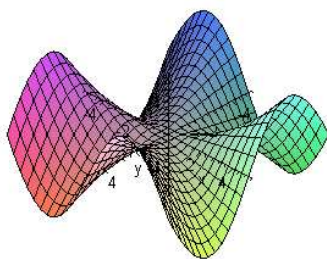


**Departamento de Matemática  
CCET – UFRN**

**Laboratório de Apoio Computacional I**

**Apostila**

**Usando o software Maple**



**Profa. Marcia Maria de Castro Cruz**  
Departamento de matemática – UFRN  
[marcia@ccet.ufrn.br](mailto:marcia@ccet.ufrn.br)

**Natal - RN – Janeiro de 2008**

# SUMÁRIO

<b>Introdução.....</b>	<b>1</b>
<b>Parte 1 – Iniciação ao Maple.....</b>	<b>2</b>
1.1 O software Maple.....	2
1.2 Trabalhando com expressões numéricas.....	2
1.3 Trabalhando com expressões algébricas.....	4
1.4 Trabalhando com intervalos.....	4
1.5 Alguns comandos básicos.....	5
1.6 Pacotes específicos do maple.....	7
1.7 Atividades 1.....	7
<b>Parte 2 – Funções e seus gráficos.....</b>	<b>9</b>
2.1 Atribuindo um nome a uma expressão ou função.....	9
2.2 Funções especiais.....	9
2.3 Funções definidas por mais de uma sentença.....	12
2.4 Mais sobre gráfico de funções.....	12
2.6 Atividades 2.....	14
<b>Parte 3 – Matrizes, determinantes e sistemas de equações lineares.....</b>	<b>17</b>
3.1 Vetores .....	17
3.2 Matrizes.....	17
3.3 Determinante.....	19
3.4 Sistemas de equações lineares.....	20
3.5 Atividades 3.....	22
<b>Parte 4 – Introdução a programação com maple.....</b>	<b>24</b>
4.1 Manipulando estruturas de dados básicas do maple.....	25
4.2 Um jogo: Simulando Gambling.....	27
4.3 Procedimentos.....	28
4.4 Variáveis locais e globais.....	28
4.5 Comandos de repetição: <b>for-do-od</b> .....	29
4.6 O comando <b>while</b> com procedimento.....	31
4.7 Condicional: <b>if – then [else] - fi</b> .....	32
4.8 O comando <b>while – do - od</b> .....	33
4.9 Atividades 4.....	34
<b>Parte 6 – Referências Bibliográficas.....</b>	<b>36</b>

## Introdução

O uso do computador como ferramenta no ensino, é hoje bastante utilizado em quase todas as áreas. A informática é uma das alternativas mais poderosas no ensino moderno, principalmente aqueles que envolvem modelos matemáticos. Foram desenvolvidos vários softwares nessa direção. Um deles é o Maple que tem uma capacidade extraordinária de lidar com os mais diversos conteúdos matemáticos. Neste mini-curso, apresenta-se um pouco do software maple e sua utilização em alguns importantes conteúdos de matemática básica. Os principais objetivos são:

- 1 - Utilizar um ambiente computacional para ensinar o aluno a realizar atividades de matemática manipulando o software **maple**;
- 2 – Ajudar o aluno na compreensão e assimilação de importantes conteúdos de matemática usando o computador como ferramenta;
- 3 - Através da utilização do **maple**, mostrar ao aluno um caminho que certamente facilitará sua vida acadêmica de um modo geral;

Os conteúdos matemáticos abordados serão divididos em quatro partes. Na parte 1 Iniciamos com os primeiros passos da utilização do software maple, apresentando também algumas informações a respeito desse pacote matemático. Na parte 2 vamos apresentar algumas formas de se trabalhar com funções bem como a visualização gráfica de alguns tipos de funções, incluindo animação. Na parte 3 usaremos um pacote especial do maple o LINALG, que é específico para se trabalhar com álgebra linear. Estudaremos vetores, matrizes, determinantes e sistemas de equações lineares. Finalmente, na parte 4, faremos uma breve introdução de programação com o maple.

