

Maple em 1 aula

Um pacote de software para manipulação matemática. Concorre com o Euler, Mathematica, MatLab, entre outros. Desenvolvido na Univ. de Waterloo no Canadá a partir de 81. Em 88, virou uma empresa (Maplesoft) e em 2009 foi comprada pela Cybernet do Japão. Uma licença full custa caro (uns US\$ 2000), embora existam licenças estudantis mais baratas (uns US\$ 100). Existe uma versão antiga (a V) que é mais ou menos disponível na Internet. É com ela que vamos operar. O produto possui as seguintes funcionalidades

- * computação algébrica (manipulação de expressões simbólicas).
- * computação numérica (solução numérica das expressões acima).
- * computação gráfica.
- * editor de textos matemáticos.
- * competente linguagem de programação *general purpose* ainda que não muito eficiente.

Para instalar a versão V, obtenha-a na Internet e crie uma pasta no seu disco rígido com o conteúdo da biblioteca. Daí execute o WMAPLE.EXE que ficou na pasta BIN.WNT. O produto não exige privilégios para ser instalado nem altera o repositório do windows.

Após chamado, o Maple apresenta o caracter > que indica que ele está pronto para receber e executar comandos. Experimente digitar >3+2; e veja que ele responde 5. Todo comando fornecido precisa ser encerrado por ;. Alternativamente o comando pode ser encerrado por : quando então o Maple não retornará nenhuma resposta. Se você quiser escrever um texto matemático (quando o maple deve se limitar a receber e formatar as linhas) sem executar nada, mude o ambiente pressionando F5. Para voltar à modalidade calculadora, F5 de novo.

Se você entregar a expressão >2/3; o maple vai manter a expressão como sendo 2/3. Para pedir o valor numérico disso, use algum ponto flutuante (como em >2/3.) ou então peça a função evalf(2/3). Para controlar o número de casas da resposta, manipule a variável Digits, colocando nela o valor esperado (por exemplo, Digits:=2;. Faça todos esses exemplos.

Em uma expressão aritmética a prioridade é potência, divisão, multiplicação, adição e subtração, mas esta ordem pode ser modificada usando parênteses como é tradicional. Dois ou mais comandos podem ser colocados na mesma linha (por exemplo 2+4; 4+8; quando o maple resolverá ambos. O contrário também pode ser feito, se um comando for muito grande ele pode ocupar mais de uma linha e ele só será executado quando o Maple detectar o ;.

Funções A seguir algumas funções do maple

raiz quadrada	sqrt(x)	sqrt(36) é 6
fatores	ifactor (x)	ifactor(12) é 2 ² (3)
máximo divisor comum	igcd (x,y)	igcd (33,45) é 3
divisão inteira	iquo (x,y)	iquo (12,7) é 1
resto de	a mod b	10 mod 3 é 1
resto de	irem (x,y)	irem (10,3) é 1
é primo ?	isprime (x)	isprime (12) é false
absoluto	abs (x)	abs (-4) é 4
resultado flutuante	evalf(x)	evalf(2/3)
coseno	cos (x)	cos (Pi/3) é 1/2
seno	sin (x)	sen (Pi/3) é 1/2 √3
tan	tan (x)	tan(Pi) é 0
exponencial	exp(x)	devolve e ^x
logaritmo neperiano	ln(x)	devolve log _e x
atribuição	a := x;	a := 3.33;

Veja a seguir uma lista das operações algébricas mais importantes do maple:

factor Fatora uma expressão algébrica, transformando-a em fatores primos. Exemplo: seja eq1:=8*x³ - 18*x*y² + 20*x²*y - 45*y³;

O factor(eq1) é (3y+2x) (5y+2x) (23y+2x)

expand Expande uma expressão algébrica. Exemplo expand((x+1)⁴); dá como resposta x⁴ + 4 x³ + 6 x² + 4 x + 1

simplify Simplifica uma expressão. Nem sempre é fácil definir o que é uma expressão mais simples. O comando possui um segundo parâmetro opcional que permite fazer suposições sobre as variáveis envolvidas. Exemplo: y:=sqrt(a² * (b + 1)²); Dá como resposta y := √a²(b+1)². Escrevendo simplify(y); fica y := √a²(b+1)². Mas, fazendo simplify(y, assume=nonnegative); a resposta é a(b+1).

subs Permite substituir variáveis por valores ou por outras variáveis em uma expressão. Seu formato é subs(x=exp1, y=exp2, ..., expressão). Por exemplo subs(x=2,x² + y² + z³); fica 4 + y² + z³.

normal Transforma a expressão em *numerador/denominador*, onde ambos são polinômios relativamente primos com coeficientes inteiros.

solve Resolve uma equação, que é uma expressão com um sinal de igual no meio. Se não tem o igual, o maple assume que a expressão dada é igual a zero. A resposta é uma sequência de raízes. Exemplo solve(4*x - 1 = 7*x + 9); A resposta é -10/3.

fsolve Igual ao anterior, mas dando o resultado na forma de números em ponto flutuante.

isolve resolve equações diofantinas (i=inteiro)

msolve resolve equações com congruência do tipo f(x) ≡ b mod p

combine Quando possível combina expressões (sobretudo expressões trigonométricas envolvendo seno e cosseno).

convert Converte uma expressão na outra (veja as opções em ?convert?. Exemplo: > convert(x³/(x²-1),parfrac,x); dá como resposta x + 1/2 (x - 1)⁻¹ + 1/2 (x + 1)⁻¹

collect Converte a expressão em um polinômio geral. Exemplo: > collect((x+1)³*(x+2)²,x); dá 4 + x⁵ + 7 x⁴ + 19 x³ + 25 x² + 16 x

rhs Lado direito de uma equação

lhs Lado esquerdo de uma equação

numer Extrai o numerador de uma expressão

denom Extrai o denominador de uma expressão

evalf Avalia uma expressão usando ponto flutuante

evalc Avalia uma expressão usando aritmética complexa e responde com o formato a+I*b.

evalb Avalia uma expressão booleana (retorna true, false ou FAIL).

Diferença entre expressão e função Suponha a expressão x² + y². Definindo-a como expressão, fica > e:=x²+y²;. Já a definição como função, fica: > f := (x,y) -> x²+y²;. Para avaliar a expressão no ponto x = 1 e y = 2, faz-se: > eval(e,[x=1, y=2]); cuja resposta é 5. A função é > f(1,2); que também dá 5.

Programação-Comando Condicional
Trata-se do comando IF, cujo formato é

if <condição> then ... [else ...] fi;

Ele serve para determinar caminhos que dependem de alguma condição. No caso, se <condição> é verdadeira os comandos que seguem o then são executados. Se a condição for falsa e a cláusula else existir (ela é opcional), os comandos que a seguem é que serão executados. Tudo se encerra com a cláusula end if, devidamente seguida por um ponto-e-vírgula.

Comandos de repetição Existem dois comandos de repetição. São eles o for e o while. Ambos resolvem todas as necessidades em termos de desvio e repetição. O comando for tem o seguinte formato

for <var> from <v1> to <v2> [by <inc>] do ... od;

A variável citada no comando, é inicializada com o valor 1. A seguir, a lista de comandos é executada. A variável é incrementada em 1 e tem o seu valor testado com o valor 2. Se a variável for menor ou igual, a lista de comandos é reexecutada. Quando a variável ultrapassar o valor2, o comando for se encerra. O comando while, também permite repetições, e seu formato é

while <condição> do ... od;

A <condição> é avaliada. Se ela for verdadeira, a sequência de comandos é executada. Quando o end do é encontrado, ocorre um desvio para o while, quando a condição é reavaliada, e assim por diante. Dentro de ambos os laços podem ser incluídos os comandos break e next. O comando break encerra prematuramente o laço, independente de outras condições. Quando break é executado, o próximo comando é o seguinte ao end do. Quando next é executado dentro de um for, a variável é incrementada e a condição testada. Quando dentro de um while, apenas a condição é reavaliada.

Procedimentos É o que em outras linguagens recebe o nome de programa. O formato de um procedimento (programa) em maple é

<nome-procedimento> := proc() ... end;

Para chamar (executar) o programa acima, basta fornecer o comando nome-procedimento(); É comum, o programa receber variáveis externas, que serão conhecidas apenas em tempo de execução. Tais variáveis são citadas dentro do parênteses e são conhecidas pelo programa através de seu valor (passagem por valor). Também é comum que o procedimento precise devolver um determinado valor. Usa-se o comando opcional return <valor>. Se este comando não for citado o procedimento devolve a última coisa que for calculada. Note que esta definição de procedimento é muito parecida com a definição de uma função.

Se a variável de chamada deve ser de um determinado tipo (sob pena de erro), seu nome pode ser citado como (<variavel>::<tipo>). Veja no exemplo:

P := proc(x::posint)
 if isprime(x) then RETURN true;
 else RETURN false; fi;
end;

Deve-se ressaltar que um procedimento pode receber outro procedimento como parâmetro em sua chamada. A quantidade de parâmetros passados para um procedimento pode ser obtida através da variável nargs. A sequência de parâmetros passados para o procedimento é guardada na variável args, que é naturalmente indexada. Maple possui o comando printf, muito similar ao seu homônimo em C, e que permite fazer impressão formatada.

Para você fazer

1. Copie e implemente o programa que calcula os dígitos verificadores de um CPF. Depois ache os DVs do CPF: 970.129.256

2. Ache as raízes do sistema
74x + 28y + 43z + 67w = 10218
5x + 23y + 33z + 20w = 5924
34x + 40y + 37z + 9w = 7990
18x + 92y + 30z + 38w = 12642

1	2.x	2.y	2.z	2.w

