its matrix is B. If $\delta = \det(B) \neq 0$, it has a unique solution and the conic has a unique center. This is the case of an ellipse, or a hyperbola or a pair of intersecting lines.

1.5 Axes

The symmetry axes of a conic are lines passing through the center. Their direction vectors are the eigenvectors of the submatrix B defined by (6).

Since B is symmetric, one has the following properties:

- the eigenvalues λ_1 and λ_2 are real (not complex);
- the eigenvectors are real too;
- the matrix can always be diagonalized in an orthonormal basis. It means that one can always find two orthogonal eigenvectors with norm equal to 1. Let us denote V_1 and V_2 these two vectors.

As a consequence, a conic has in general two axes which are orthogonal.

The eigenvalues are the roots of the characteristic polynomial associated to matrix B. It is defined as

$$\begin{aligned} P(\lambda) &= \det(B - \lambda I) \\ &= \lambda^2 - \mathrm{Tr}(B)\lambda + \det(B) \\ &= \lambda^2 - (a_1 + a_3)\lambda + \delta \\ &= 0 \end{aligned} \tag{13}$$

If λ is an eigenvalue, the corresponding eigenvector V can be calculated by solving the following equation :

$$(B - \lambda I)V = 0 \tag{14}$$

In the particular case where $\lambda_1 = \lambda_2$, the eigenspace has dimension 2 which means that any direction is a possible eigenvector. This corresponds to a circle: in a circle indeed any diameter is a symmetry axis.

1.6 Reduced equation

In the case of a conic with a center (ellipse or hyperbola), one can change the coordinate system by translating the origin to the center C and by rotating the axes to vectors V_1 and V_2. In the $\{C, V_1, V_2\}$ basis, let us designate the coordinates by y_1 and y_2. The equation of the conic in this basis is remarkably simple :

$$\boxed{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \frac{\Delta}{\delta} = 0} \tag{15}$$

The relation between the (x_1, x_2) and (y_1, y_2) coordinates are given by the transformation matrix T whose columns are the eigenvectors V_1 and V_2. One has

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = T \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{16}$$

2 The *conics* package

The R package *conics* makes use of the previous results to plot conic curves. The package must be loaded with the *library* command like this:

```
> library(conics)
```

2.1 Basic functions

In the R *conics* package, conics can be specified either by a 6-length vector containing the coefficients of the polynomial in equation (1), or by the symmetric matrix A defined by equation (5).

There is a convenience function named **conicMatrix** which computes the matrix A given the vector of coefficients of the polynomial P defined by (1). Here is a simple example: let us consider the conic with equation

$$2\,x_1^2 + 2\,x_1 x_2 + 2\,x_2^2 - 20\,x_1 - 28\,x_2 + 10 = 0$$

The vector of coefficients is

```
> v <- c(2,2,2,-20,-28,10)
```

and the corresponding matrix can be obtained by the following instruction:

```
> A <- conicMatrix(v)
```

```
     [,1] [,2] [,3]
[1,]    2    1  -10
[2,]    1    2  -14
[3,]  -10  -14   10
```

The center and the axes of the conic can be calculated using the functions **conicCenter** and **conicAxes** respectively. For instance:

```
> conicCenter(v)
```

```
[1] 2 6
```

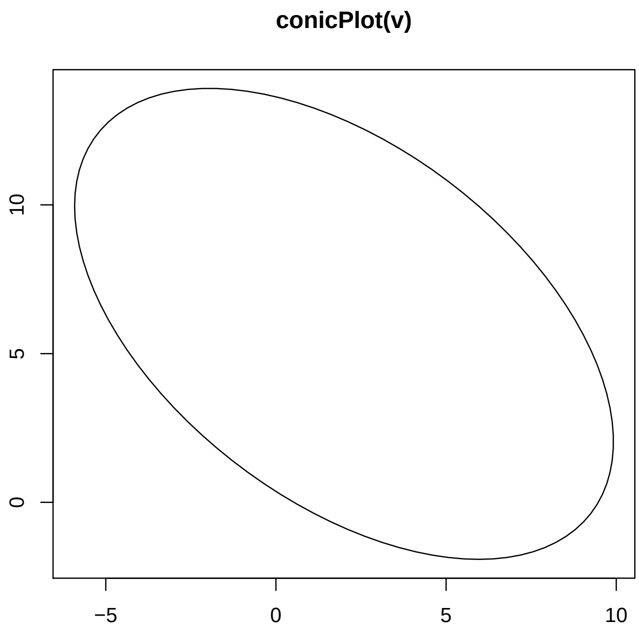
```
> conicAxes(v)

          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

Alternatively, one can specify the matrix instead of the vector:

