ABC do Maple V

Ma To Fu

LabMAC-UEM 1998

(www.geocities.com/cnumap)

INDICE

Apresentação

O Maple V é um sistema de computação algébrica bastante popular nos meios acadêmicos e científicos. Similarmente ao sistema Mathematica, é capaz de efetuar operações simbólicas e cálculos complexos de uma maneira simples e também possui recursos para programação. Há contudo algumas limitações para cálculos numéricos de grande porte, onde o Fortran continua a imperar.

Nestas notas apresentamos apenas um pequeno tutorial dos comandos básicos do Maple V. O objetivo é tornar o leitor capaz de fazer cálculos simples e programação elementar, bem como plotar gráficos 2D e 3D. Esperamos que o leitor, após ter utilizado estas notas, sinta-se encorajado a explorar por si mesmo as outras possibilidades do Maple V. Os interessados em tópicos avançados tais como Álgebra Linear, Equações Diferencias, Progamação Linear ou Estatística, poderão encontrar referências e *links* no final do texto.

Este texto é uma versão *simplificada* e adaptada para a *Release 4* das notas de minicursos realizados pelo autor na Universidade Estadual de Maringá entre 1993 e 1997, e no Laboratório Nacional de Computação Científica (LNCC/CNPq) em 1995. O autor aproveita para agradecer os professores Doherty Andrade (Maringá) e Jaime Muñoz Rivera (Rio de Janeiro) pelas sugestões e correções apresentadas. O autor também agradece a hospitalidade do LNCC (Petrópolis) onde estas notas foram preparadas.

Petrópolis, 18/12/98.

(modificado em 24/01/2000)

1 Primeiros Passos

Nesta primeira parte, através de exemplos, discutiremos alguns comandos que são absolutamente indispensáveis. Toda instrução Maple inicia-se após o sinal (>) e termina com o sinal (**;**) ou (**:**).

As operações aritméticas básicas são feitas com os seguintes símbolos:

+ (adição) **-** (subtração) ***** (multiplicação) / (divisão) ^ (potênciação).

A multiplicação e divisão são efetuadas antes da adição e subtração e potências são efetuadas antes da multiplicação. Para evitar confusões podemos utilizar parênteses **()** para agrupar expressões. Porém colchetes **[]** e chaves **{}** não devem ser utilizados para este fim.

Operações Básicas

```
> 5*3+9;
```

$$24$$

```
> 3^2*2;
```

$$18$$

```
> (11*4^5)/(-5+14*3);
```

$$\frac{11264}{37}$$

Vejamos o que acontece quando esquecemos o sinal de ponto e vírgula (;).

```
> 2*3
>
```

Warning, incomplete statement or missing semicolon

O sistema reclama em azul dizendo que a instrução **2*3** está incompleta ou que falta um ponto e vírgula. Neste caso devemos voltar e corrigir o problema.

```
>
```

A Representação Decimal

Podemos usar o comando **evalf** (avaliar com ponto flutuante) para se obter uma representação decimal de um número. Normalmente, o sistema utiliza dez algarismos significativos.

```
> evalf(176/47);
```

$$3.744680851$$

```
> 25*sqrt(2);
```

$$25\sqrt{2}$$

```
> evalf(");
```

$$35.35533905$$

As aspas (") significam *o valor acima* . Assim, no exemplo anterior, as aspas representam o valor $25\sqrt{2}$ algarismos significativos.

> **evalf(Pi,30);**

$$3.14159265358979323846264338328$$

>

Atribuindo Letras e Nomes

Nos trabalhos mais complexos é importante podermos representar por letras expressões complicadas. No Maple esta representação é feita através do símbolo (**:=**). A regra é simples: **A := B** significa que o lado direito (B) é a definição do lado esquerdo (A).

> **A1 := x*sqrt(7);**

$$A1 := x\sqrt{7}$$

> **A1^2;**

$$7x^2$$

>

Operações Simbólicas

Uma das vantagens da computação algébrica é a capacidade de se fazer cálculos simbólicos. Veremos a seguir os comandos **expand** (expandir), **factor** (fatorar) e **simplify** (simplificar).

> **A2 := (x^3*y+x^3-y^4-y^3)/(y+1);**

$$A2 := \frac{x^3y + x^3 - y^4 - y^3}{y+1}$$

> **simplify(A2);**

$$-y^3 + x^3$$

> **A3:=factor(A2);**

$$A3 := (x-y)(x^2 + xy + y^2)$$

> **expand(A3);**

$$-y^3 + x^3$$

>

Funções Elementares

No Maple as funções elementares já estão pré-definidas, como por exemplo, as funções trigonométricas **sin** (seno), **cos** (cosseno) e **tan** (tangente). A função logarítmo natural é representada por **ln** e a função exponencial é representada por **exp** . A constante de Euler **e** é definida por **exp(1)** . Vejamos alguns exemplos.

> **expand(cos(x+y));**

$$\cos(x)\cos(y) - \sin(x)\sin(y)$$

> **evalf(exp(1),25);**

$$2.7182818284590452353602874$$

> **exp(ln(xyz));**

$$xyz$$

>

Definindo Funções

Vamos agora definir funções. A maneira mais simples de se definir uma função é assim:

f := (*variáveis* **) ->** **(** *expressão contendo variáveis* **)**

> **f := x -> sin(x)*cos(x);**

$$f := x \rightarrow \sin(x)\cos(x)$$

> **f(z);**

$$\sin(z)\cos(z)$$

> **f(Pi/4);**

$$\frac{1}{2}$$

> **g := exp + f;**

$$g := \exp + f$$

> **g(-x);**

$$e^{(-x)} - \sin(x)\cos(x)$$

> **h := (x,y) -> x^2-5*y^3;**

$$h := (x,y) \to x^2 - 5y^3$$

> **h(s,t);**

$$s^2 - 5t^3$$

Funções definidas por duas ou mais expressões podem ser definidas com o uso de procedimentos **proc** . Veremos um exemplo no Capítulo 4 onde estudaremos a programação em Maple.

