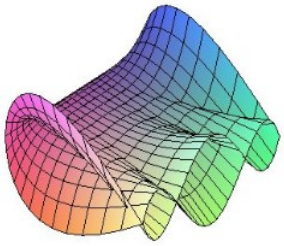


# Cálculo Diferencial e Integral: um kit de sobrevivência "SageMath"

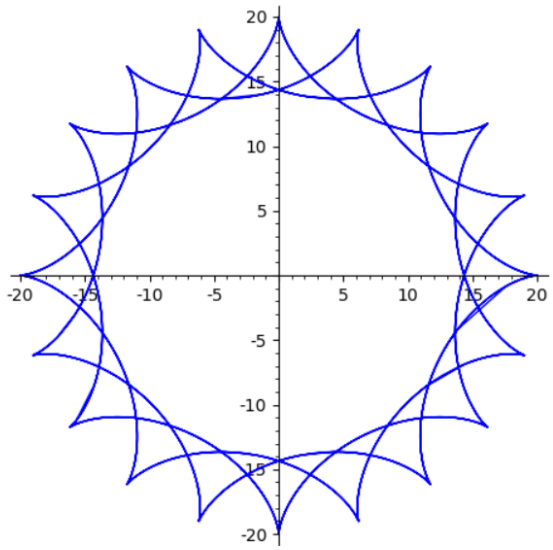
Ester Heloisa Bento.
Orientador: Prof. Dr. Rodrigo Martins.

## Comando *Parametric_plot*

Para elaborar um gráfico paramétrico no sage basta escrever:
var('t')
$parametric\_plot([f_1, f_2], (t, t_{min}, t_{max})$
Onde $f_1, f_2$ são funções coordenadas e $t_{min}, t_{max}$ são números reais.

```
In [1]: t=var('t')
        parametric_plot([17*cos(t) + 3 * cos(17*t/3), 17*sin(t) - 3 * sin(17*t/3)], (t, 0, 18*pi))
```

Out[1]:




O Sage plota automaticamente gráficos $2d$ ou $3d$ e uma curva ou superfície, dependendo de quantas variáveis e coordenadas você especificar. Mas normalmente usamos parametric_plot para objetos parametrizados por uma ou variável em $\mathbb{R}^2$ e duas variáves com parametric_plot3d em $\mathbb{R}^3$.

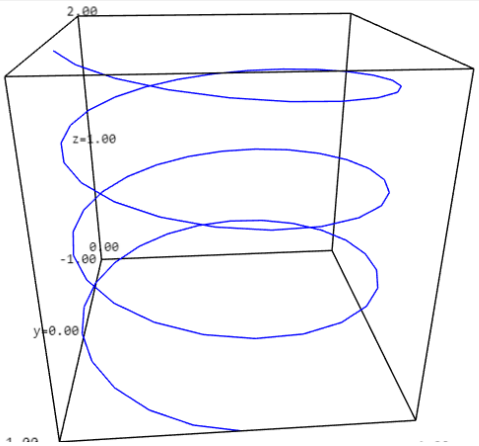
Existem quatro maneiras de chamar esta função:
$parametric\_plot3d([f_1, f_2, f_3], (t_{min}, t_{max})$ $f_1, f_2, f_3$ são funções e $t_{min}, t_{max}$ são números reais.
$parametric\_plot3d([f_1, f_2, f_3], (t, t_{min}, t_{max})$ $f_1, f_2, f_3$ podem ser vistos como funções de $t$.

$parametric\_plot3d([f_1, f_2, f_3], (t_{min}, t_{max}), (u_{min}, u_{max})$ $f_2, f_2, f_3$ são cada uma função de duas variáveis.
$parametric\_plot3d([f_1, f_2, f_3], (t, t_{min}, t_{max}), (u, u_{min}, u_{max})$ $f_1, f_2, f_3$ podem ser vistos como funções de $t$ e $u$.

1. Uma curva de espaço definida por três funções de 1 variável.

```
In [2]: t=var('t')
        parametric_plot3d((sin(t), cos(t), lambda t: t/10), (0, 20))
```

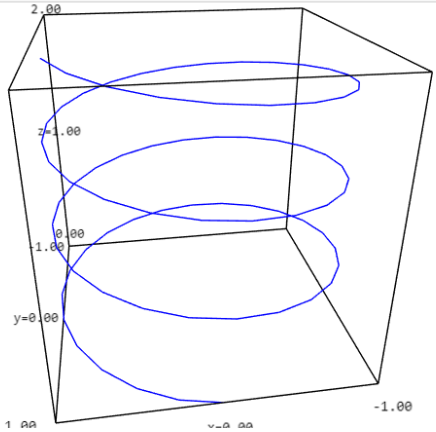
Out[2]:




2. Em seguida, desenhamos o mesmo gráfico acima, mas usando funções simbólicas:

```
In [3]: t=var('t')
        parametric_plot3d((sin(t), cos(t), t/10), (t,0, 20))
```

Out[3]:

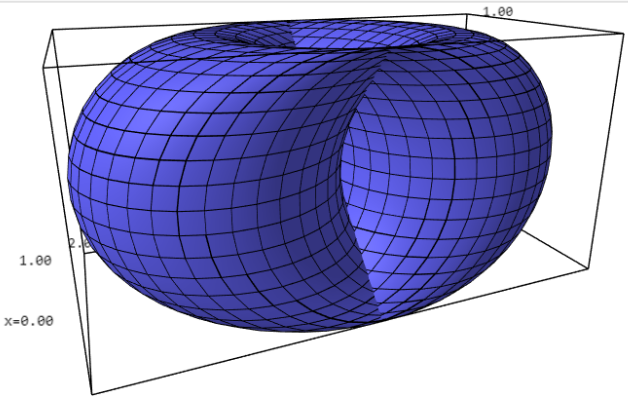



Demonstramos duas das quatro maneiras de chamar esta função, pois as outras duas formas são analogas.