```
> conicCenter(A)
> conicAxes(A)
```

Finally the conic can be plotted with the **conicPlot** function like this:

```
> conicPlot(v, main="conicPlot(v)", xlab="", ylab="")
```



2.2 Plotting parameters

The **conicPlot** function calculates a set of points on the conic and ultimately calls the usual *plot* function from the *graphics* package. Any of the numerous arguments defined with the *plot* function can be specified in the **conicPlot** function as well. For instance, in order to draw the previous ellipses in red with a dotted contour, one can write :

```
conicPlot(v, col="red", lty=3)
```

The **conicPlot** function also has a set of optional arguments of its own. Currently the following arguments are defined :

add is a boolean argument. If it is set to `TRUE`, the drawing is added to the current plot instead of erasing the current graphical device.

as.col specifies the color of the asymptotes. It can take the same values as the *col* argument of the *plot* function.

as.lty specifies the line type for the asymptotes. It can take the same values as the *lty* argument of the *plot* function.

ax.col specifies the color of the axes. It can take the same values as the *col* argument of the *plot* function.

ax.lty specifies the line type for the axes. It can take the same values as the *lty* argument of the *plot* function.

asymptotes is a boolean argument. If it is set to `TRUE`, the asymptotes will be drawn. This argument is meaningful only in the case of a hyperbola.

center is a boolean argument. If it is set to `TRUE`, the center will be marked by a small circle.

npoints is a numeric argument indicating the number of points to calculate in order to draw the curve. The default value is 100.

sym.axes is a boolean argument. If it is set to `TRUE`, the symmetry axes will be drawn.

... any other arguments will be passed verbatim to the basic *plot* function from the *graphics* package. See the documentation of the *par* function to know which arguments are supported. In particular, the following two arguments are very useful:

xlim is a 2-elements numeric vector specifying the range of the x-coordinate.

ylim is a 2-elements numeric vector specifying the range of the y-coordinate.

2.3 Aspect ratio

In order to avoid distorsions due to the difference of units between the x-axis and the y-axis, the *asp* argument can be very useful. It is defined as the ratio in length between one data unit in the y direction and one data unit in the x direction.

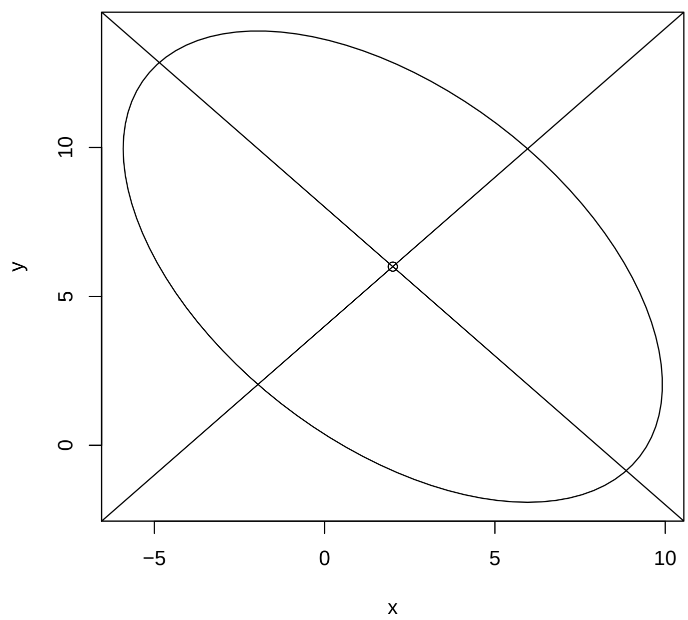
Setting $asp = 1$ will ensure that the same unit length is used for both coordinate axes so that distances between points are represented accurately. For instance:

