Rappresentazione parametrica di curve piane

Matematica Open Source `http://www.extrabyte.info`

Sia data una rappresentazione parametrica di una curva piana γ:

$x = x(t)$, $y = y(t)$ $t \in [a,b]$, (1)

dove (x,y) sono le usuali coordinate cartesiane nel piano, mentre t è il parametro della rappresentazione. Come è noto, se le funzioni $x(t)$ e $y(t)$ sono tali da poter eliminare nelle (1) il parametro t, è possibile passare alla rappresentazione cartesiana

$y=f(x)$, $x \in X$

Il caso più semplice è quello in cui le funzioni $x(t)$ e $y(t)$ sono lineari. Ad esempio:

$x(t) = x_0 + \lambda t$, $y(t) = y_0 + \mu t$ (2)

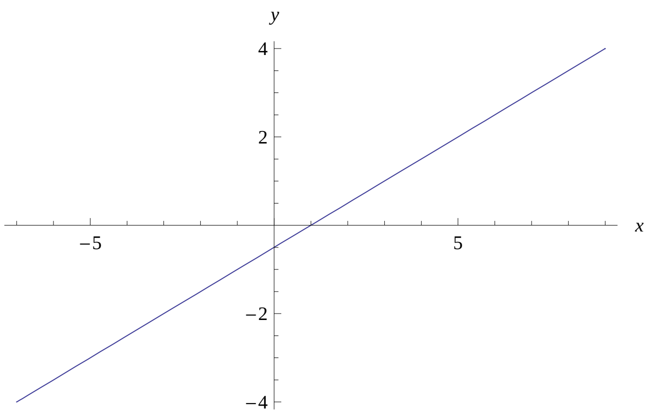
per cui eliminando t

$y = \frac{\mu}{\lambda}(x - x_0) + y_0$ (3)

che è l'equazione di una retta passante per (x_0, y_0) e di coeffiente angolare $\frac{\mu}{\lambda}$. Tuttavia, nella maggior parte dei casi non è possibile ricavare l'equazione cartesiana della curva. *Mathematica* dispone del comando **`ParametricPlot`** che permette di tracciare curve piane conoscendo la sola rappresentazione parametrica (lo stesso comando può essere utilizzato per tracciare una superficie di cui è nota la rappresentazione parametrica). Nel caso (2), definiamo le funzioni:

```
x[t_, x0_, λ_] := x0 + λ * t; y[t_, y0_, μ_] := y0 + μ * t
```

```
ParametricPlot[
 {x[t, 1, 2], y[t, 0, 1]}, {t, -4, 4},
 AxesLabel → {"x", "y"},
 AspectRatio → 0.6
]
```



Consideriamo ora:

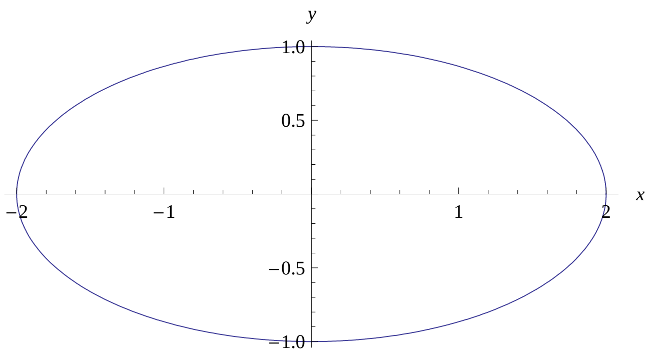
$x = 2\cos(t)$, $y = \sin(t)$, $t \in [0, 2\pi]$ (4)

In realtà è possibile eliminare il parametro, ottenendo

$x = 2\sqrt{1 - y^2}$

Proviamo comunque a plottare parametricamente:

```
ParametricPlot[
 {2 Cos[t], Sin[t]}, {t, 0, 2 Pi},
 AxesLabel → {"x", "y"}
]
```



Il comando **RotationTransform** permette di eseguire una rotazione nel piano cartesiano $x\,y$ di un angolo θ. Come è noto, la matrice di rotazione è data da:

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Ad esempio, ruotando il vettore (0, 1):

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix}$$

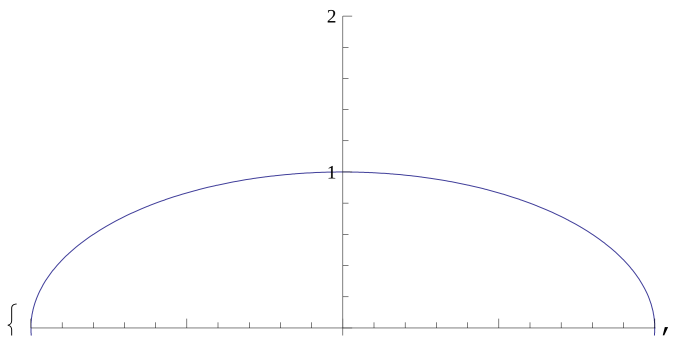
Tali operazioni sono incorporate in **RotationTransform**

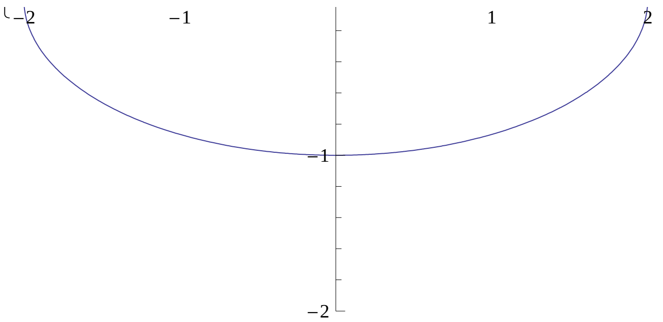
```
RotationTransform[θ][{0, 1}]
```

```
{-Sin[θ], Cos[θ]}
```

