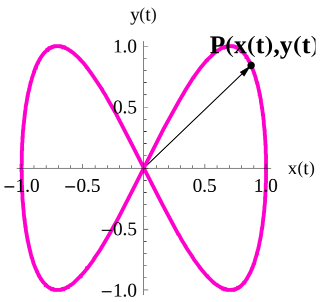
4

REPRESENTACIÓN DE CURVAS EN FORMA PARAMÉTRICA

4.1. Parametrización de curvas en el plano

Una curva en forma paramétrica es la representación gráfica, en un sistema de ejes coordenados rectangulares OXY, de los pares (x(t),y(t)) en el plano cartesiano OXY , donde t pertenece al dominio de x(t) y de y(t).



▼ ParametricPlot[ ]

```
? ParametricPlot
```

ParametricPlot[$\{f_x, f_y\}, \{u, u_{min}, u_{max}\}$] generates a parametric plot of a curve with x and y coordinates f_x and f_y as a function of u.

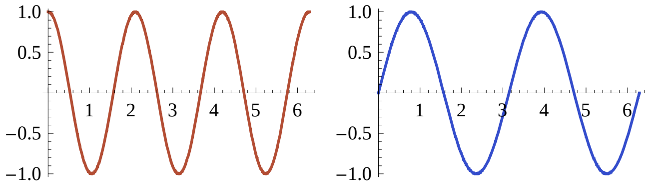
ParametricPlot[$\{\{f_x, f_y\}, \{g_x, g_y\}, \ldots\}, \{u, u_{min}, u_{max}\}$] plots several parametric curves.

ParametricPlot[$\{f_x, f_y\}, \{u, u_{min}, u_{max}\}, \{v, v_{min}, v_{max}\}$] plots a parametric region.

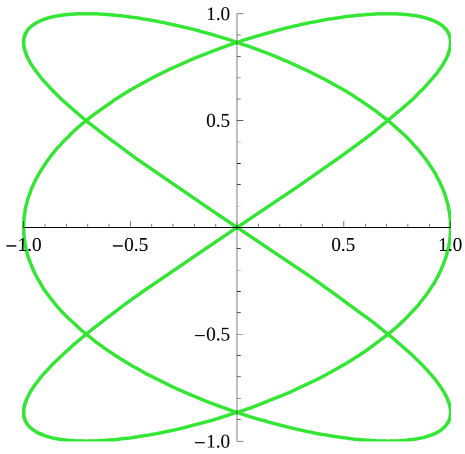
ParametricPlot[$\{\{f_x, f_y\}, \{g_x, g_y\}, \ldots\}, \{u, u_{min}, u_{max}\}, \{v, v_{min}, v_{max}\}$] plots several parametric regions. ≫

```
r[t_] = {x[t_], y[t_]} = {Cos[3 * t], Sin[2 * t]};
```

```
GraphicsArray[
 {Plot[x[t], {t, 0, 2*Pi}, PlotStyle → {RGBColor[0.7, 0.3, 0.2], Thickness[0.01]}],
  Plot[y[t], {t, 0, 2*Pi}, PlotStyle → {RGBColor[0.2, 0.3, 0.8], Thickness[0.01]}]}]
```