```
conicPlot(v, asp=1)
```

2.4 Examples

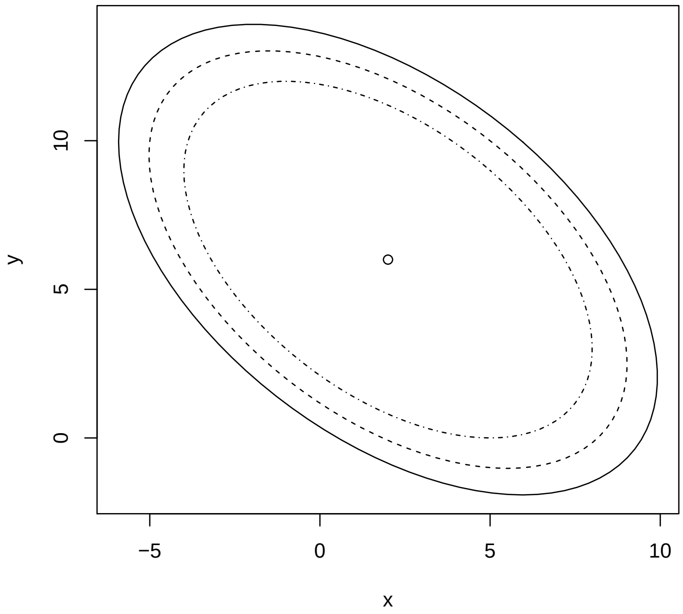
Here is an example using the previous vector and demonstrating the *center* and the *sym.axes* parameters :

```
> conicPlot(v, center=T, sym.axes=T)
```



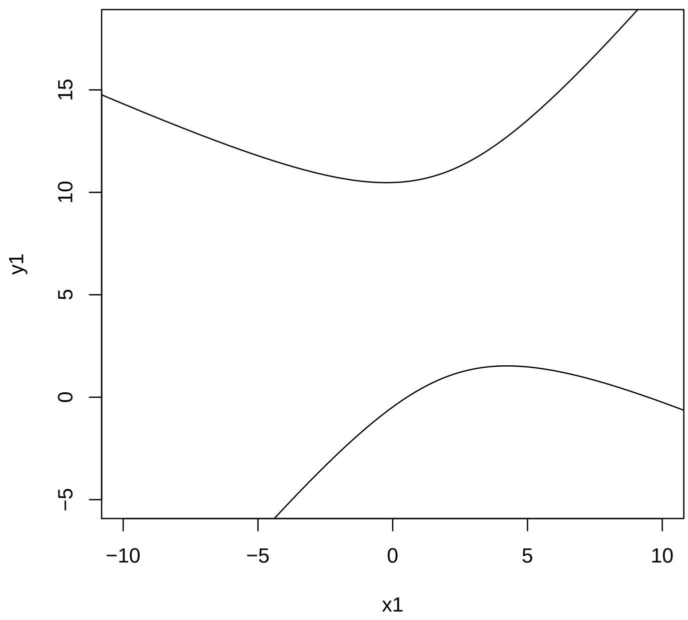
Here is another example where several ellipses are drawn on the same plot using the *add* parameter :

```
> v <- c(2,2,2,-20,-28,10)
> conicPlot(v, center=T, lty=1)
> v[6] <- 30
> conicPlot(v, add=T, lty=2)
> v[6] <- 50
> conicPlot(v, add=T, lty=4)
```



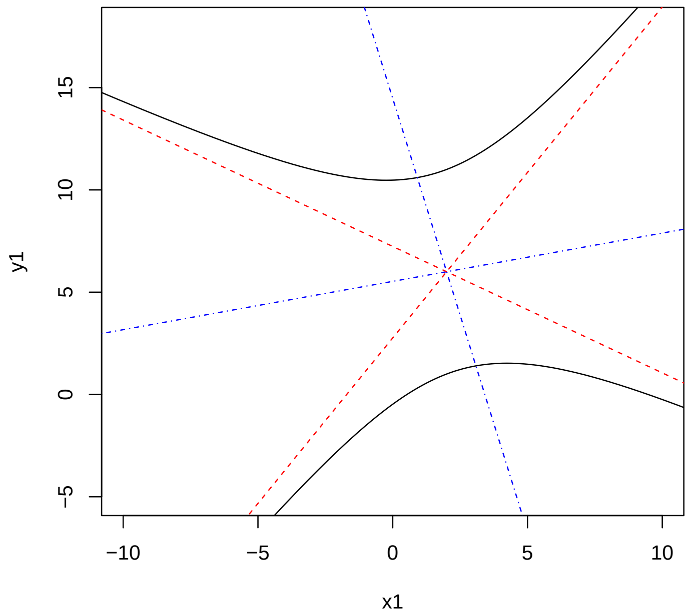
Here is now an example of a hyperbola making use of the *xlim* and *ylim* parameters :

