

# Apresentações em Beamer

Congresso de Estudantes de Matemática da UFABC

---

Daniel Miranda<sup>1</sup>

17 de outubro de 2023

<sup>1</sup>UFABC



Uso de figuras

Geogebra

Ipe

Tikz

Introdução

Estruturas de Repetição

Gráficos de funções

- Arquivos: <https://danielmiranda.prof.ufabc.edu.br/figuras/>
- Modelo overleaf: <https://www.overleaf.com/read/kytcqmfqbfbvh>

## Uso de figuras

---

As figuras podem assumir muitas formas. Podem ser gráficos, diagramas, fotos, desenhos ou mapas.

- A ilustração auxilia na compreensão de conceitos.
- As imagens são processadas mais rapidamente pela mente do que as sentenças.
- Devem ser usadas em conjunção com o texto escrito.
- Pense deliberadamente sobre o seu propósito e use o bom senso para escolher a figura mais eficaz para comunicar o ponto principal.

- A consideração mais importante para as figuras é a simplicidade. Escolha imagens que o espectador possa captar e interpretar de forma clara e rápida.
- Considere o tamanho, a resolução, a cor e o destaque de recursos importantes.
- As figuras devem ser grandes e com resolução suficiente.
- Considere também o formato que o texto será publicado: em preto e branco ou em cores.
- Devem estar centralizados na página.
- Devem ser numeradas e ter uma legenda descritiva
- Você deve usar texto para orientar o leitor na interpretação das informações incluídas em uma figura, tabela ou gráfico – diga ao leitor o que a figura ou tabela transmite e por que foi importante incluí-la. “A Figura 1 mostra que...”

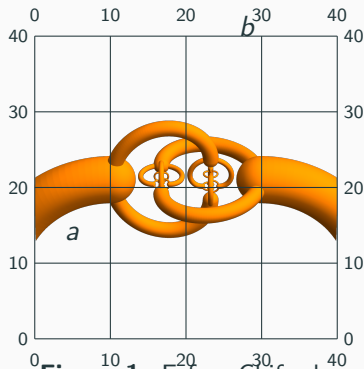
# Inserindo Imagem

Como inserir uma figura em  $\text{\LaTeX}$ :

```
1 \begin{figure}[opção]
2   \centering
3
4   [Figura] %imagem, tikzpicture
5
6   \caption{Legenda da Figura}
7   \label{nomedafigura}
8 \end{figure}
```

# Colocando $\text{\LaTeX}$ sobre uma imagem

```
1 \begin{overpic}[grid,abs,unit=1mm,scale=.25,width=4cm]{horned 2. pdf}  
2 \put(4,13){$a$}  
3 \put(27,40){$b$}  
4 \end{overpic}
```

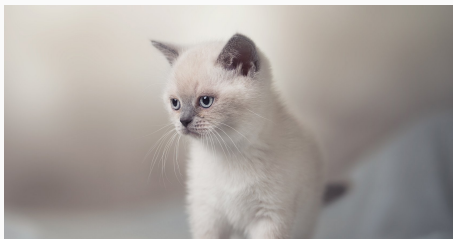


**Figura 1:** Esfera Chifruda

# Cortando uma Imagem

1

```
\includegraphics[trim=25cm 4.5cm 0cm 0cm,clip, width=6cm]{gato.jpg}
```



# Geogebra

---

- GeoGebra (aglutinação de Geometria e Álgebra) é um aplicativo computacional livre de matemática dinâmica que combina conceitos de geometria e álgebra em uma única interface.
- Tem a capacidade de exportar figuras em TikZ.
- **Download:** <https://www.geogebra.org/download>
- **Manual:** <https://wiki.geogebra.org/en/Manual>

**Ipe**

---

Ipe é um editor de desenhos para criação de figuras em formato PDF (voltado para o  $\text{\LaTeX}$ ).

As principais características do Ipe são:

- Entrada de texto como código-fonte LaTeX. Isto facilita a inserção de expressões matemáticas
- Produz PDF puro, incluindo o texto. O Ipe converte a fonte LaTeX em PDF quando o arquivo é salvo.
- **Download:** <https://ipe.otfried.org/>
- **Manual:** <https://otfried.github.io/ipe/>

# Tikz

---

- TikZ é uma linguagem para produzir imagens vetoriais,
- Imagens vetoriais são criadas diretamente a partir de formas geométricas definidas em um plano cartesiano, como pontos, linhas, curvas e polígonos. As imagens vetoriais são usualmente mais leves e não perdem qualidade ao serem ampliadas.

## Vantagens

- Nitidez/qualidade das imagens construídas.
- Fácil edição.
- Reaproveitáveis.
- Criadas dentro do próprio  $\text{\LaTeX}$

Para começar a usar o TikZ, precisamos de carregar o pacote tikz:

```
1 \usepackage{tikz}
```

```
2
```

Sempre que quisermos criar uma figura em TikZ, precisamos de utilizar o ambiente tikzpicture.

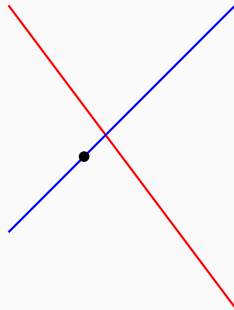
```
1  
2 \begin{tikzpicture}[opções globais da figura]  
3  
4 [código]  
5  
6 \end{tikzpicture}
```

- As coordenadas cartesianas de um ponto são fornecidas no formato  $(a, b)$ .
- As coordenadas polares de um ponto são fornecidas no formato  $(\theta : r)$  onde  $\theta$  é medido em graus.

- `\draw`: para desenhar linhas.
- `\fill` : para preencher áreas com cores sólidas.
- `\shade` : para preencher áreas com cores gradientes.
- `\clip` : para cortar.

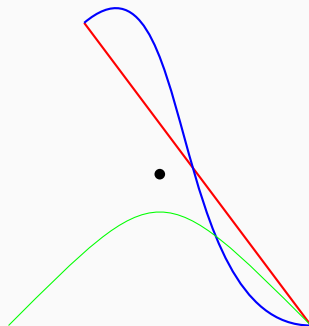
## Exemplo

```
1 \begin{tikzpicture}  
2   \draw[red, thick] (-1,2) -- (2,-2);  
3   \draw[blue, thick] (-1,-1) -- (2,2);  
4   \fill[black] (0,0) circle (2pt);  
5 \end{tikzpicture}
```

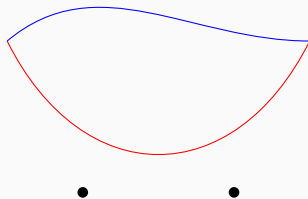


# Pontos, Retas e Caminhos

```
1 \begin{tikzpicture}  
2   \draw[red, thick] (-1,2) -- (2,-2);  
3   \fill[black] (0,0) circle (2pt);  
4   \draw[blue, thick] (-1,2) to[out=40, in=180]  
5     (2,-2);  
6   \draw[green] (-2,-2) .. controls (0,0) .. (2,-2);  
7 \end{tikzpicture}
```

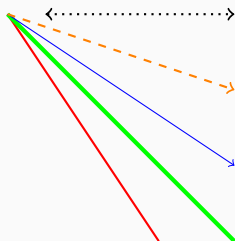


- `\draw[red] (-2,2) .. controls (-1,0) and (1,0) .. (2,2);`: Desenha uma curva de Bézier. Existem 4 pontos que o definem:  $(-2,2)$  e  $(2,2)$  são seus pontos finais,  $(-1,0)$  e  $(1,0)$  são pontos de controle que determinam "quão curvo" ela é. Você pode pensar nesses dois pontos como "pontos atratores".
- `\draw[blue] (-2,2) to[out=40, in=180] (2,2);`: Desenha uma curva ligando  $(-2,2)$  a  $(2,2)$ , saindo de  $(-2,2)$  com um ângulo de 40 graus e chegando a  $(2,2)$  com um ângulo de 180 graus.



# Decoração

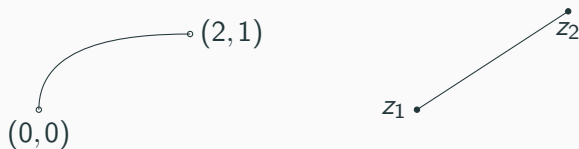
```
1 \begin{tikzpicture}  
2 \draw[red, thick] (-1,2) -- (1,-1);  
3 \draw[green, ultra thick] (-1,2) -- (2,-1);  
4 \draw[blue, ->] (-1,2) -- (2,0);  
5 \draw[orange, thick, dashed, ->] (-1,2) -- (2,1);  
6 \draw[black, thick, dotted, <->] (-0.5,2) -- (2,2);  
7 \end{tikzpicture}
```