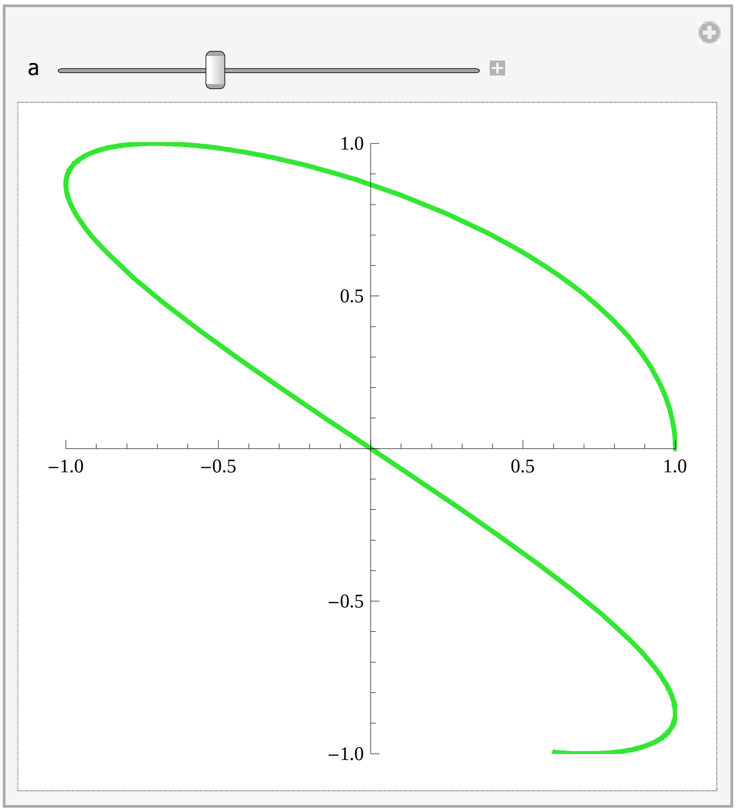
```
ParametricPlot[{Cos[3*t], Sin[2*t]}, {t, 0, 2*Pi},
 PlotStyle → {RGBColor[0.2, 0.9, 0.2], Thickness[0.008]}, PlotRange → {{-1, 1}, {-1, 1}}]
```



▼ Manipulate[ ]

Con este comando podemos ver como se dibuja la gráfica cuando recorremos el parámetro t.

```
Manipulate[ParametricPlot[{Cos[3 * t], Sin[2 * t]},
  {t, 0, a}, PlotStyle → {RGBColor[0.2, 0.9, 0.2], Thickness[0.008]},
  PlotRange → {{-1, 1}, {-1, 1}}], {a, 0.2, 2 * Pi}]
```



4.2. Parametrización de curvas dadas en forma explícita

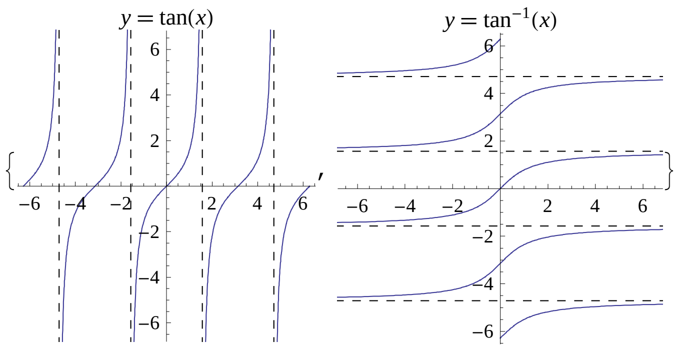
Dada una función y = f (x) definida de forma explícita , siempre se puede parametrizar en la forma

x (t) =x(t)

y (t) = f (x(t))

★ **Ejemplo 1**

```
{ParametricPlot[{u, Tan[u]}, {u, -2 Pi, 2 Pi},
  ExclusionsStyle → Dashed, Exclusions → {Cos[u] == 0}, PlotLabel → y == Tan[x]],
 ParametricPlot[{Tan[u], u}, {u, -2 Pi, 2 Pi}, ExclusionsStyle → Dashed,
  Exclusions → {Cos[u] == 0}, PlotLabel → y == ArcTan[x]]}
```



4.3. Parametrización de curvas dadas en forma implícita

Dada una curva definida de forma implícita , siempre se puede parametrizar en la forma

x (t) =x(t)

y (t)=y , solución de la ecuación f (x(t),y)=0

▼ Parametrización de una circunferencia de centro (a,b) y radio r

```
cir = (x - a) ^2 + (y - b) ^2 == r^2
```

$(-a + x)^2 + (-b + y)^2 == r^2$

```
x[t_] = a + r * Cos[t]
```

a + r Cos[t]

```
Solve[cir, y] /. x → x[t] // Simplify
```

$\{\{y \to b - \sqrt{r^2 \, \text{Sin}[t]^2}\}, \{y \to b + \sqrt{r^2 \, \text{Sin}[t]^2}\}\}$

```
circulo[t_, a_, b_, r_] = {x[t_], y[t_]} = {a + r * Sin[t], b + r * Cos[t]}
```