```
> v <- c(2,2,-2,-20,20,10)
> conicPlot(v, xlim=c(-10,10), ylim=c(-5,18))
```



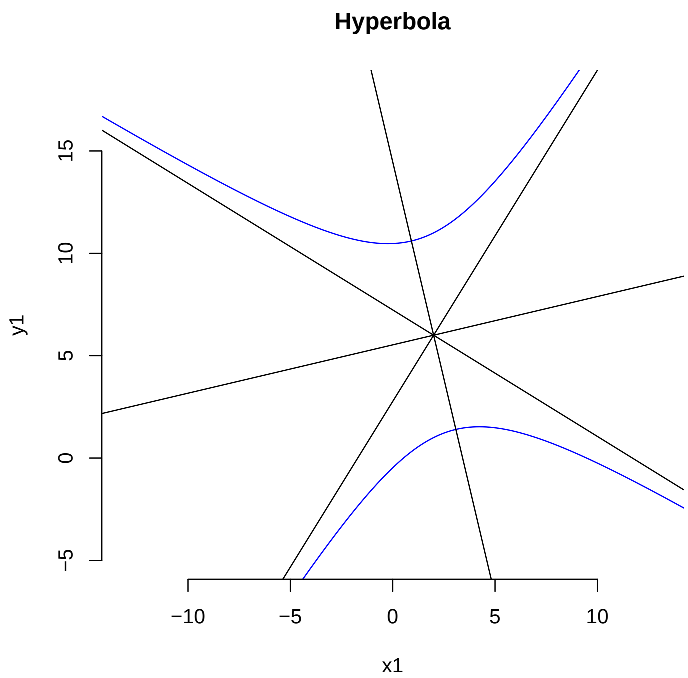
Here is an example with the same hyperbola demonstrating the *as.col*, *as.lty*, *ax.col*, and *ax.lty* options :

```
> conicPlot(v, asymptotes=T, sym.axes=T,
+ as.col="red", as.lty=2, ax.col="blue", ax.lty=4,
+ xlim=c(-10,10), ylim=c(-5,18))
```



Here is an example of extra arguments which are ultimately passed to the *plot* function :

```
> conicPlot(v, asymptotes=T, sym.axes=T,
+ xlim=c(-10,10), ylim=c(-5,18),
+ asp=1, col="blue", main="Hyperbola", bty="n")
```



The *asp* argument (aspect ratio) is set to 1 to ensure accurate distances between points. The *col* argument sets the color of the conic itself. The *main* argument adds a title to the plot. The *bty* argument set to ”n” suppresses the box around the plot.

2.5 Return value

The return value of the **conicPlot** function is invisible, i-e it is not printed on the console by default but it can be stored in a variable in order to get its contents.

It is a list of computed values corresponding to various elements of the conic. Some of the following elements can be found in the return list, depending on the kind of the conic:

kind the kind of the conic. It is a character string whose value is *"ellipse"*, *"hyperbola"*, *"parabola"*, or *"lines"* ;

axes the symmetry axes. This value is a 2×2 matrix whose columns are direction vectors of the axes ;

center the center of the conic ;

asymptotes the slopes of the asymptotes ;

vertices the vertices of the conic ;

foci the focal points of the conic ;

eccentricity the eccentricity of the conic. It is a value between 0 and 1 for an ellipse, equal to 1 for a parabola and greater than 1 for an hyperbola ;

intercepts the intercepts in the case of parallel lines.

points the coordinates of the points used to plot the conic. This value is returned only if the *type* option is equal to 'n' and if the conic is non-degenerate. In that case, nothing is drawn.

Here is an example

```
> v <- c(2,2,-2,-20,20,10)

[1]   2   2  -2 -20  20  10

> res <- conicPlot(v)

$kind
[1] "hyperbola"

$axes
           [,1]       [,2]
[1,] -0.9732490  0.2297529
[2,] -0.2297529 -0.9732490

$center
[1] 2 6

$asymptotes
[1]  1.618034 -0.618034

$vertices
```

```
$vertices$x
[1] 3.0864345 0.9135655

$vertices$y
[1]  1.39779 10.60221


$foci
$foci$x
[1] 3.5364504 0.4635496

$foci$y
[1] -0.5085083 12.5085083


$eccentricity
[1] 1.414214
```

In the next graphic, the vertices and the foci are drawn like this :

```
> v <- c(-4,0,1,0,0,1)
> cp <- conicPlot(v, sym.axes=TRUE, asymptote=TRUE, asp=1,
+                     ax.lty=2, as.col="gray")
> points(cp$foci$x,cp$foci$y,col="red",pch=19)
> text(cp$foci$x,cp$foci$y+0.1,paste("F",2:1,sep=""))
> points(cp$vertices$x,cp$vertices$y,col="blue",pch=19)
```