>

Resolução de Equações e Sistemas

O comando **solve** (resolver), serve para resolver equações diversas. No exemplo abaixo resolveremos uma equação na variável *x* .

> **equa1 := x^3+3*x^2-4=0;**

$$equa1 := x^3 + 3x^2 - 4 = 0$$

> **solve(equa1);**

$$1, -2, -2$$

Quando a equação possui mais de uma variável, é fundamental indicar ao sistema a incógnita do problema.

> **equa2 := 2*x+y=0;**

$$equa2 := 2x + y = 0$$

> **solve(equa2, x);**

$$-\frac{1}{2}y$$

> **solve(equa2, y);**

$$-2x$$

O comando **solve** resolve também equações com raízes complexas. A letra (**I**) representa a unidade imaginária dos números complexos.

> **solve(z^2+1,z);**

$$I, -I$$

Se o comando **solve** não conseguir apresentar exatamente as raízes desejadas, podemos então executar o comando **fsolve** (resolver com ponto flutuante) para se obter raízes aproximadas.

> **equa3 := x^6 - 2*x^2 + 2*x;**

$$equa3 := x^6 - 2x^2 + 2x$$

> **solve(equa3,x);**

$$0, \mathrm{RootOf}(\_Z^5 - 2\_Z + 2)$$

> **fsolve(equa3,x);**

$$0, -1.364430112$$

Por questões práticas, a função **fsolve** não mostra automaticamente as raízes complexas. É preciso adicionar a opção **complex** .

> **fsolve(equa3,x,complex);**

$$0, -1.364430112, -.1929606049 - 1.268917574\,I, -.1929606049 + 1.268917574\,I, .8751756609 - .3519221356\,I, .8751756609 + .3519221356\,I$$

Os comandos **solve** e **fsolve** também funcionam com sistemas. A sintaxe é a seguinte:

solve({ equações **},{** incógnitas **})** .

Observe a utilização das chaves **{}** para representar conjunto de equações e de incógnitas.

> **equa4 := x + 2*y + 3*z = 7;**

$$equa4 := x + 2y + 3z = 7$$

> **equa5 := 5*x - 2*y = 12;**

$$equa5 := 5x - 2y = 12$$

> **equa6 := 2*x - y + 3*z = -6;**

$$equa6 := 2x - y + 3z = -6$$

> **sol := solve({equa4,equa5,equa6} , {x,y,z});**

$$sol := \left\{x = \frac{62}{13}, z = \frac{-125}{39}, y = \frac{77}{13}\right\}$$

> **evalf(sol,13);**

$$\{x = 4.769230769231, z = -3.205128205128, y = 5.923076923077\}$$

>

Operações com Polinômios

Em computação algébrica também podemos operar polinômios simbolicamente. Consideremos os dois polinômios abaixo.

> **p := x^4 - x^3 - 10*x^2 + 10*x + 6;**

$$p := x^4 - x^3 - 10x^2 + 10x + 6$$

> **q := x^3 - 4*x^2 + x + 6;**

$$q := x^3 - 4x^2 + x + 6$$

Escrevendo *p* (*x*) em fatores primos:

> **factor(p);**

$$(x - 3)(x^3 + 2x^2 - 4x - 2)$$

Dividindo *p* (*x*) por *q* (*x*):

> **p/q;**

$$\frac{x^4 - x^3 - 10x^2 + 10x + 6}{x^3 - 4x^2 + x + 6}$$

Normalizando (simplificando) a expressão racional acima:

> **normal(");**

$$\frac{x^3 + 2x^2 - 4x - 2}{x^2 - x - 2}$$

Ainda podemos converter a expressão racional acima em frações parciais:

> **convert(p/q, parfrac, x);**

$$x + 3 - \frac{1}{x + 1} + \frac{2}{x - 2}$$

>

Comandos Diversos

Agora veremos mais alguns comandos que poderão ser úteis. Observe que o símbolo (**#**) é um sinal de comentário e o que vem depois não é levado em consideração pelo Maple.

> **33!; # fatorial de 33**

$$8683317618811886495518194401280000000$$

> **ifactor(''); # fatoração em primos do inteiro acima**

$$(2)^{31}\ (3)^{15}\ (5)^{7}\ (7)^{4}\ (11)^{3}\ (13)^{2}\ (17)\ (19)\ (23)\ (29)\ (31)$$

> **gcd(34,51); # máximo divisor comum entre 34 e 51.**

$$17$$

> **lcm(2,4,5); # mínimo múltiplo comum entre 2,4 e 5.**

$$20$$

> **(3-5*I)*(1+I); # multiplicando dois complexos.**

$$8-2I$$

> **convert(90*degrees, radians); # convertendo graus em radianos.**

$$\frac{1}{2}\pi$$

> **max(1,-20,13); # máximo entre 1, -20 e 13.**

$$13$$

> **Sum(k^2, k=1..5); # com S (maiúscula) indica uma somatória.**

$$\sum_{k=1}^{5} k^2$$

> **sum(k^2, k=1..5); # com s (minúscula) calcula a somatória.**

$$55$$

> **S10 := Sum(1/k^2, k=1..infinity); # somatória infinita.**

$$S10 := \sum_{k=1}^{\infty} \frac{1}{k^2}$$

> **value(S10); # value (valor de).**

$$\frac{1}{6}\pi^2$$

>

Comentários Finais

Via de regra, Maple imprime (na tela) todos resultados quando executados com (**;**). Se quisermos que o resultado seja calculado mas não impresso, devemos então substituir (**;**) por (**:**).

> **ano := 1998: # usamos dois pontos.**

Observe que o Maple não escreveu nada na tela. Mas o valor 1998 foi de fato atribuído à constante *ano* .

> **ano;**

$$1998$$

Para se saber sobre outros comandos e funções não abordados aqui, o leitor poderá executar **?contents** (no MapleVR4) e **?introduction** (no MapleVR5) . Em alguns casos podemos obter mais informações sobre um assunto executando o símbolo de interrogação seguido do assunto. Por exemplo, para se saber sobre trigonometria executa-se **?trig** .

> **?trig**

>

2 Cálculo Diferencial e Integral

Neste Capítulo discutiremos alguns dos aspectos práticos do Cálculo Diferencial e Integral de uma variável real.

Aproveitamos para sugerir a utilização de comando **restart** (reiniciar) que faz com que o sistema "limpe" a memória do Maple. Os símbolos e letras já atribuídos anteriormente ficam também liberados. É como (quase) se o Maple fosse recarregado.

> **restart;**

Calculando Limites

Os limites podem ser calculados com o comando **limit** , que pode ser aplicado às funções e expressões.

> **f1 := x-> (x^2+5)/(x^3); # definindo uma função.**

$$f1 := x \to \frac{x^2+5}{x^3}$$

> **limit(f1(x), x=1); # limite para x tendendo a 1.**

$$6$$

> **limit(f1(x), x=infinity); # limite para x tendendo a infinito.**

$$0$$

> **f2 := sin(x)/x; # definindo uma expressão contendo x.**

$$f2 := \frac{\sin(x)}{x}$$

Observe que $f2$ não é uma função para o Maple, mas tão somente uma expressão contendo a variável x . Portanto no comando **limit** escrevemos **f2** e não **f2(x)** .

> **limit(f2, x=0);**

$$1$$

> **limit(f2(x),x=0); # não faz sentido.**

$$\lim_{x \to 0} \frac{\sin(x)(x)}{x(x)}$$

Para se calcular limites laterais basta acrescentar as opções **left** (esquerda) ou **right** (direita). Vejamos um exemplo com a função tangente **tan** .

> **limit(tan(x), x=Pi/2, left);**

$$\infty$$

> **limit(tan(x), x=Pi/2, right);**

$$-\infty$$

Se desejamos apenas indicar um limite, então podemos utilizar o comando **Limit** com a letra L maiúscula. A sintaxe é a mesma.

> **Limit(x^2*sin(1/x), x=0, right);**

$$\lim_{x \to 0+} x^2 \sin\left(\frac{1}{x}\right)$$

> **value(");**

$$0$$

>

Cálculo de Integrais

As integrais indefinidas ou definidas são obtidas através do comando **int** (com letras minúsculas). Também podemos usar o comando **Int** (com letra i maiúscula) no caso de querermos a integral apenas indicada.

> **f3 := x -> a*x^2; # definindo uma função.**

$$f3 := x \to a\,x^2$$

> **Int(f3(x), x);**

$$\int a\,x^2\,dx$$

> **int(f3(x), x);**

$$\frac{1}{3}a\,x^3$$

> **int(ln(x),x);**

$$x\ln(x) - x$$

Para se calcular integrais definidas precisamos fornecer os limites de integração. A notação **x=a..b** significa que o x varia de a até b .

> **Área := Int(f3(x), x=0..1);**

$$Área := \int_0^1 a\,x^2\,dx$$

> **value(Área);**

$$\frac{1}{3}a$$

> **f4 := 1/x^2; # definindo uma expressão.**

$$f4 := \frac{1}{x^2}$$

> **Int(f4, x=1..infinity);**

$$\int_1^{\infty} \frac{1}{x^2}\,dx$$

> **value(");**

$$1$$

> **Int(Int(x^2+y^2, x=1..2), y=0..3);**

$$\int_0^3 \int_1^2 x^2+y^2\,dx\,dy$$

> **value(");**

$$16$$

>

Derivadas

Existem duas maneiras de se derivar funções no Maple. Uma delas se faz com o uso do operador diferencial **D** . Aqui daremos exemplos através da função **diff** (diferenciar).

> **f5 := x^2+sin(x);**

$$f5 := x^2 + \sin(x)$$

> **diff(f5, x);**

$$2x + \cos(x)$$

> **diff(f5, x,x); # derivando f5 duas vezes.**

$$2 - \sin(x)$$

> **f6 := x^3+y^2;**

$$f6 := x^3 + y^2$$

> **diff(f6, x);**

$$3x^2$$

> **diff(f6, y);**

$$2y$$

> **diff(u(x)*v(x),x);**

$$\left(\frac{\partial}{\partial x}u(x)\right)v(x) + u(x)\left(\frac{\partial}{\partial x}v(x)\right)$$

>

Séries de Taylor

A função **series** produz a série de Taylor para funções analíticas. Em geral a resposta é dada em termos de uma expansão de ordem 5.

> **T1 := series(exp(x), x=0); #expansão em torno de x=0.**

$$T1 = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + O(x^6)$$

Em seguida vamos converter a parte principal da série *T1* num polinômio.

> **Poli := convert(T1, polynom);**

$$Poli = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5$$

Podemos observar que o resultado *Poli* acima é uma expressão contendo x . Porém não se trata de uma função cujo argumento é x . Para transformar uma expressão contendo x para uma função de x devemos executar **unapply** .

> **f7 := unapply(Poli, x); #transforma o polinômio Poli numa função f7.**

$$f7 := x \to 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5$$

> **f7(1.0);**

$$2.716666667$$

>

Equações Diferenciais

Resolver equações diferenciais é uma das principais tarefas da computação científica. O assunto é complexo e extenso. Aqui faremos alguns exemplos do comando **dsolve** .

> **ed1 := diff(y(t),t,t) + y(t) = sin(t);**

$$ed1 = \left(\frac{\partial}{\partial t}\left(\frac{\partial}{\partial t}y(t)\right)\right) + y(t) = \sin(t)$$

> **dsolve(ed1, y(t));**

$$y(t) = -\frac{1}{2}\cos(t)\,t + \frac{1}{2}\sin(t) + \_C1\sin(t) + \_C2\cos(t)$$

> **ed2 := diff(y(t),t) = y(t)^2;**

$$ed2 = \frac{\partial}{\partial t}y(t) = y(t)^2$$

> **dsolve(ed2, y(t));**

$$\frac{1}{y(t)} = -t + \_C1$$

A maioria dos problemas de equações diferenciais não podem ser resolvidas analiticamente (de forma exata). O comando **dsolve** possui uma opção de solução numérica. Como exemplo, resolveremos um problema de valor inicial não linear.

> **ed3 := diff(y(t),t)+sin(y(t))=cos(t);**

$$ed3 = \left(\frac{\partial}{\partial t}y(t)\right) + \sin(y(t)) = \cos(t)$$

> **yy := dsolve({ed3, y(0)=0}, y(t), type=numeric);**

$$yy := \mathbf{proc}(rkf45\_x) \ldots \mathbf{end\ proc}$$

Vejamos agora como se trabalha com a solução *yy* acima, que é de fato uma função.

> **yy(0);**

$$[t = 0, y(t) = 0]$$

> **yy(2.1);**

$$[t = 2.1, y(t) = .1267904867182338]$$

>

Notas Finais

Neste capítulo exploramos os seguintes comandos Maple:

diff int Int limit Limit series unapply dsolve

Existem muitos outros comandos para o cálculo de derivadas e integrais. Explore os tutoriais contidos em **?contents** ou **?introduction** , conforme o caso. Para se fazer trabalhos mais específicos com equações diferenciais é fundamental consultar os textos especializados. O leitor poderá começar por experimentar **?dsolve** .

>

3 Gráficos em 2 e 3 Dimensões

Os gráficos de funções de uma ou duas variáveis são produzidos através das funções **plot** e **plot3d** . Algumas expressões de três variáveis podem ser plotadas utilizando-se a opção **implicit** .

> **restart;**

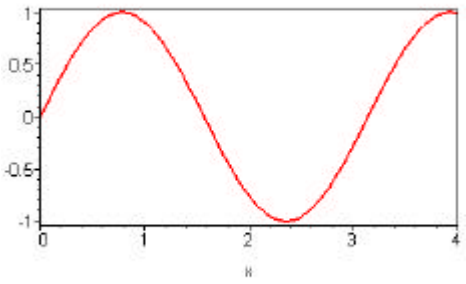
Funções de uma Variável

A sintaxe básica de **plot** é a seguinte:

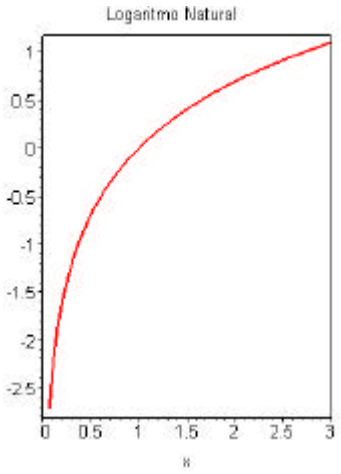
plot(função **,** domínio **,** contra-domínio **,** opções **)** .

O domínio de uma função $f(x)$ é indicada por $x = a..b$. O uso do contra-domínio não é obrigatório e serve para fazer um controle vertical do gráfico. Entre as opções do comando **plot** usaremos o **title** (título) , o **color** (cor) e o **discont=true** (funções com descontinuidades).

> **plot(sin(2*x), x=0..4);**



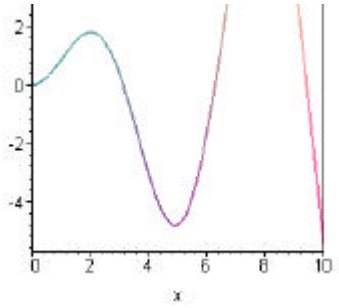
> **plot(ln(x), x = 0..3, title=`Logaritmo Natural`);**



> **f1 := x*sin(x);**

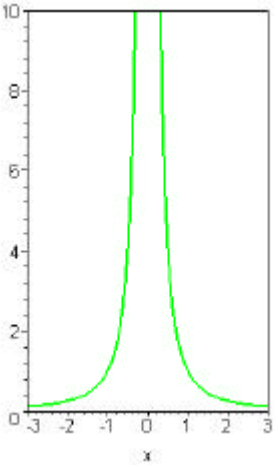
$$f1 := x \sin(x)$$

> **plot(f1, x=0..10, color=blue); # cor azul.**



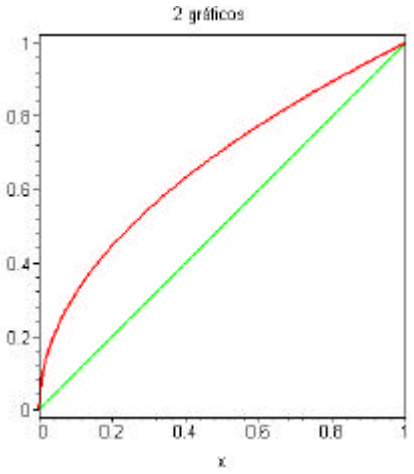
A função $f(x)=\frac{1}{x^2}$ possui uma descontinuidade (de segunda espécie) em $x=0$. Vejamos o seu gráfico com a imagem (eixo vertical) variando de 0 a 10.

> **plot(1/x^2, x=-3..3, 0..10, discont=true);**



Para se plotar dois gráficos simultaneamente escreve-se as duas funções entre chaves **{}** .

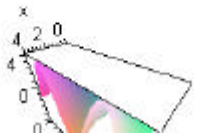
> **plot({x,sqrt(x)}, x=0..1, title=`2 gráficos`);**



Gráficos de Funções de Duas Variáveis

Os gráficos de funções reais de duas variáveis são tridimensionais. O comando Maple utilizado neste caso é o **plot3d** .

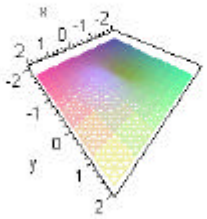
> **plot3d(x*sin(y), x=0..4, y=0..10);**



> **f2 := x*exp(-x^2-y^2);**

$$f2 := x\,e^{(-x^2-y^2)}$$

> **plot3d(f2, x=-2..2, y=-2..2);**



Comandos Especiais em Plots

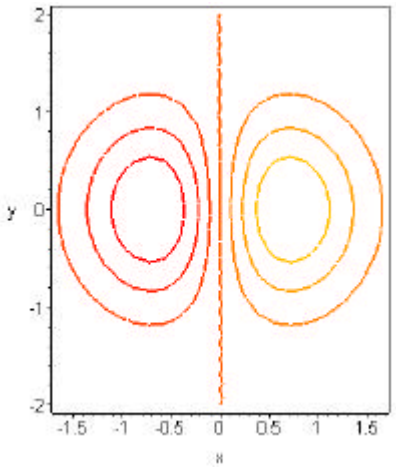
O Maple possui uma biblioteca especialmente dedicada a confecção de gráficos. O acesso a esses comandos se faz executando **with(plots)** . Então poderemos plotar figuras cilíndricas, funções implícitas, coordenadas polares, etc...

> **with(plots);**

[animate, animate3d, changecoords, complexplot, complexplot3d, conformal, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot, pareto, pointplot, pointplot3d, polarplot, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot]

Veremos como utilizar os comandos **contourplot** (curvas de nível) e **spacecurves** (curvas parametrizadas).

> **contourplot(f2, x=-2..2, y=-2..2); # curvas de nível da f2 acima.**

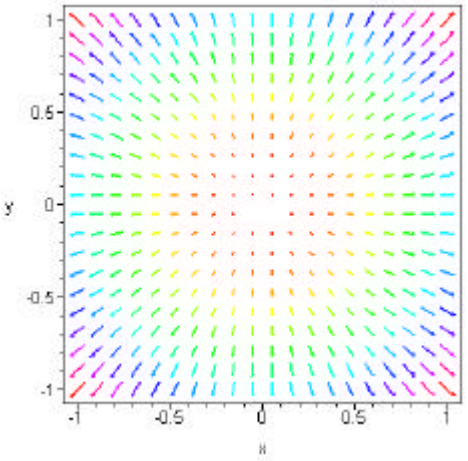


> **spacecurve([cos(t),sin(t),t], t=0..20, title=`Espiral`);**

Espiral

Todos os comandos contidos em **plots** podem ser usados diretamente sem executar o **with(plots)** . Basta saber o *nome* do comando desejado e executá-lo da seguinte forma: **plots[nome]** . Vamos ver um exemplo com **gradplot** (campo gradiente).

> **plots[gradplot](x^2+y^2, x=-1..1, y=-1..1, color = x^2+y^2);**



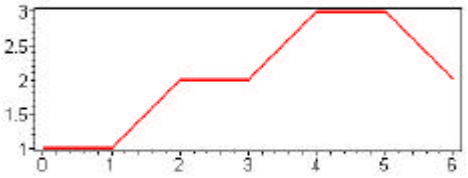
Plotando Pontos

Em experimentos científicos é comum querermos plotar gráficos a partir dos dados obtidos. Esses dados em geral vem em forma de um conjunto finito de pontos. No Maple os dados são inseridos com o comando **seq** (sequências) ou simplesmente colocados em colchetes **[]** .

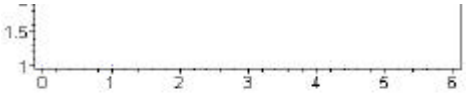
> **Dados := [[0,1], [1,1], [2,2], [3,2], [4,3], [5,3], [6,2]];**

$$Dados := [[0, 1], [1, 1], [2, 2], [3, 2], [4, 3], [5, 3], [6, 2]]$$

> **plot(Dados); # sem parâmetros extras.**



> **plot(Dados, style=point, symbol=circle, color=blue);**



Notas Finais

A produção de gráficos em 2D e 3D é um dos pontos fortes do Maple V. O leitor interessado não deve deixar de consultar o tutorial em **?plots** .

> **?plots**

4 Programação Básica

Atualmente todos os sistemas de computação algébrica possuem recursos para programação. A estrutura básica de programação no Maple é derivado do Algol e do Pascal.

O nosso objetivo é apresentar os elementos essenciais em programação Maple de forma que o leitor possa prosseguir por si mesmo para programas mais avançados. A nossa abordagem não fará uso de nenhum conhecimento prévio em linguagens de programação. Entretanto algum conhecimento em algoritmos matemáticos facilitará a compreenção dos exemplos apresentados.

> **restart;**

Comandos de Entradas e Saídas

Um programa deve começar por ler os dados e terminar por escrever os resultados. Durante as sessões interativas do Maple, a leitura de dados é feita através do comando de atribuição (**:=**). Tabelas de dados armazenados em arquivos também podem ser lidas, com o uso do comando **read** . Execute **?read** para se saber como funciona.

Os comandos específicos para escrever na tela são o **print** (imprimir) e o **lprint** . Esses comandos possuem a seguinte estrutura: **print(** *expressão1* **,** *expressão2* **,** *etc...* **)** . As expressões podem ser valores numéricos ou comentários. Em caso de comentários, esses devem estar entre aspa simples (`).

> **print(`O valor do pi é quase`, evalf(Pi,17));**

$$O\ valor\ do\ pi\ é\ quase, 3.1415926535897932$$

> **m := 2*x;**

$$m = 2\,x$$

> **print(`O cubo m é`, m^3);**

$$O\ cubo\ m\ é, 8\,x^3$$

> **lprint(`O cubo m é`, m^3); # usando lprint.**

O cubo m é 8*x^3

O comando **lprint** possui a mesma sintaxe que o **print** mas escreve os resultados alinhados à esquerda e só usa caracteres ASCII.

>

Comandos de Repetição e Iteração

Durante a concepção de um algoritmo deparamo-nos muitas vezes com situações onde uma certa instrução é repetida várias vezes. Para isso temos o comando **for** . A sua utilização segue um equema **for-do-od** da seguinte forma:

for *j* **from** *início* **to** *fim* **do**

expressões a serem repetidas

od

O esquema é bastante legível se adotarmos as seguintes traduções: **for** (para), **from** (a partir de), **to** (preposição a) e **do** (faça). Como exemplo vamos calcular os quadrados de 1, 2, 3 ,4 e 5.

> **for j from 1 to 5 do**

> **j^2**

> **od;**

1

4

9

16

25

A representação de sequências indexadas no Maple se faz com colchetes **[]** . Vamos escrever os 5 primeiros termos da sequência $y_k = \frac{1}{1+k}$.

```
> for k from 1 to 5 do
> y[k] := 1/(1+k)
> od;
```

$$y_1 := \frac{1}{2}$$

$$y_2 := \frac{1}{3}$$

$$y_3 := \frac{1}{4}$$

$$y_4 := \frac{1}{5}$$

$$y_5 := \frac{1}{6}$$

Em processos iterativos (recursivos) devemos utilizar o conceito da reatribuição dinâmica de variáveis. Digamos que estamos interessados em somar os números de 1 a 100. Para isso começamos com a soma $S = 0$. Na primeira etapa fazemos $S = S + 1$ (agora *S* vale 1). Na segunda etapa fazemos $S = S + 2$ (agora *S* vale 3). Na terceira etapa fazemos $S = S + 3$ (agora *S* vale 6). Na quarta etapa fazemos $S = S + 4$ (agora *S* vale 10), e assim sucessivamente. Ao chegarmos na centésima etapa teremos $S = \sum_{j=1}^{100} j$. Vejamos como essa soma é obtida no Maple.

```
> S:=0;
> for j from 1 to 100 do
> S:=S+j
> od:
> Soma_Final:=S;
```

$$S := 0$$

$$Soma\_Final = 5050$$

>

Agora podemos estudar um exemplo típicamente acadêmico. O problema é o cálculo da raiz quadrada via aproximações sucessivas. O algorítmo, baseado no Método de Newton, é muito simples. Suponhamos que se quer calcular a raiz quadrada de *a* . Então, a partir de um valor inicial r_0 (arbitrário) a raiz quadrada de *a* é o limite da sequência r_k onde $r_k = .5\left(r_{k-1} + \frac{a}{r_{k-1}}\right)$, $k = 1, 2, 3$, ... Vamos obter o valor aproximado de $\sqrt{11.3}$ com $r_0 = 1$, fazendo-se somente 5 iterações.

```
> rr:=1;
```

$$rr = 1$$

```
> for k from 1 to 5 do
> rr:= 0.5 * (rr + 11.3/rr)
> od;
```

$$rr = 6.15$$

$$rr = 3.993699187$$

$$rr = 3.411578079$$

$$rr = 3.361914114$$

$$rr = 3.361547283$$

> **sqrt(11.3);**

$$3.361547263$$

>

Comandos de Seleção

Os comandos de seleção (ou de desvio) são utilizados para se decidir se um certo valor satisfaz ou não uma certa condição. Essas condições são determinadas pelas relações = (igualdade), < (menor que), > (maior que), <= (menor ou igual), >= (maior ou igual) e <> (diferente). Os operadores lógicos são: **if** (se), **elif** (ou se), **else** (ou então) e **then** (então). Trabalha-se com o seguinte esquema: **if-then-elif-then-else-fi** .

> **if 1 = 2 then AZUL**

> **else VERMELHO**

> **fi;**

$$VERMELHO$$

> **if 1 > 10 then print(GRANDE)**

> **elif 1 < -10 then print(PEQUENO)**

> **else print(MEDIO)**

> **fi;**

$$MEDIO$$

>

Procedimentos Maple

Veremos agora uma forma muito prática de se construir pequenos programas "executáveis". No Maple são chamados **procedure** (procedimento). A sintaxe para se contruir procedimentos é a seguinte:

Nome := proc(*argumentos* **)**

instruções contendo os argumentos

end .

É claro que existem muitas outras opções a serem consideradas. O leitor poderá consultar o tutorial em **?proc** . O primeiro procedimento que escreveremos mostra a soma de 2 números dados.

> **Soma := proc(x,y)**

> **print(`A soma procurada é`, x+y)**

> **end;**

$$Soma := \mathbf{proc}(x, y)\ \mathrm{print}(`A\ soma\ procurada\ é`, x+y)\ \mathbf{end\ proc}$$

> **Soma(2,2);**

$$A\ soma\ procurada\ é, 4$$

> **Soma(alpha, beta);**

$$A\ soma\ procurada\ é, \alpha+\beta$$

>

Definindo Funções com Procedimentos

Os procedimentos Maple são na verdade funções dos argumentos de entrada. Logo também podemos utilizar o comando **proc** para definir funções matemáticas. Vejamos como definir a função $f(x) = x^3\sqrt{x}$ de duas maneiras diferentes.

> **f1 := x -> x^3*sqrt(x); # maneira usual.**

$$f1 := x \to x^3\sqrt{x}$$

> **f1(7);**

$$343\sqrt{7}$$

> **f2 := proc(x) x^3*sqrt(x) end;**

$$f2 := \mathbf{proc}(x)\ x\text{^}3*\mathrm{sqrt}(x)\ \mathbf{end\ proc}$$

> **f2(7);**

$$343\sqrt{7}$$

Certas funções podem exigir algum conhecimento em programação para serem definidas. Vamos construir uma função que vale 1 para $x < 0$ e $\cos(10.x)$ para $0 \le x$.

```
> f3 := proc(x)
> if evalf(x) < 0 then 1
> else cos(10.0*x)
> fi
> end;
```

$$f3 := \mathbf{proc}(x)\ \mathbf{if}\ \mathrm{evalf}(x) < 0\ \mathbf{then}\ 1\ \mathbf{else}\ \cos(10.0*x)\ \mathbf{end\ if}\ \mathbf{end\ proc}$$

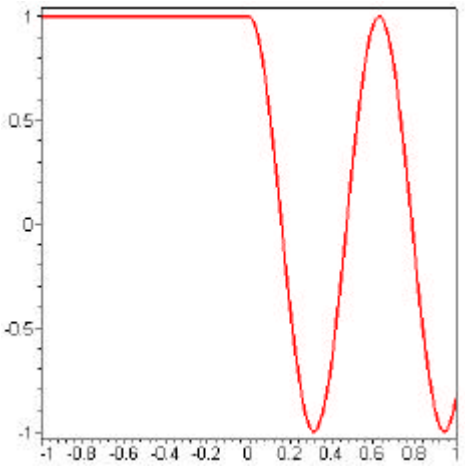
```
> f3(-1);
```

$$1$$

```
> f3(1);
```

$$-.8390715291$$

```
> plot(f3, -1..1);
```



>

Comentários Finais

As técnicas de programação são geralmente objetos de muitos livros e manuais. Entretanto, esperamos que o leitor acredite que a programação Maple é acessível e bastante intuitiva. Veja as referências do Capítulo 5.

5 Dicas e Referências

Funções Especiais (Pacotes)

Os comandos específicos para Equações Diferenciais, Álgebra Linear, Estatística, Gráficos, etc... estão colecionados separadamente em pacotes (*bibliotecas de funções*). Uma lista completa pode ser vista executando-se **?index[package]** . Esses pacotes são carregados com auxílio do comando **with** . Veremos a seguir alguns exemplos do pacote **linalg** , que são específicos para matrizes, vetores e transformações lineares.

```
> restart:
> with(linalg);
```

Warning, new definition for norm

Warning, new definition for trace

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stack, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonds, vecpotent, vectdim, vector, wronskian*]

>

> **A := matrix([[1,2,3],[2,0,1],[0,3,0]]);**

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 1 \\ 0 & 3 & 0 \end{bmatrix}$$

> **b := [5,5,0];**

$$b := [5, 5, 0]$$

> **X := linsolve(A,b); # Resolvendo o sistem AX=b.**

$$X := [2, 0, 1]$$

> **multiply(A,X); # Multiplicando as matrizes A e X.**

$$[5, 5, 0]$$

> **det(A); # Determinante de A.**

$$15$$

> **inverse(A); # Calculando a inversa de A.**

$$\begin{bmatrix} \frac{-1}{5} & \frac{3}{5} & \frac{2}{15} \\ 0 & 0 & \frac{1}{3} \\ \frac{2}{5} & \frac{-1}{5} & \frac{-4}{15} \end{bmatrix}$$

>

Maple no WEB

Relacionamos abaixo alguns sites especializados em Maple V na internet. Todos esses sites possuem artigos técnicos, notas de cursos e programas em computação algébrica.

Stat/Math Center em Indiana University (o meu favorito)

http://www.indiana.edu/statmath/math/maple

CyberMath (Site Oficial)

http://www.cybermath.com

Maple em Los Alamos National Laboratory

http://saaz.lanl.gov/Maple/Maple_Home.html

Maple e LaTeX em Português

http://www.geocities.com/cnumat

Livros Especializados

Os três livros listados abaixo formam o núcleo de toda bibliografia em Maple.

B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, S. M. Watt , **First Leaves:**

A Tutorial Introduction to Maple V , Springer-Verlag, 1992. (ISBN 0-387-97621-3).

B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, S. M. Watt , **Maple V Library Reference Manual** , Springer-Verlag, 1991. (ISBN 0-387-97592-6).

B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, S. M. Watt , **Maple V Language Reference Manual** , Springer Verlag, 1991. (ISBN 0-387-97622-1).

A seguir indicamos alguns livros que são destinados ao ensino universitário.

M. Abell & J. Braselton , **Differential Equations with Maple V** , Academic Press Professional, 1994. (ISBN 0-12-041548-8).

D. Barrow , **Solving Ordinary Differential Equations with Maple V** , Brooks/Cole Publishing Co., 1997. (ISBN 0-5343-4402-X).

W. C. Bauldry, B. Evans & J. Johnson , **Linear Algebra with Maple** , John Wiley & Sons, 1995. (ISBN 0-471-06368-1).

J. S. Devitt , **Calculus with Maple V** , Brooks/Cole Publishing Co. ,1993. (ISBN 0-534-16362-9).

R. J. Lopez , **Maple V: Mathematics and Its Application** , Birkhäuser, 1994. (ISBN 0-8176-3791-5).

Lista de Comandos (em inglês)

Aqui incluímos uma lista de funções e comandos Maple que foi obtida executando-se **?index[functions]** . Cada ítem listado está ligado automaticamente ao seu correspondente tutorial através de hyperlinks.

AFactor AFactors AiryAi AiryBi AngerJ Berlekamp BesselI BesselJ BesselK BesselY Beta C
Chi Ci CompSeq Content D DESol Det Diff Dirac DistDeg Divide Ei
Eigenvals EllipticCE EllipticCK EllipticCPi EllipticE EllipticF EllipticK EllipticModulus EllipticNome EllipticPi Eval Expand FFT Factor Factors FresnelC
FresnelS Fresnelf Fresnelg Frobenius GAMMA GaussAGM Gaussejord Gausselim Gcd Gcdex HankelH1 HankelH2 Heaviside Hermite Im Indep
Interp Inverse Irreduc Issimilar JacobiAM JacobiCD JacobiCN JacobiCS JacobiDC JacobiDN JacobiDS JacobiNC JacobiND JacobiNS JacobiSC JacobiSD
JacobiSN JacobiTheta1 JacobiTheta2 JacobiTheta3 JacobiTheta4 JacobiZeta KelvinBei KelvinBer KelvinHei KelvinHer KelvinKei KelvinKer LambertW Lcm
LegendreE LegendreEc
LegendreEc1 LegendreF LegendreKc LegendreKc1 LegendrePi LegendrePic LegendrePic1 Li Linsolve MOLS Maple_floats MeijerG Norm Normal Nullspace
Power
Powmod Prem Primfield Primitive Primpart ProbSplit Product Psi Quo RESol Randpoly Randprime Ratrecon Re Rem Resultant
RootOf Roots SPrem Searchtext Shi Si Smith Sqrfree Ssi StruveH StruveL Sum Svd TEXT Trace WeberE
WeierstrassP WeierstrassPPrime WeierstrassSigma WeierstrassZeta Zeta abs add addcoords addressof algebraic algsubs alias allvalues anames antisymm applyop
arccos arccosh arccot arccoth arccsc arccsch arcsec arcsech arcsin arcsinh arctan arctanh argument array assign assigned
asspar assume asubs asympt attribute bernstein branches bspline cat ceil chrem close close coeff coeffs coeftayl
collect combine commutat comparray compoly conjugate content convergs convert coords copy cos cosh cost cot coth
csc csch csgn dawson define degree denom depends diagonal diff dilog dinterp disassemble discont discrim dismantle
divide dsolve eliminate ellipsoid entries eqn erf erfc eulermac eval evala evalapply evalb evalc evalf evalfint
evalgf evalhf evalm evaln evalr exp expand expandoff expandon extract factor factors fclose feof fflush filepos
fixdiv float floor fnormal fopen forget fortran fprintf frac freeze fremove frontend fscanf fsolve galois gc
gcd gcdex genpoly harmonic has hasfun hasoption hastype heap history hypergeom iFFT icontent identity igcd igcdex
ilcm ilog ilog10 implicitdiff indets index indexed indices inifcn ininame initialize insert int interface interp invfunc
invztrans iostatus iperfpow iquo iratrecon irem iroot irreduc iscont isdifferentiable isolate ispoly isqrfree isqrt issqr latex
lattice lcm lcoeff leadterm length lexorder lhs limit ln lnGAMMA log log10 lprint map map2 match
matrix max maximize maxnorm maxorder member min minimize minpoly modp modp1 modp2 modpol mods msolve mtaylor
mul nextprime nops norm normal numboccur numer op open optimize order parse pclose pclose pdesolve piecewise
plot plot3d plotsetup pochhammer pointto poisson polar polylog polynom powmod prem prevprime primpart print printf procbody
procmake product proot property protect psqrt quo radnormal radsimp rand randomize randpoly range rationalize ratrecon readbytes
readdata readlib readline readstat realroot recipoly rem remove residue resultant rhs root roots round rsolve savelib
scanf searchtext sec sech select seq series setattribute shake showprofile showtime sign
signum simplify sin singular sinh sinterp solve sort sparse spline split splits sprem sprintf sqrfree sqrt
sscanf ssystem stack sturm sturmseq subs subsop substring sum surd symmdiff symmetric
system table tan tanh testeq testfloat thaw thiele time translate traperror trigsubs
trunc type typematch unames unapply unassign unload unprotect updatesR4 userinfo value vector
verify whattype with writebytes writedata writeline writestat writeto zip ztrans

> **# FIM**