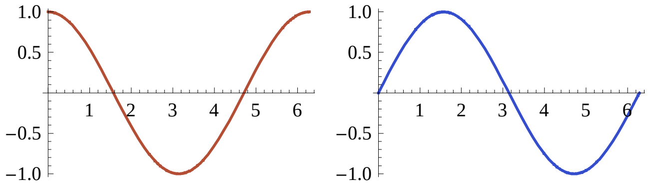
{a + r Sin[t], b + r Cos[t]}

▼ Parametrización de circunferencias con centro en el origen y radio r

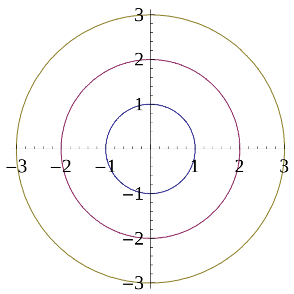
★ **Primera parametrización con orientación positiva**

```
r[t_] = {x[t_], y[t_]} = {Cos[t], Sin[t]};
```

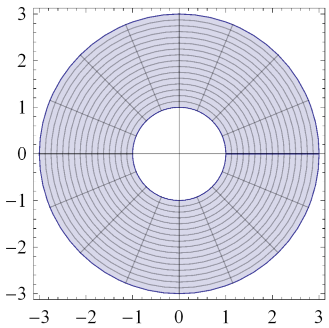
```
GraphicsArray[
 {Plot[Cos[t], {t, 0, 2 * Pi}, PlotStyle → {RGBColor[0.7, 0.3, 0.2], Thickness[0.01]}],
  Plot[Sin[t], {t, 0, 2 * Pi}, PlotStyle → {RGBColor[0.2, 0.3, 0.8], Thickness[0.01]}]}]
```

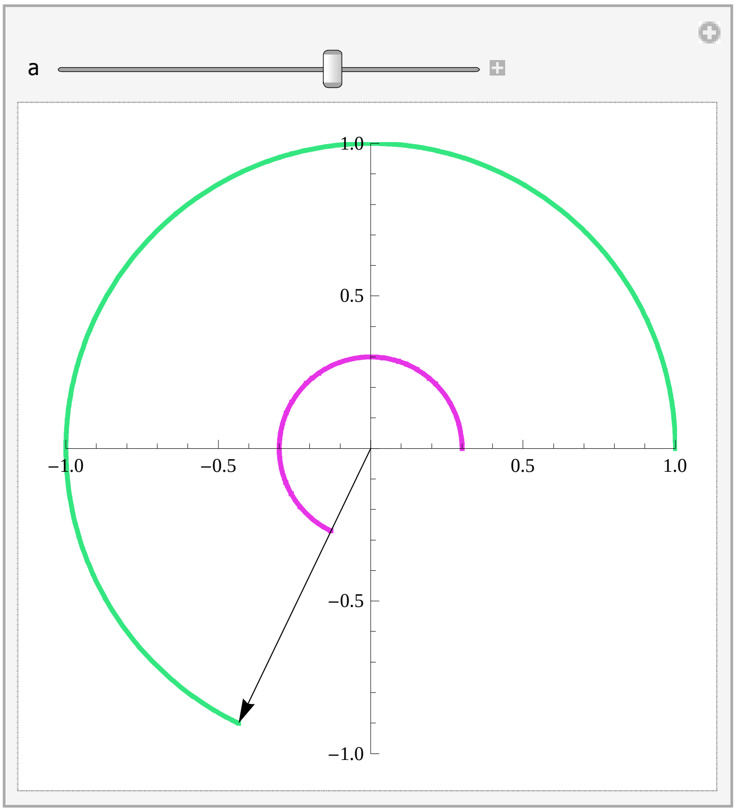


```
ParametricPlot[Evaluate[Table[{i Cos[u], i Sin[u]}, {i, 1, 3}]], {u, 0, 2 Pi}]
```



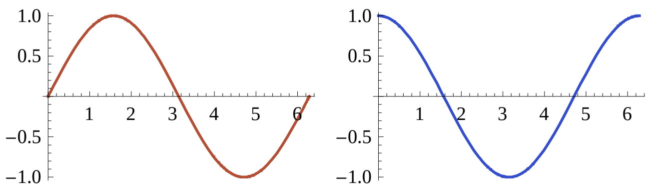
```
ParametricPlot[{i Cos[u], i Sin[u]}, {i, 1, 3}, {u, 0, 2 Pi}]
```

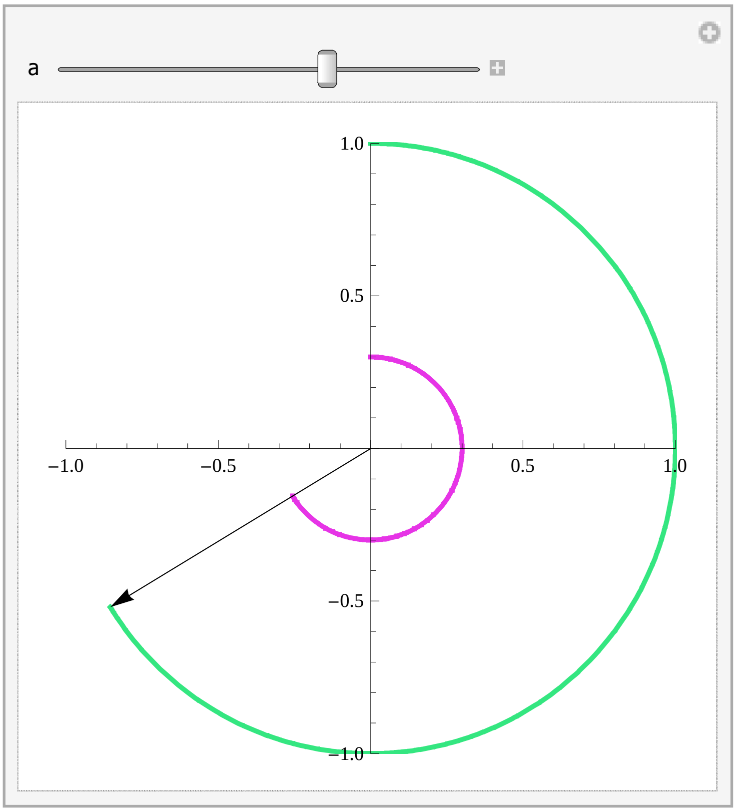




★ **Segunda parametrización: en el sentido de las agujas del reloj**

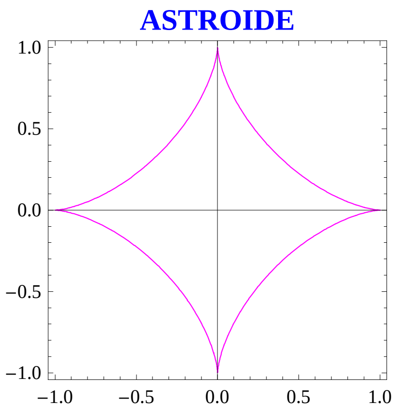
```
r[t_] = {x[t_], y[t_]} = {Cos[t], Sin[2 t]};

GraphicsArray[
 {Plot[Sin[t], {t, 0, 2 * Pi}, PlotStyle → {RGBColor[0.7, 0.3, 0.2], Thickness[0.01]}],
  Plot[Cos[t], {t, 0, 2 * Pi}, PlotStyle → {RGBColor[0.2, 0.3, 0.8], Thickness[0.01]}]}]
```

