

Plotagens 2d (by Marcelo Pedro)

O Sage possui varios metodos de plotagem. Vamos Explorar alguns deles.

In []:

```
plot?
```

In []:

```
plot(x^2)
```

In []:

```
plot(x^2,(x,0,2))
```

In []:

```
plot(x^2,(x,0,2))+plot(x^3,(x,0,2),color='red')+plot(x^4,(x,0,2),color='yellow')
```

In []:

```
plot(x^2,(x,0,2))+plot(x^3,(x,0,2),color='red')+plot(x^4,(x,0,2),color='yellow')
```

In []:

```
plot(x^2,(x,0,2),legend_label='$f(x)=x^2$',title='Grafico da Funcao Quadratica')
```

Podemo criar um acumulador de graficos.

In []:

```
G=Graphics()
for n in range(1,10):
    G = G + plot(x^n, (x, -1,1),color=hue(n/10))
```

In []:

```
G
```

Exercício

Plote na mesma figura o grafico do seno na cor vermelho e o grafico do cosseno na cor verde.

Coordenadas Parametricas

In []:

```
t=var('t')
parametric_plot((e^t*cos(2*pi*t),e^t*sin(2*pi*t)),(t,0,4)) ###Espiral Logaritmica
```

Coordenadas Polares

In []:

```
##r(t) onde t representa o angulo polar
polar_plot(e^(t/(2*pi)), (t, 0, 8*pi)) ##Mesma Curva Anterior em coordenadas polares.
```

In []:

```
polar_plot(sin(15*x)^2, (x, 0, 2*pi)) #rosacea de 30 petalas
```

Exercício

1. Plote o grafico de uma circunferencia usando coordenadas parametricas.
2. Plote o grafico de uma circunferencia usando coordenadas polares.

Curvas Definidas Implicitamente

Elipse.

In []:

```
x,y=var('x,y')
```

In []:

```
implicit_plot(x^2/4+y^2/9==1, (x,0,1),(y,-3,3)) ## Atencao aos valores de x
```

In []:

```
implicit_plot(x^2/4+y^2/9==1, (x,-2,2),(y,-3,3)) ## Atencao aos valores de x
```

Elipse em coordenadas polares

In []:

```
b=3;
a=2;
# e=c/a=sqrt(b^2-a^2)/b
e=sqrt(b^2-a^2)/b
```

In []:

```
polar_plot(b*(1-e^2)/(1+e*cos(x)), (x, 0, 2*pi)) ###Falta Transladar e Girar.(o Foco esta na origem)
```

Customizando as plotagens.

In []:

```
G1 = plot(2*exp((-x^4)), (-1.5,1.5), linestyle='solid')
G2 = plot(3*exp((-x^4)), (-1.5,1.5), linestyle='dashed')
G3 = plot(4*exp((-x^4)), (-1.5,1.5), linestyle='dotted')
(G1 + G2 + G3).show()
```

In []:

```
G1 = plot(2*exp((-x^4)), (-1.5,1.5), thickness=1)
G2 = plot(3*exp((-x^4)), (-1.5,1.5), thickness=3)
G3 = plot(4*exp((-x^4)), (-1.5,1.5), thickness=6)
(G1 + G2 + G3).show()
```

In []:

```
plot(sin(x), (0,4*pi), legend_label='y=sin(x)',title='Grafico de f(x)=sin(x)')
```

Preenchimentos

In []:

```
p1 = plot(sin(x), (x, 0, 4*pi), fill='axis')
p2 = plot(sin(x), (x, 0, 4*pi), fill='min',fillalpha=1.0)
p3 = plot(sin(x), (x, 0, 4*pi), fill='max', fillalpha=0.25)
p4 = plot(sin(x), (x, 0, 4*pi), fill=x,fillcolor='red', fillalpha=0.3)
G = graphics_array([[p1, p2], [p3, p4]])
G.show(figsize=[7,5], aspect_ratio=1)
```

Como Salvar Figuras.

Para salvar figuras podemos guarda-la em uma variavel e então usar a sintaxe **save(figura,' figura.pdf')**(outros tipos de arquivos possiveis são png, jpg, etc)

In []:

```
p=line([[1,0],[1,1],[0,1],[1,0]]) ##Segmento de reta ligando os pontos da lista.
```

In []:

```
p
```

Exercicio

Plote o grafico do cosseno entre 0 e 2π e salve ele.

In []: