

Gráficos em Maple V

A Maple além de ser um processador simbólico, linguagem de programação, editor de textos matemáticos, é também um excelente produtor de gráficos. Vão-se estudar aqui algumas ferramentas disponíveis no Maple V.

Gráfico de uma função

O comando é o

```
plot(f(x), x=a..b, y=c..d, opção1, opção2, ...);
```

onde cada um destes operandos indica:

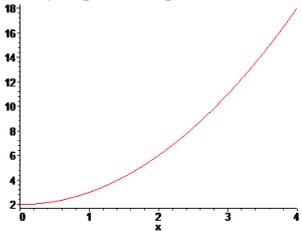
- plot é o comando maple que gera o gráfico
- f(x) é a função da qual se quer o gráfico
- x=a..b variação do domínio da função - começa em a e termina em b
- y=c..d variação da imagem da função - começa em c e termina em d e é opcional
- opção... opções desejadas para o gráfico

Seja o exemplo de construir a parábola $2 + x^2$, com intervalo de x de $[0, 4]$.

Um comando simples para isso é

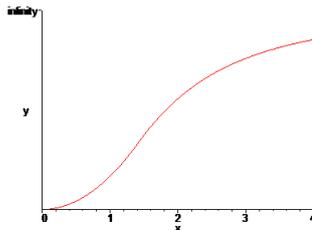
```
> plot(2+x^2, x=0..4);
```

Que gera o seguinte resultado



Acompanhe o que muda com o comando

```
> plot(2+x^2, x=0..4, y=2..infinity);
```



Note que no primeiro caso, como o eixo y não foi definido, o Maple calcula-o e mandar ver. No segundo caso, como há uma especificação, o Maple a obedece.

A seguir a lista de opções. Todas são fornecidas na forma de uma palavra chave, seguida por um igual e depois um valor. Assim: *opção=valor*.

axes Tem 4 valores: *FRAME*, *BOXED*, *NORMAL* ou *NONE*. *Boxed* desenha uma caixa em volta do gráfico e *None* não desenha os eixos.

axesfont Dá o nome da família de tipos usadas para marcar os eixos. Por exemplo `axesfont=[TIMES, ROMAN, 20]`.

color Dá a cor do gráfico. Pode ser (entre outras): *green*, *red*, *yellow*, *orange*, *pink*, Pode ser também na forma `COLOR(RED, r, g, b)` onde r , g e b são números reais entre 0 e 1 que compõe a cor pedida.

coords Indica o sistema de coordenadas utilizado. Pode ser (entre outros) *bipolar*, *cardioid*, *cartesian*, *polar*, etc. O padrão é *cartesian*.

filled Se for *true* indica que a região entre o eixo e a curva será pintada. O padrão é *false*.

labels Define o nome dos eixos. No formato `labels = [x, y]`. O padrão é usar os nomes das variáveis na função.

linestyle Inteiro de 1 a 4, indicando sólido (1) ou pontilhado (4), ou valores intermediários.

scaling Pode ser *CONSTRAINED* ou *UNCONSTRAINED*. O padrão é *UN...*. Indica a proporção entre as escalas dos eixos.

Outras opções Que devem ser estudadas na literatura: *adaptative*, *discont*, *font*, *labelsdirection*, *labelfont*, *legend*, *numpoints*, *style*, *symbol*, *symbolsize*, *thickness*, *thickmarks*, *title*, *titlefont*, *view*, *xtickmarks*, *ytickmarks*.

Vários gráficos no mesmo desenho

Para colocar mais de um gráfico no mesmo desenho, deve-se usar o mesmo comando, mas em cada um dos parâmetros deve-se usar uma lista, formada pelos parâmetros individuais de cada gráfico. Quando um parâmetro é comum, não é necessário criar a lista. Note que os domínios das funções (o x) devem ser comuns.

Veja-se a construção de um gráfico envolvendo as curvas de seno e cosseno.

```
plot([sin(x), cos(x)], x=-2*Pi..2*Pi, color=[yellow, red]);
```

Digite isso no maple e veja o resultado

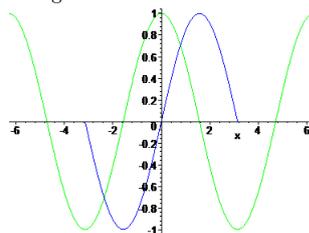
Outra hipótese, boa quando não há nenhuma coincidência e os gráficos devem ser mostrados juntos por alguma outra razão é:

1. Chamar o pacote `plots` (`with(plots)`).
2. Criar cada gráfico, com suas especificidades, mas colocando-o em uma variável qualquer.
3. Usar o comando `display` para mostrar todas as variáveis (gráficos) criadas.

Veja no exemplo

```
> with(plots):
> u1:=plot(sin(x), x=-Pi..Pi, color=blue):
> u2:=plot(cos(x), x=-2*Pi..2*Pi, color=green):
> display(u1, u2);
```

Com o seguinte resultado:



Gráficos Tridimensionais

O comando agora é `plot3d` que é muito parecido com o `plot`, sujeito apenas às modificações devidas ao novo eixo.

Antes, um exemplo

```
> plot3d(sin(x)*cos(y), x=-Pi..Pi, y=-3*Pi..3*Pi);
```

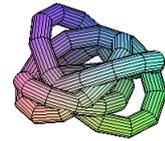
Execute este comando no Maple e veja que belo desenho.

As opções do comando `plot3d` são muitas e variadas. Vale uma consulta à referência para estudá-las.

Antes de encerrar uma minhoca infinita:

```
> with(plots):
> a:=2: b:=4: c:=3: d:=5: r:=2/3: s:=2:
> cur:=[-r*cos(a*t)+s*cos(b*t), r*sin(a*t)+s*sin(b*t), sin(d*t)+sin(c*t)]/2:
> tubeplot(cur, radius=0.4, t=0..2*Pi, grid=[50, 20]);
```

Que apresentou:

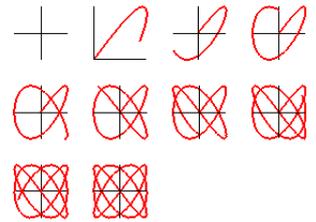


Animações e gravação direto

Há todo um conjunto de comandos que preparam animações, aqui consideradas como um conjunto de arquivos planos, gerados em sequência. Veja-se por exemplo, a seguinte sequência de figuras de Lissajous:

```
with(plots):
Y:=animatecurve([sin(3*t), sin(4*t)], t=0..2*Pi, numpoints=80, thickness=2, frames=10, tickmarks=[0,0], scaling=constrained):
> display(Y);
```

Com o seguinte resultado:



Para gravar as imagens em arquivos em disco (formatos EPS, GIF, BMP, JPG, etc) O jeito mais fácil é clicar com o botão direito na imagem e depois responder o nome do arquivo como resposta à questão *Export as* do Maple.

Mas, existe um comando que faz isso, chamado `plotsetup`, que estabelece onde os desenhos vão ser gravados. O comando é

```
plotsetup(device, arquivo, opções)
```

`device` é o tipo de arquivo e pode ser *ps*, *gif*, *jpeg*, *bmp*,... `arquivo` é o nome completo do arquivo a gravar e opções dependem do `device`. Consulte as referências para saber mais deste comando.

Para você fazer

1. Refaça todos os exemplos desta folha, primeiro de maneira literal e depois variando um dos parâmetros de cada vez para verificar qual o resultado gerado.

2. Na gincana 2012 da Eng. Mecânica as equipes precisam produzir um carrinho movido à ar comprimido. Suponha que o seu carrinho teve o seguinte desempenho:

tempo (s)	1	2	3	4	5
distância (m)	10	20	26	32	37

De posse destes dados, use o Maple para construir um gráfico mostrando o desempenho do seu carrinho. Escreva no verso desta folha qual o comando Maple usado e rascunhe a lápis qual o jeito do gráfico produzido.