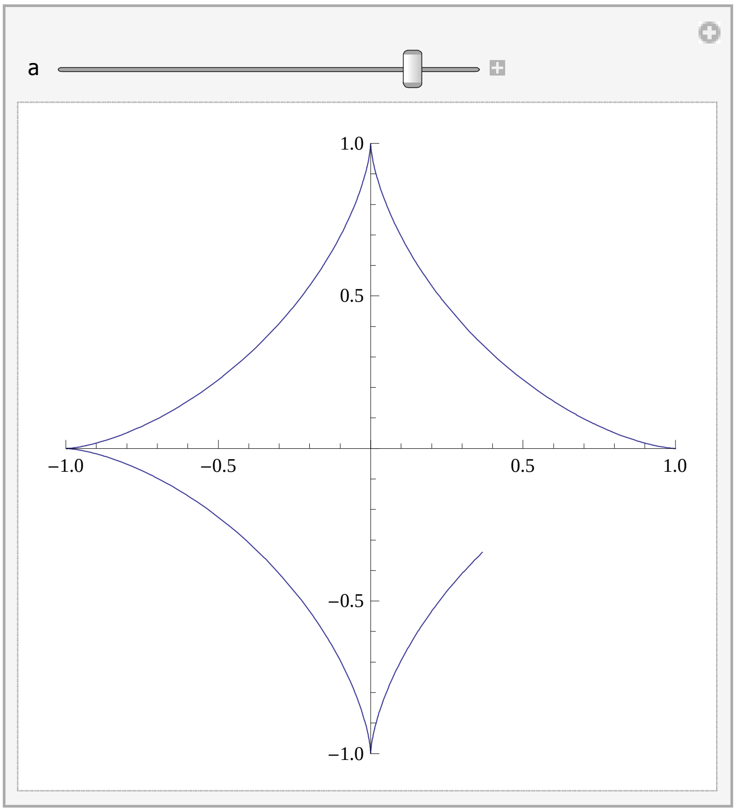




▼ Astroide

```
Clear[a]

astroide[t_, a_] = {a * Cos[t]^3, a * Sin[t]^3};

ParametricPlot[{Cos[t]^3, Sin[t]^3}, {t, 0, 2 π}, AspectRatio → Automatic,
 PlotStyle → Flatten[Table[RGBColor[a, 0, c], {a, 0, 1}, {c, 0, 1}]],
 PlotLabel → Style["ASTROIDE", Bold, Blue, 14], Frame → True]
```



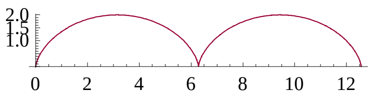


Cicloide

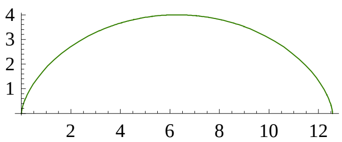
```
cicloide[t_, a_] = a * {t - Sin[t], 1 - Cos[t]}
```

```
{a (t - Sin[t]), a (1 - Cos[t])}
```

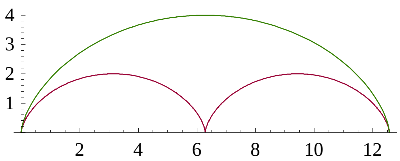
```
c1 = ParametricPlot[{cicloide[t, 1]}, {t, 0, 4 * Pi}, PlotStyle → RGBColor[0.6, 0, 0.2]]
```




```
c2 = ParametricPlot[{cicloide[t, 2]}, {t, 0, 2 * Pi}, PlotStyle → RGBColor[0.2, 0.5, 0]]
```



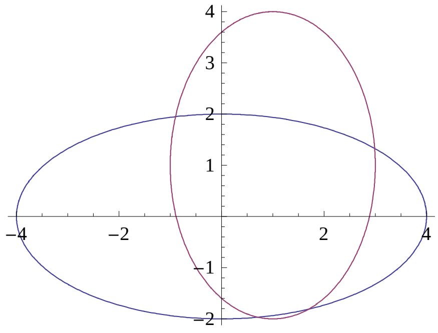

```
Show[{c1, c2}, PlotRange → {0, 4}]
```



Elipse

```
elipse[t_, a_, b_, c_, d_] = {a * Sin[t], b * Cos[t]} + {c, d};
```

```
ParametricPlot[Evaluate[{elipse[t, 4, 2, 0, 0], elipse[t, 2, 3, 0, 0] + {1, 1}},
  {t, 0, 2 Pi}, AspectRatio → Automatic]]
```



4.4. Parametrización de curvas dadas en forma polar

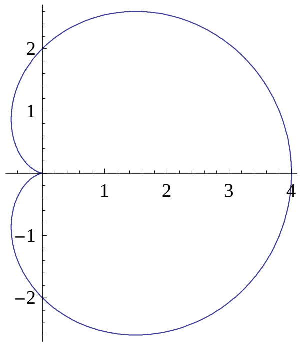
Si la curva viene dada en forma polar por la función r=r(t), se puede escribir en forma paramétrica como

x (t) =r(t) cost

y (t) =r(t) sent

▼ Elipse

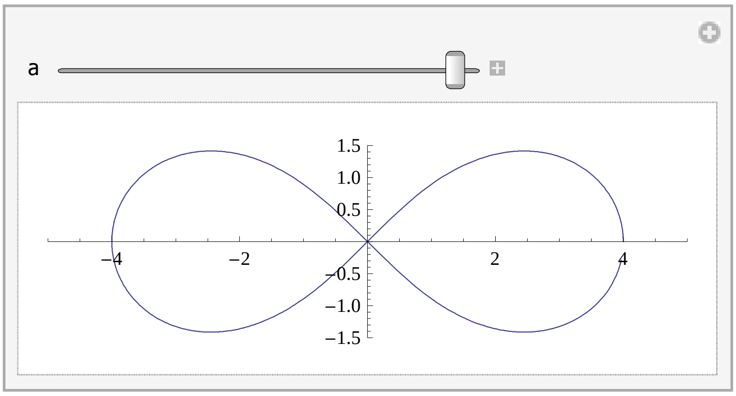
```
Clear["Global`*"]

cardioide[t_, a_] = {a * Cos[t] * (1 + Cos[t]), a * Sin[t] (1 + Cos[t])};

ParametricPlot[cardioide[t, 2], {t, 0, 2 Pi}]
```



▼ Lemniscata

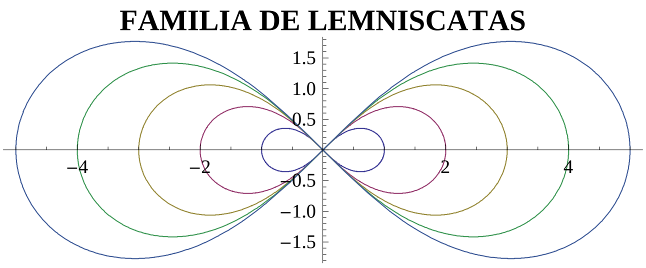
```
lemniscata[t_, a_] = {a * Cos[t] / (1 + Sin[t]^2), a * Sin[t] * Cos[t] / (1 + Sin[t]^2)};
```

```
Manipulate[
 ParametricPlot[{4 * Cos[t] / (1 + Sin[t]^2), 4 * Sin[t] * Cos[t] / (1 + Sin[t]^2)}, {t, 0, a},
  AspectRatio → Automatic, PlotRange → {{-5, 5}, {-1.5, 1.5}}], {a, 0.01, 2 * Pi, 0.1}]
```




```
lemniscata[t_, a_] = {a * Cos[t] / (1 + Sin[t]^2), a * Sin[t] * Cos[t] / (1 + Sin[t]^2)};
```

```
ParametricPlot[Evaluate[Table[lemniscata[t, a], {a, 1, 5}]], {t, 0, 2 π},
 AspectRatio → Automatic, PlotLabel → Style["FAMILIA DE LEMNISCATAS", Bold, 14]]
```

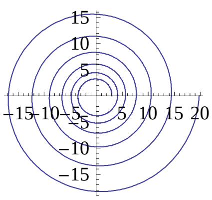


▼ Espiral Logarítmica

```
espiralog[t_, a_, b_] = {a * E^(b * t) * Cos[t], a * E^(b * t) * Sin[t]}
```

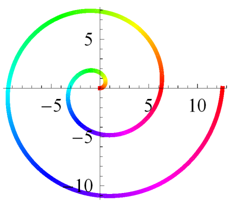
$\{a\, e^{b\,t}\, \text{Cos}[t],\ a\, e^{b\,t}\, \text{Sin}[t]\}$

```
ParametricPlot[espiralog[t, 3, 0.05], {t, 0, 12 Pi}, AspectRatio → Automatic]
```

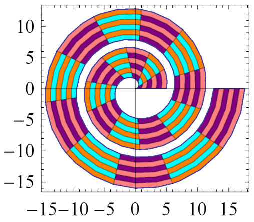


▼ Espiral de Arquímedes

```
ParametricPlot[{u Cos[u], u Sin[u]}, {u, 0, 4 Pi}, PlotStyle → Thick,
 ColorFunction → Function[{x, y, u, v}, Hue[u / (2 Pi)]], ColorFunctionScaling → False]
```



```
ParametricPlot[{(v + u) Cos[u], (v + u) Sin[u]}, {u, 0, 4 Pi},
 {v, 0, 5}, Mesh → {20, 5}, MeshShading → {{Purple, Cyan}, {Pink, Orange}}]
```



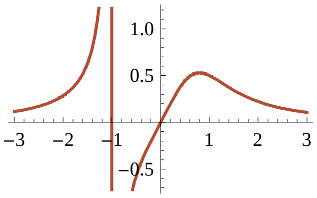
4.5. Curvas con ramas infifnitas

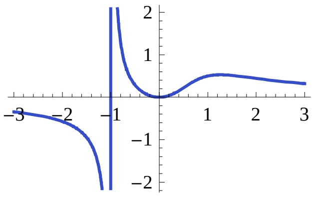
▼ Folium de Descartes

★ Definimos la parametrización

$$r[t\_] = \{x[t\_], y[t\_]\} = \left\{\frac{t}{1+t^3}, \frac{t^2}{1+t^3}\right\};$$

```
GraphicsArray[
 {Plot[x[t], {t, -3, 3}, PlotStyle → {RGBColor[0.7, 0.3, 0.2], Thickness[0.01]}],
  Plot[y[t], {t, -3, 3}, PlotStyle → {RGBColor[0.2, 0.3, 0.8], Thickness[0.01]}]}]
```





★ Analizamos los puntos de corte y las ramas infinitas

Puntos de corte

```
Solve[x[t] == 0, t]
```

{{t → 0}}

```
Solve[y[t] == 0, t]
```

{{t → 0}, {t → 0}}

Ramas Infinitas

```
to = -1;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

$-\infty$

∞

```
to = -∞;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

0

0

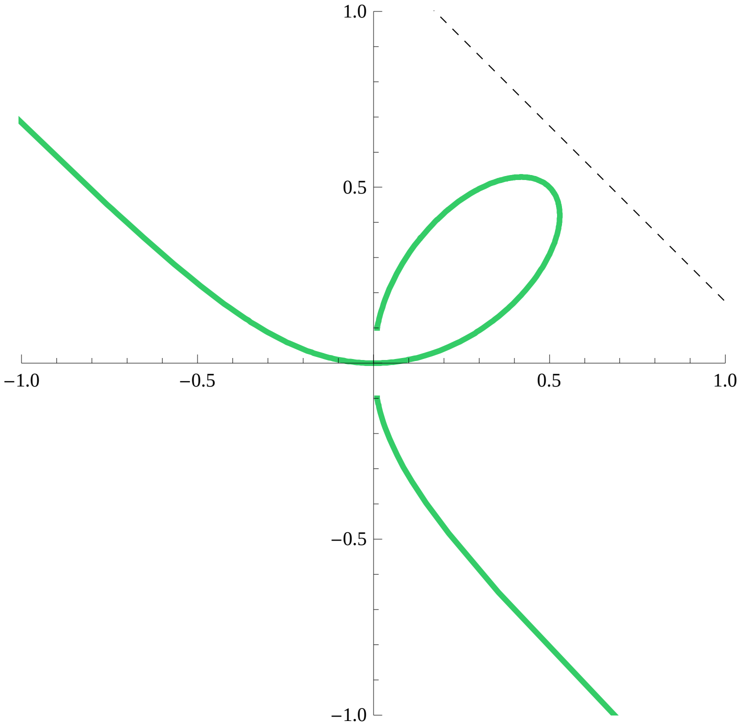
```
to = ∞;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

0

0

★ Gráfica en coordenadas paramétricas

$$\text{ParametricPlot}\left[\left\{\frac{t}{1+t^3}, \frac{t^2}{1+t^3}\right\}, \{t, -10, 10\},\right.$$
$$\text{ExclusionsStyle} \to \text{Dashed}, \text{Exclusions} \to \{1+t^3 == 0\},$$
$$\left.\text{PlotStyle} \to \{\text{RGBColor}[0.2, 0.8, 0.4], \text{Thickness}[0.008]\}, \text{PlotRange} \to \{\{-1, 1\}, \{-1, 1\}\}\right]$$

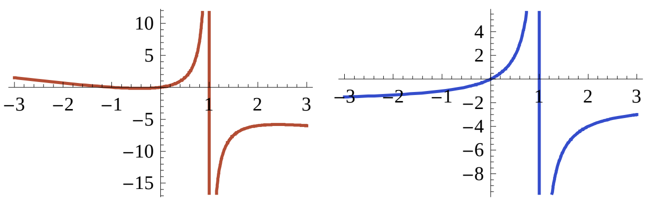


▼ Función paramétrica con asíntota

★ Definimos la parametrización

$$r[t\_] = \{x[t\_], y[t\_]\} = \left\{\frac{t^2+t}{1-t}, \frac{2*t}{1-t}\right\};$$

```
GraphicsArray[
 {Plot[x[t], {t, -3, 3}, PlotStyle → {RGBColor[0.7, 0.3, 0.2], Thickness[0.01]}],
  Plot[y[t], {t, -3, 3}, PlotStyle → {RGBColor[0.2, 0.3, 0.8], Thickness[0.01]}]}]
```



★ Analizamos los puntos de corte y las ramas infinitas

Puntos de corte

```
Solve[x[t] == 0, t]
```

{{t → -1}, {t → 0}}

```
Solve[y[t] == 0, t]
```

{{t → 0}}

Ramas Infinitas

```
to = 1;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

-∞

-∞

```
to = -∞;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

∞

-2

```
to = ∞;
Limit[x[t], t → to]
Limit[y[t], t → to]
```

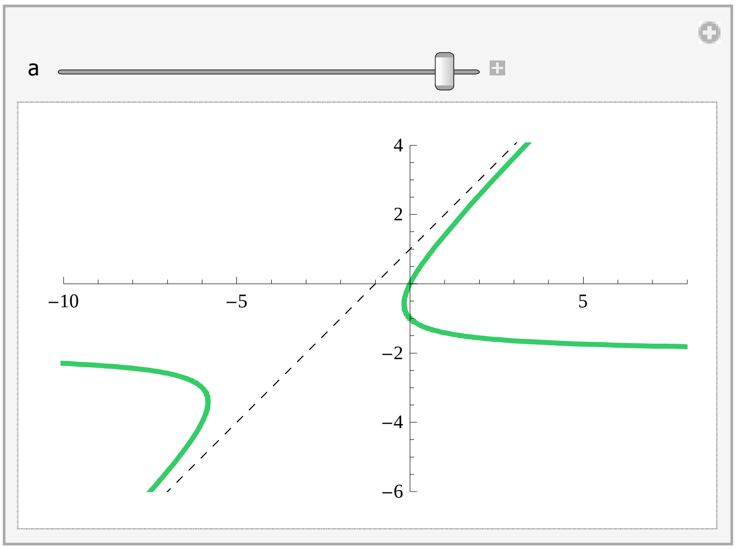
-∞

-2

★ Gráfica en coordenadas paramétricas

```
ParametricPlot[{(t^2 + t)/(1 - t), (2 * t)/(1 - t)}, {t, -10, 10}, ExclusionsStyle → Dashed,
  Exclusions → {t == 1}, PlotStyle → {RGBColor[0.2, 0.8, 0.4], Thickness[0.008]},
  PlotRange → {{-10, 8}, {-6, 4}}];
```

```
Manipulate[ParametricPlot[{(t^2 + t)/(1 - t), (2 * t)/(1 - t)}, {t, -10, a}, ExclusionsStyle → Dashed,
  Exclusions → {t == 1}, PlotStyle → {RGBColor[0.2, 0.8, 0.4], Thickness[0.008]},
  PlotRange → {{-10, 8}, {-6, 4}}], {a, -9.95, 10, 0.05}]
```



4.6. Curvas parametrizadas en 3D

▼ ParametricPlot3D

Helicoide

```
ParametricPlot3D[{Sin[u], Cos[u], u / 10}, {u, 0, 20},
 PlotStyle → Directive[Red, Thick], ColorFunction → "DarkRainbow"]
```