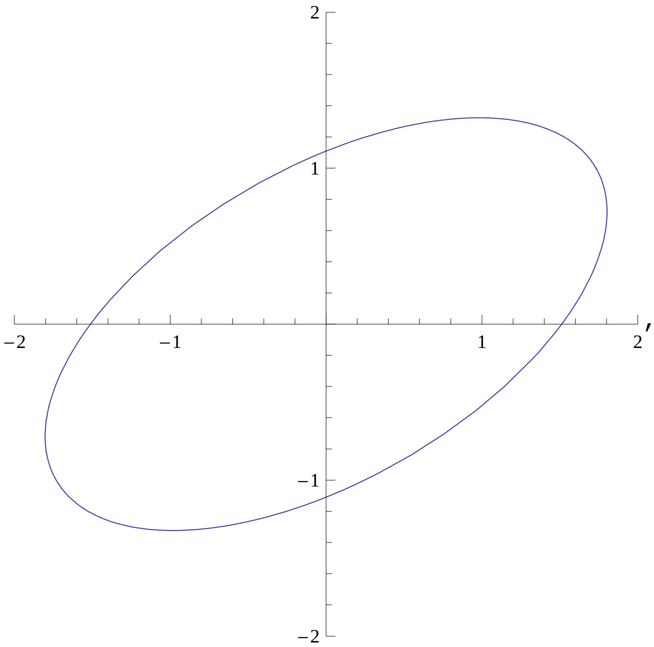
Ci si può chiedere se sia possibile ruotare un grafico parametrico di un angolo assegnato utilizzando **RotationTransform**. La risposta è affermativa. Ad esempio:

```
Table[
 ParametricPlot[
  Evaluate@RotationTransform[θ][{2 Cos[u], Sin[u]}], {u, 0, 2 Pi},
  PlotRange → 2
 ],
 {θ, {0, Pi / 6, Pi / 3, Pi / 2}}
]
```

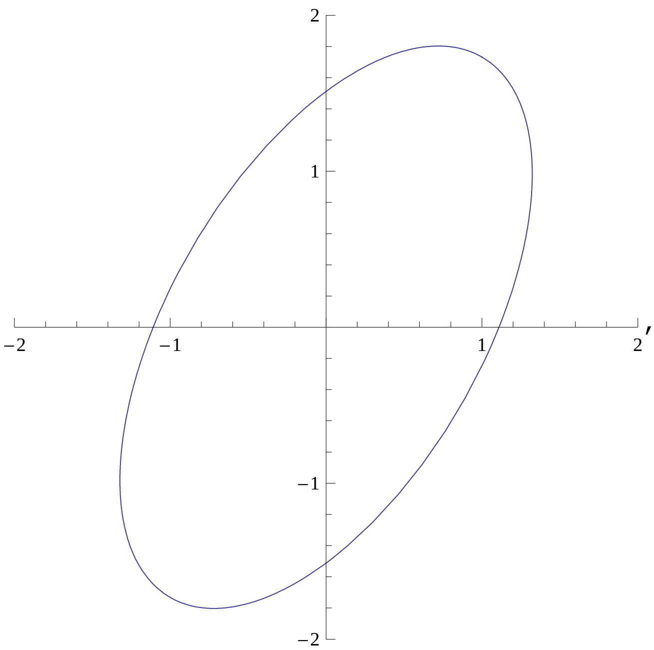




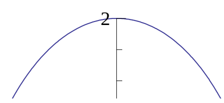
−2
−1
1
2
−1
−2



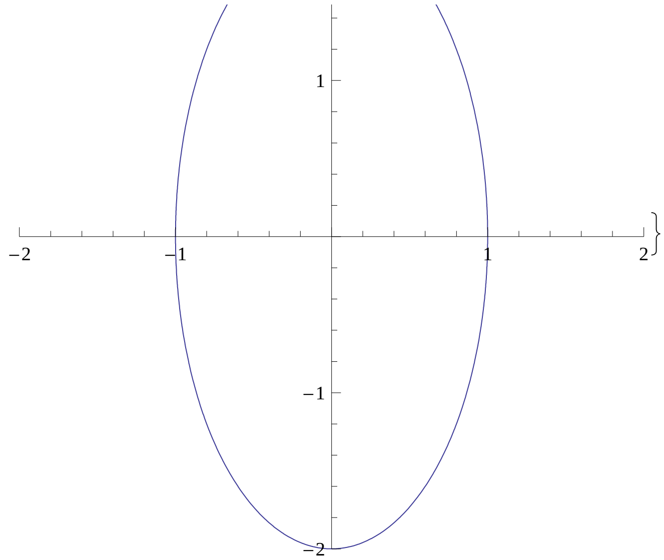
2
1
−2
−1
1
2
−1
−2



2
1
−2
−1
1
2
−1
−2



2



Tali risultati possono essere inglobati in un unico grafico:

```
ParametricPlot[Evaluate@RotationTransform[θ][{2 Cos[t], Sin[t]}],
 {t, 0, 2 Pi}, {θ, 0, Pi / 2}]
```