# Legendas

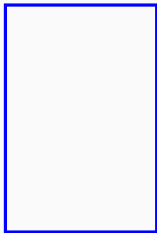
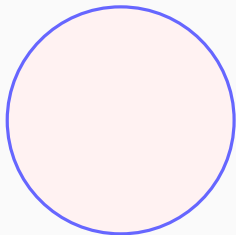
```
1 \node at (1,2) {$x$};  
2 \draw(0,0) node[below]{$(0,0)$}to[out=90,in=180] (2,1) node[right]{$(  
   2,1)$};  
3 \draw(0,0) circle[radius=1pt] (2,1) circle[radius=1pt];  
4 \draw(5,0) node[left]{$z_1$}—(7,1.3) node[below]{$z_2$};  
5 \draw[fill](5,0) circle[radius=1pt] (7,1.3) circle[radius=1pt];
```

x



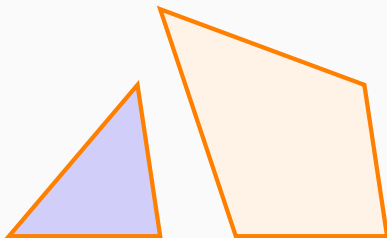
# Círculos, Elipses e Retângulos

```
1 \begin{tikzpicture}  
2   \filldraw[color=blue!60, fill=red!5, very thick](-1,0) circle (1.5);  
3   \fill[blue!40] (2.5,0) ellipse (1.5 and 0.5);  
4   \draw[blue, very thick] (5,-1) rectangle (7,2);  
5   \shade[top color=blue,bottom color=red](9,0) ellipse (1.5 and 0.7);  
6 \end{tikzpicture}
```



# Polígonos e Arcos

```
1 \begin{tikzpicture}  
2 \draw[orange, ultra thick, fill=orange!10!blue!20] (5,0) — (7,0) — (  
   6.7,2) — cycle;  
3 \draw[orange, ultra thick, fill=orange!10] (8,0) — (10,0) — (9.7,2) —  
   (7,3) — cycle;  
4 \draw[ultra thick, ->] (13.5,0) arc (0:150:1);  
5 \end{tikzpicture}
```



`\draw[ultra thick, ->] (13.5,0) arc (0:150:1);` Este comando irá desenhar um arco começando em (13.5,0). Além do ponto inicial, você deve fornecer três valores adicionais: os ângulos inicial e final e o raio; aqui, esses três valores de parâmetros são fornecidos no formato (0:150:1).

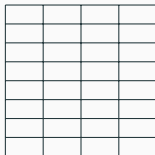


# Grid

Para desenhar um grid ancorado nos pontos  $(0, 0)$  e  $(2, 2)$  com passos horizontais de 0.5 e verticais de 0.25

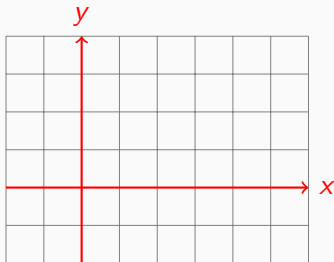
1

```
\draw[xstep=0.5,ystep=0.25] (0,0) grid (2,2);
```



# Grid

```
1 \draw[step=0.5cm,help lines] (-1,-1) grid (3,2);  
2 \draw[->,red,thick] (-1,0)---(3,0) node[right]{$x$};  
3 \draw[->,red,thick] (0,-1)---(0,2) node[above]{$y$};  
4
```



```
\foreach <variavel> in {<lista>} {<comando>}
```

No caso mais simples:

- <variables> é um único comando TEX como \i.
- <list> é uma lista de valores separados por vírgulas. Qualquer coisa pode ser usada como um valor, mas os números são mais prováveis.
- {<commands>} é um conjunto de comandos a ser executado, sempre entre colchetes.

# Exemplo

```
1 \begin{tikzpicture}  
2 \foreach \x in {0,1,...,5}  
3 {  
4 \draw[black] (\x,0) circle (0.1cm) node[above] {\x};  
5 }  
6 \end{tikzpicture}  
7
```

0 1 2 3 4 5



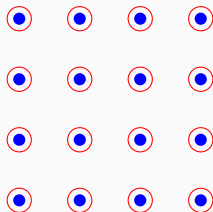
# Exemplo

```
1 \begin{tikzpicture}  
2 \foreach \x in {0,1,...,5}  
3 {  
4 \draw[black] (\x,0) circle (0.1cm) node[above] {\x};  
5 }  
6 \end{tikzpicture}
```



# Exemplo

```
1 \begin{tikzpicture}  
2   \foreach \x in {0,1,2,3}{  
3     \foreach \y in {0,1,2,3}{  
4       \draw[red] (\x,\y) circle (0.2cm);  
5       \fill[blue] (\x,\y) circle (0.1cm);  
6     } }  
7 \end{tikzpicture}
```



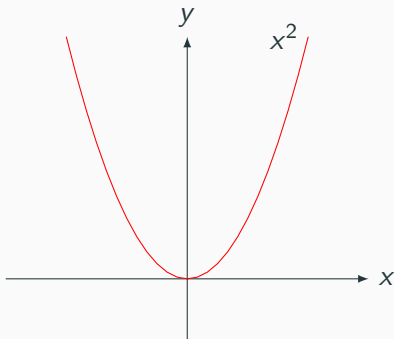
Para esboçar uma curva parametrizada por como  $(f(x), g(x))$  com  $a \leq x \leq b$  basta utilizar o comando

```
plot({f(x)},{g(x)})
```

onde  $x$  é a variável e  $f$  e  $g$  são as componentes da parametrização.

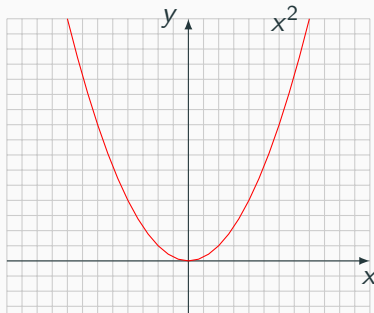
## Exemplo

```
1 \draw[>,<-latex] (-3,0)---(3,0) node[right]{$x$};  
2 \draw[>,<-latex] (0,-1)---(0,4) node[above]{$y$};  
3 \draw[red,domain=-2:2] plot({\x},{(\x)^2});  
4 \node at (1.6,4) {$x^2$};
```



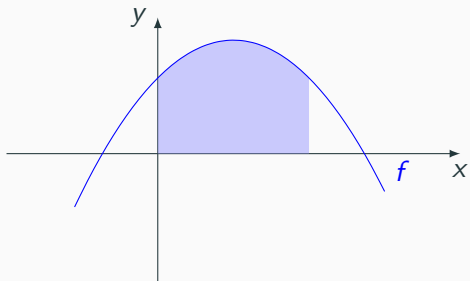
## Exemplo

```
1 \draw[gray!40,xstep=0.25,ystep=0.25] (-3,-1) grid (3,4);
2 \draw[->,-latex] (-3,0)---(3,0) node[below]{$x$};
3 \draw[->,-latex] (0,-1)---(0,4) node[left]{$y$};
4 \draw[red,domain=-2:2] plot({\x},{(\x)^2});
5 \node at (1.6,4) {$x^2$};
```



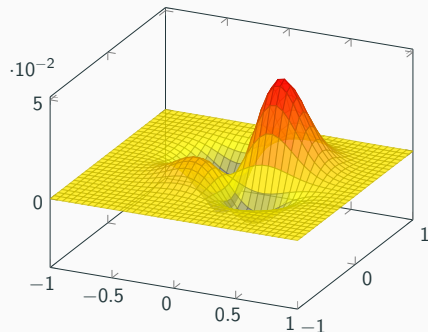
# Sombreado

```
1 \draw[>,<-latex] (-2,0)---(4,0) node[below]{$x$};  
2 \draw[>,<-latex] (0,-3.8)---(0,3.6) node[left]{$y$};  
3 \fill[blue!40!,opacity=0.5] (0,0) --- plot[domain=0:2] (\x,{-(\x)^2+2*\x+2}) --- (2,0) --- cycle;  
4 \draw[blue,domain=-1.1:3,smooth] plot(\x,{-(\x)^2+2*\x+2}) node[above right]{$f$};
```



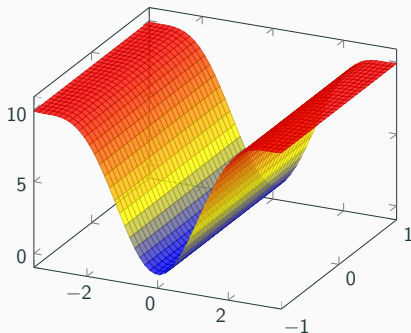
## Gráfico 3D

```
1 % precisa de \usepackage{pgfplots}  
2 \begin{axis}[domain=-1:1,y domain=-1:1]  
3 \addplot3[surf,opacity=0.8,samples=40] {x*y*exp(x+2*y-9*x^2-9*y^2)};  
4 \end{axis}
```



## Gráfico 3D

```
1 \begin{axis}  
2 \addplot3[surf,opacity=0.8,samples=50, samples y=30,domain=-3.5:3.5,  
   domain y=-1:1]{((x+\sin(\deg(x)))^2};  
3 \end{axis}
```



## Conclusão

# Apêndice

---

-  The not so short introduction to  $\text{\LaTeX}$  <http://alfarrabio.di.uminho.pt/~albie/lshort/>
-  CTAN: Comprehensive  $\text{\TeX}$  Archive Network <http://www.ctan.org/>
-  Pequeno resumo de comandos  $\text{\LaTeX}$  <http://natura.di.uminho.pt/~jj/latex2e.html>
-  The  $\text{\LaTeX}$  Companion por Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, maxChris Rowley
-  The  $\text{\LaTeX}$  Graphics Companion por Michel Goossens, Sebastian Rahtz, Frank Mittelbach