- Adicionar Malha;
  t,u=var('t,u')
  $parametric\_plot3d((f_1, f_2, f_3), (t, t_{min}, t_{max}), (u, u_{min}, u_{max}), mesh = true)$
  São duas opções para malha: True ou False.

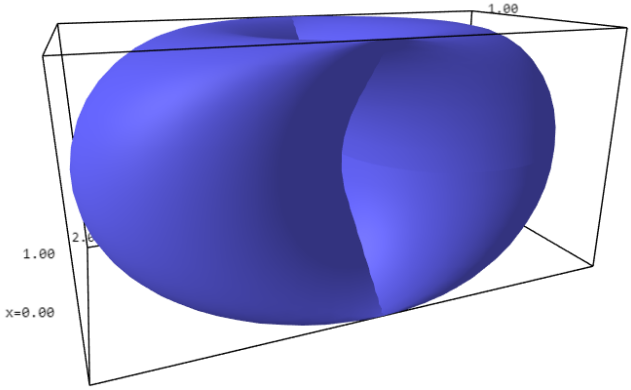
In [4]:
```
u, v = var('u,v')
parametric_plot3d((cos(u), sin(u)+cos(v), sin(v)), (u,0,2*pi), (v,-pi,pi), mesh=True)
```

Out[4]:




In [5]:
```
u, v = var('u,v')
parametric_plot3d((cos(u), sin(u)+cos(v), sin(v)), (u,0,2*pi), (v,-pi,pi), mesh=False)
```

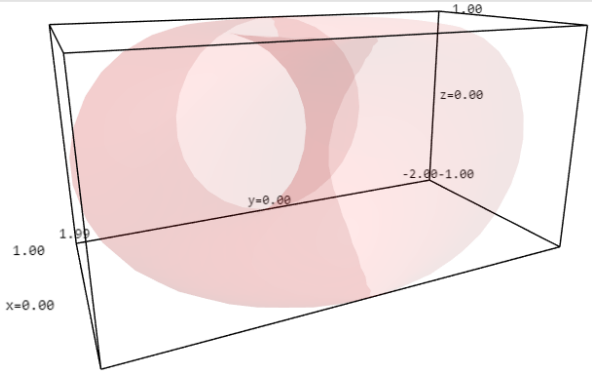
Out[5]:




- Aumentamos o número de pontos de plotagem e tornamos a superfície transparente;
  t,u=var('t,u')
  $parametric\_plot3d([f_1, f_2, f_3], (t, t_{min}, t_{max}), (u, u_{min}, u_{max}), opacity = n, plot\_points = [j, l])$

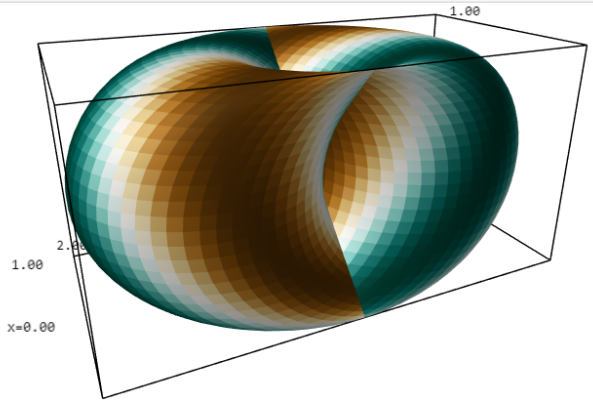
In [6]:
```
t,u=var('t,u')
parametric_plot3d((cos(t), sin(t)+cos(u), sin(u)), (t,0,2*pi), (u,-pi,pi), color='red', opacity=0.1, plot_points=[30,30])
```

Out[6]:




- Também é possível colorir a superfície usando uma função de coloração e um mapa de cores da seguinte maneira;
  t,u=var('t,u')
  def cf(t,u): return função
  $parametric\_plot3d([f_1, f_2, f_3], (t, t_{min}, t_{max}), (u, u_{min}, u_{max}), color = (cf, colormaps.cor)$

In [7]:
```
t,u = var('t,u')
def cf(t,u): return sin(t+u/2)**2
P = parametric_plot3d((cos(t), sin(t)+cos(u), sin(u)),(t,0,2*pi), (u,-pi,pi), color=(cf,colormaps.BrBG), plot_points=[60,60])
P.show()
```




Essas colorações podem ser visualizadas utilizando o comando sorted(colormaps).

**Observação:** As mesmas opções que valem para o comando *plot* tambem valem para o comando *parametric_plot*.

# Referências


[1] BARD, Gregory V. Sage para Estudiantes de Pregrado. Cochabamba: Sagemath, 2014. Tradução de: Diego Sejas Viscarra.

[2] Sage,Manual de referencias do sage 9.1. Disponível em: $< https://doc.sagemath.org/html/en/reference/plot3d/sage/plot/plot3d/parametric\_plot3d.html >$ Acesso em: 08 de dezembro de 2020.