## Parte 1 : Iniciando o Maple

A seguir, iniciaremos a utilização do Maple, apresentando os primeiros passos de seu conteúdo, para que a partir de então, o usuário possa fazer uso de seus inúmeros recursos de forma realmente eficaz, dentro de suas necessidades matemáticas.

### 1.1 O software Maple

O Maple é uma linguagem de computação que possui quatro aspectos gerais que são:

- computação algébrica
- computação numérica
- computação gráfica
- programação

Estes aspectos, de uma certa forma estão integrados. Veja por exemplo que, a partir de um resultado algébrico, uma análise numérica ou gráfica pode imediatamente ser feita. Em geral, na análise de um problema, várias ferramentas são necessárias.

Os aspectos novos trazidos por esse **software** juntamente com outros sistemas algébricos são a computação algébrica e a programação simbólica. A computação algébrica é uma área que teve um forte desenvolvimento nas décadas de 60 e 70, quando foram encontrados importantes algoritmos para integração analítica e fatoração de polinômios.

### MAPLE

*sistema matemático simbólico interativo, possuindo recursos extraordinários para resolver questões como cálculo algébrico, interpretação de conceitos, visualização gráfica, modelagem de problemas, etc.*

Para escrever textos vá na barra de ferramenta acima e clique com o cursor em " T ". Para escrever expressões matemáticas clique em  $\Sigma$  . Para execução de comandos matemáticos clicar em [ $>$  ] .

*Qualquer comando na utilização do Maple deve terminar com " ; " .*

### 1.2 Trabalhando com Expressões Numéricas

O Maple usualmente trabalha com os números de maneira exata.

```
> (43*5 + 3/11)^2;
```

Observe que o resultado é um número racional. Para obter uma aproximação decimal, devemos usar o comando

**evalf** (*evaluate in floating point*):

```
> evalf(123/71);
```

Observe que o resultado tem 10 dígitos (valor *default*), porém é possível aproximar o resultado com quantos dígitos quisermos, por exemplo, com 40 dígitos:

> **Digits;**

> **Digits := 40;**

Obs. Maple reconhece vários tipos numéricos, dos quais destacamos os tipos *integer*, *fraction*, *float* e *complex*. Ao digitarmos no *prompt* do Maple uma expressão aritmética, por *default*, o valor numérico da expressão é calculado no tipo numérico mais abrangente usado na expressão. Exemplos:

> **23 + 67; 23 + 57.2; 1/2 + 5;**

> **30/9;**

O Maple, por *default*, simplifica frações de inteiros

Listamos a seguir os símbolos das operações aritméticas; um símbolo à esquerda de outro tem precedência na avaliação de expressões. O uso de parênteses é necessário para modificar a ordem de precedência.

<b>!</b>	<b>^ ou **</b>	<b>/</b>	<b>*</b>	<b>+</b>	<b>-</b>
fatorial	potenciação	divisão	multiplicação	adição	subtração

> **4+ 3! ; 3^(2\*4); 2\*3/3\*6;**

Um exemplo de função do Maple:

> **sqrt(28);**

Vemos como o Maple conserva o tipo numérico dos dados de entrada. Por outro lado, é possível provocar um retorno em um tipo numérico desejado. Uma forma é usar a função *convert*.

> **convert(sqrt(28), float);**

Vejamos agora o número de Euler com 40 dígitos:

> **exp(1.);**

Números irracionais:

> **((2+sqrt(3))/2)^2;**

Vamos ver agora um exemplo mais elaborado:

> **sin(2\*Pi\*n)/5!;**

Pedimos para o Maple calcular  $\sin(2 \pi n)$  e dividir por fatorial de 5. Podemos pensar que o resultado é zero, já que o seno de um múltiplo de  $\pi$  é zero. Porém,  $n$  é uma letra sobre a qual nada foi dito sobre ela.

Para o Maple  $n$  é uma variável como qualquer valor real. Isso pode ser mudado com o seguinte comando:

> **assume(n, integer);**

> **sin(2\*Pi\*n)/5!;**

> **cos(Pi\*n);**

De agora em diante, o Maple tratará  $n$  como uma variável inteira. Ela passa a ser mostrada na tela com um til, da forma  $n\tilde{}$ , para o usuário saber que essa variável tem uma determinada propriedade atribuída pelo comando `assume`.

Antes de continuar, vamos reiniciar o trabalho com o comando

**> restart**

Assim, o Maple reinicia como se nada tivesse sido feito antes e a variável  $n$  agora volta a ser uma letra que pode assumir qualquer valor.

Vamos ver um exemplo com números complexos. Queremos encontrar as raízes cúbicas de:

**> z := (-8)^(1/3);**

Podemos ver que o Maple nada fez, ou melhor, não houve nenhuma simplificação do resultado. Dessa forma, vamos pedir explicitamente:

**> simplify(z);**

A variável  $I$  representa  $\sqrt{-1}$ .

Outro exemplo:

**> (3+4\*I)/(1+I);**

### 1.3 Trabalhando com Expressões Algébricas

Escrevendo uma expressão algébrica

**> E := sqrt(5) x^3 + 2\*x^2/3 - 1/4;**

Ex.1

**> (5\*x^3-2\*x^2+3)/x;**

Ex.2  $9\sqrt{x+1} - 4x^5 - \frac{2}{x^3} + \frac{2}{3}$

**> 9\*sqrt(x+1) - 4\*x^5 - 2/(x^3) 2/3;**

### 1.4 Trabalhando com Intervalos

Na descrição de intervalos devemos estar atentos a algumas expressões que usualmente usamos em matemática, mas que o pacote matemático do Maple V não é capaz de entender.

Por exemplo, pensemos no intervalo  $[\infty, 3]$ , que significa todos os números reais menores que 3. Como grafar isso no Maple ?

**> x<=3;**

Observe o que ocorre quando queremos escrevermos o sinal igual antes da desigualdade:

**> x=<3;**

é importante que grafemos sempre a desigualdade antes do igual. Agora observe o que ocorre quando queremos grafar o intervalo de números reais entre -3 e 5.

```
> -3<=x<=5;
```

Para grafar intervalos deste tipo é necessário que utilizemos os conectivos lógicos **and** e **or**.  
Por exemplo,

```
> 3<=x and x<=5;
```

```
> x<=-3 or x>=5;
```

### 1.5 Alguns comandos básicos:

**Expand:** O comando "**expand**" serve para expandir expressões incluindo também expressões na forma trigonométrica, logarítmica etc.

Exemplos:

```
> (x-1)^5;
```

```
> expand((x-1)^5);
```

```
> cos(alpha+beta) = expand(cos(alpha+beta));
```

```
> ln(3*x^2*y^2) = expand(ln(3*x^2*y^2));
```

Um outro efeito do comando **expand** se refere a expressões com denominador. Ele coloca o denominador embaixo de cada numerador, sem expandir o denominador:

```
> expr := (x+y)^2/(a+b)^2;
```

```
> expand(expr);
```

**factor:** Fatora expressões

Ex.

```
> factor((x-1)^5);
```

**Simplify:** O comando **simplify** é um comando geral de simplificação. É possível aplicar as regras de simplificação de determinadas funções de maneira selecionada. Para isso deve-se dar o nome da função em questão no segundo argumento do comando **simplify**, que pode ser um dos seguintes nomes: **trig**, **hypergeom**, **radical**, **power**, **exp**, **ln**, **sqrt**, etc.

Exemplos:

```
> simplify((x^2-16)/(x+4));
```

```
> simplify((sin(x)^2-cos(x)^2)^2,trig);
```

```
> expr := (sin(x)^3 + cos(x)^3)*exp(a)/exp(a+b);
```

```
> simplify(expr);
```

É possível também dizer ao comando **simplify** que as variáveis obedecem as certas restrições, que são as mesmas usadas no comando **assume**. Por exemplo, a raiz quadrada do produto de vários termos só é o produto das raízes quadradas se os termos forem reais e positivos. Vejamos:

```
> expr := simplify(sqrt(x^2*y^2));
```

```
> simplify(expr, assume=nonneg);
```

**Combine**

o comando **combine** requer maior atenção que os comandos anteriores, pois para usá-lo com eficiência é necessário conhecer as opções que devem ser fornecidas como segundo argumento. A sintaxe é: **combine( expressão, opção )**. A opção pode ser: exp, ln, power, trig, Psi, polylog, radical, abs, signum, plus, atatsign, conjugate, plot, product ou range entre outras. A opção trig engloba todas as funções trigonométricas e a opção power, expressões que envolvem potenciação.

Exemplos:

```
> (x^a)^2*x^b = combine((x^a)^2*x^b, power);
> 4*sin(x)^3 = combine(4*sin(x)^3, trig);
> exp(x)^2*exp(y) = combine(exp(x)^2*exp(y), exp);
```

### Convert

A sintaxe deste comando é: **convert( expressão, tipo )**

onde tipo pode ser um dos seguintes nomes: trig, tan, ln, exp, expln, expsincos, rational, parfrac, radians, degree, GAMMA, factorial, entre outros, no caso de ser tratar de conversão de uma expressão algébrica em outra expressão algébrica. Vejamos alguns exemplos:

```
> expr := (1+I)*(exp(-I*x)-I*exp(I*x))/2;
> convert(cosh(x),exp);
> convert(arcsinh(x),ln);
> binomial(n,k) = convert(binomial(n,k), factorial);
```

### Solve

Com esse comando obtém-se a solução de equações

A sintaxe é: **solve(equação,variável)**

Ex.

```
> solve(x^3-2*x^2-x,x);
```

Com o comando solve você pode também solucionar sistemas de várias equações e várias incógnitas

#### Exemplo:

Nesse caso convém atribuir um nome as expressões, para depois pedir a solução

```
> eqns := {x+y+2*z=1, 3*x+2*y=2, x-2*y-z=0};
> solve( eqns );
```

Do mesmo modo que o **solve(exp.,var.)** é utilizado para resolver equações (achando as suas raízes) ele também pode ser usado para resolver desigualdades. Por exemplo:

```
> solve(x-3>=5,x);
> solve(x^2-1>=-1,x);
> solve(x^2-1>=0,x);
> solve(x^2-1>=0 and x^2-1<2,x);
> solve(x^2-1<-2,x);
```

Observe que no caso de não haver solução, ou seja, o conjunto solução ser o conjunto vazio, nada é produzido pelo Maple. Portanto, cuidado com a leitura das respostas obtidas

### Subs: Esse comando serve para substituir subexpressões em expressões

Exemplos:



```
> subs( x=1/3, x^3+2*x^2-x+5);
> subs(x=a^2+1,3*x+21);
```

**Sum** - Calcula somatórios definidos e indefinidos

Exemplos:

```
> sum('k^3+2', 'k'=0..4);
```

```
> sum('k^3+2', 'k'=0..n)
```

```
>
```

*Usando o símbolo de somatório*

Ex.

```
> Sum('a[k]', 'k'=0..10);
```

Para obter o resultado da soma , toma-se o s minúsculo.

```
> sum('a[k]', 'k'=0..5);
```

Ou ainda

```
> Sum('a[k]', 'k'=0..5)=sum('a[k]', 'k'=0..5);
```

## 1.6 Pacotes do maple

Alguns comandos do maple são de uso específico e por isto são agrupados em pacotes. A forma de disponibilizar destes pacotes é usando o comando with(nome do pacote).

Exemplos:

With(student); With(plots); With(linalg); With(DEtools);

Como podemos ver, cada um desses pacotes diz respeito a determinados conteúdos matemáticos. Por exemplo, **linalg** é um pacote específico para se trabalhar com álgebra linear. **DEtools**, pacote para trabalhar com equações diferenciais.

## 1.7 Atividades 1

1) Escreva as seguintes expressões numéricas:

$$\text{a) } \frac{20 - 5\sqrt{6}}{\pi - 1} ; \quad \text{b) } \frac{\frac{1}{3} - \frac{4}{7}}{3e^2 + 2\pi} ;$$

2) Escreva as seguintes expressões algébricas:

$$\text{a) } x^4 + 3x^3 + \sqrt{x+3} + 1 ; \quad \text{b) } \frac{3x^5 - 7x^3}{e^x + \frac{3}{x}}$$

2) Use o comando **expand** para expandir a expressão  $(y^3 + 5)^2 (2y - 1)^3$ ;

3) Use o comando **simplify** para simplificar a expressão  $(y^2 - 25)^2 / (y - 5)$

4) Use o comando ***solve*** para resolver as seguintes equações

a)  $x^3 - x^2 - 6x = 0$  ;    b)  $3y^4 + 2y^3 - 21y^2 + 4y + 12 = 0$ ; c)  $x(x^2 + 5)^2 = 1.5$

5) Com o comando ***solve*** resolva o sistema dado pelas equações

$$x - 2y + 3z = 5; \quad 4x + 5y - z = 9; \quad 2.5x + 4.2y - 7z = 1$$

6) Com o comando ***subs*** encontre o valor da expressão  $(y^3 + 5)^2(2y - 1)^3$  para  $y = -2$

7) Com o maple descreva os seguintes intervalos:

a)  $[3, 7]$ ; b)  $(-2, 3]$ ; c)  $(-10, 12)$ ;  $(-\infty, 0]$ ;  $(1, \infty)$

## Parte 2

### Funções e seus Gráficos

#### 2.1 Atribuição de um nome a uma expressão ou uma função

Uma expressão não precisa necessariamente ter nome, porém algumas vezes é melhor nomeá-las, pois isso muitas vezes facilita o trabalho, principalmente quando precisamos utilizar-la diversas vezes.

##### Exemplo

```
> eq:= 3*x^2 + 2*x-1 = 0;
```

Ex. 1 Escrever a função  $f(x) = 3x^5 - 2x^3 + e^x + 1$

```
> f(x):=3*x^5 - 2*x^3 + exp(x) + 1;
```

```
> s:= (t^2 - 4)^2 * sqrt(2t - 3)
```

#### 2.2 Funções especiais

##### Funções polinomiais

Um polinômio de grau  $n$  é uma função da forma

$$P(x) = a_n x^n + a_{n-1} x^{(n-1)} + \dots + a_2 x^2 + a_1 x + a_0$$

onde os coeficientes  $a_0, a_1, \dots, a_n$  são números reais conhecidos com  $a_n \neq 0$  e  $n$  é um número natural. Os exemplos mais simples de polinômios são as funções potências da forma  $1, x, x^2, x^3, \dots, x^n$

##### Exemplos:

1) Com o comando plot do maple `maple`, vamos esboçar os gráfico, em um mesmo sistema de eixos dos polinômios de grau ímpar  $x, x^3, x^5$  e observar as características peculiares a essas funções.

2) Agora vamos fazer o mesmo com os polinômios de grau par  $x^2, x^4, x^6$

3) Consideremos o polinômio  $P(x) = x(x-1)(x+3)(-x+5)(2x-3)$

a) Use o comando ***expand*** para expandir o polinômio P ;

b) Use o comando `solve` para achar as raízes do polinômio;

4) Observe os gráficos das funções  $x^9$  e  $x^9 + 3x^6 + 7x^4 - x$  traçados na mesma janela.

```
> plot([x^9,x^9+3*x^6+7*x^4-x],x=-3..3,y=-40..40);
```

Observe agora o que acontece com estes gráficos quando os valores de  $x$  aumentam.

```
> plot([x^9,x^9+3*x^6+7*x^4-x],x=-100..100);
```

### Exercícios

1) Examine, abaixo, os gráficos das funções  $x^{10}$  e  $x^{10} + 3x^7 + 7x^4$ , traçados na mesma janela. Proceda como no exemplo 4.

2) Escreva um polinômio de grau 7 que possua raízes reais e complexas; Calcule essas raízes com o maple.

### Funções Racionais

Os polinômios podem ser, evidentemente, multiplicados por constantes, somados, subtraídos e multiplicados, e os resultados serão novamente polinômios. No entanto, se dividirmos polinômios nem sempre obteremos outro polinômio. Esse quociente é chamado de função racional, isto é, uma função racional  $f(x)$  é do tipo

$$f(x) = \frac{p(x)}{q(x)}$$

onde  $p(x)$  e  $q(x)$  são polinômios. Se o denominador  $q(x)$  for uma constante não nula, esse quociente será ele próprio um polinômio. Assim, os polinômios estão incluídos entre as funções racionais.

Evidentemente, nos pontos onde  $q(x) = 0$ , a função  $f$  não está definida e portanto, o maior domínio de uma função racional é constituído pelo conjunto dos números reais excetuando-se esses pontos. Os zeros de  $q(x)$  são chamados de polos ou pontos singulares da função  $f$ .

Exemplo 1: Estudar o comportamento das funções  $\frac{1}{x}$  e  $\frac{1}{x^2}$

a) Plotar o gráfico

**plot(1/x^2,x=-2..2,y=-1..10);plot(1/x,x=-2..2,y=-10..10);**

Repare que, nos dois casos, o polo das duas funções é o ponto  $x = 0$  e que os valores das duas funções se tornam ilimitados quando  $x$  se aproxima de 0. A reta  $y = 0$  é uma assíntota vertical ao gráfico das funções.

**Exercício:** Considere a função  $g(x) = \frac{x}{x^2 - 1}$ .

Estude o comportamento desta função, considerando os seguintes itens

- Qual seu maior domínio?
- Quais seus pólos?
- Plote o gráfico e observe o comportamento de  $g$  em torno de seus pólos.

## Funções Exponenciais e logarítmicas

Exponencial de base  $a$ :  $y = a^x$

> **y=3^(x)**

Exponencial de base  $e$ :  $y = e^x$

> **y:exp(x)**

Logaritmo de base  $a$ :  $y = \log_a x$

> **y=log[a](x);**

Logaritmo de base  $e$ :  $y = \ln x$

> **y=ln(x)**

### Exemplos:

1 - Calcule o valor de  $e^x$  para valores  $x=5$ ;  $x=17$ ;  $x=295$ ;  $x=-5$ ;  $x=-123$  e  $x=-3/11$ ;

2 - Calcule o valor de  $\ln(x)$  para valores  $x=10$ ;  $x=27$ ;  $x=300$ ;

## Funções Trigonométricas

Considere uma circunferência  $C$  centrada na origem e de raio unitário. Seja  $P$  um ponto pertencente a  $C$  situado a " $t$ " radianos do ponto  $(1,0)$  medidos no sentido anti-horário se " $t$ " for positivo e no sentido horário se " $t$ " for negativo. Considere a projeção perpendicular  $P_x$  do ponto  $P$  sobre o eixo dos " $xx$ " e a projeção perpendicular  $P_y$  de  $P$  sobre o eixo dos " $yy$ ". Definimos seno do número real " $t$ " como sendo a medida orientada da origem  $(0, 0)$  ao ponto  $P_y$  e definimos cosseno de " $t$ " como sendo a medida orientada da origem  $(0, 0)$  até o ponto  $P_x$ .

Na animação abaixo mostramos um ponto  $P$  na cor cinza percorrendo a circunferência unitária e completando um ciclo a cada  $2\pi$  radianos. Mostramos também a projeção  $P_y$  em azul e a projeção  $P_x$  em vermelho.

Compile o comando abaixo e execute a animação no modo cíclico. Caso necessário diminua a velocidade de exibição.

```
> with(plots):filme:=seq(display([
plot([cos(k),sin(k),k=0..2*Pi],color=grey,axes=normal,thickness=1),
pointplot([cos(t*2*Pi/50),0]),symbol=box,color=red),
pointplot([cos(t*2*Pi/50),sin(t*2*Pi/50)],symbol=circle,color=black),
pointplot([0,sin(t*2*Pi/50)],symbol=box,color=blue)
]),t=0..50):
display(filme,insequence=true,scaling=constrained);
```

Observando os deslocamentos de  $P_y$  e de  $P_x$  verificamos que as funções seno e cosseno têm domínio igual a  $\mathbb{R}$  e suas imagens pertencem ao intervalo  $[-1, 1]$ .

Para que o Maple mostre parte dos gráficos de seno e cosseno podemos compilar o comando abaixo.

```
> with(plots):plot([sin(x),cos(x)],x=0..2*Pi,color=[red,blue]);
```

### 2.3 Funções definidas por mais de uma sentença

Para se escrever com o Maple uma função definida por mais de uma sentença, usa-se o comando `piecewise`.

Exemplos

```
> p:= piecewise(x<=0,x+2,x>0,3);
> q:=piecewise(x<=2,-4*x,x>2,-4*x^2+x+1);
> s:=piecewise(x>0,x);
> f:= piecewise(x<>3,abs(x-3)/(x-3),x=3,1);
```

**Exercício:** Escreva a seguinte função com o comando `piecewise`

$C = (x < -4, 0, -4 \leq x \text{ and } x < -2, 4x + 16, x > -2 \text{ and } x \leq 1, 2x^2, 1 < x \text{ and } x \leq 4, 5 \sqrt{x+3} - 8, 7);$

### 2.4 Mais sobre gráfico de Funções

Uma das grandes utilidades de software Maple, é sem dúvida plotar gráficos de funções, em duas e em três dimensões. O esboço de um gráfico, em algumas vezes é um trabalho desgastante quando feito manualmente e em alguns casos até mesmo foge do nosso alcance obtê-los. Usando o Maple podemos obter gráficos de duas ou três dimensões; plotar vários gráficos em um mesmo sistema de eixos, obter gráficos definidos por várias sentenças, gráficos com animação, etc.

A seguir, vamos apresentar alguns exemplos de como obter gráficos de funções de uma variável.

Em geral, gráficos bi-dimensionais no Maple são obtidos através do comando "plot" e a sintaxe mais simples é: **plot(expressão, x = a..b)** ou **plot(expressão, x = a..b, y = c..d)**

Exemplos:

1) Obtenha o gráfico da função  $y = x^3 - 6x^2 + 2x^2 - 3x$

Para obter o gráfico devemos considerar um intervalo de definição da função

```
> plot(x^3-6*x+2*x^2-3*x,x=-5..5);
```

Observe que o gráfico pode ser melhor visualizado se especificarmos também um intervalo para o eixo y.

```
> plot(x^3-6*x+2*x^2-3*x,x=-5..5,-10..20);
```

**Observação.** Conforme você viu, para plotar um gráfico com o Maple, é necessário que especifiquemos um intervalo para x. Algumas vezes isso é suficiente, porém, em alguns casos, se não especificarmos também uma janela (ou intervalo) para y, não teremos uma boa visualização do gráfico. Mas, ATENÇÃO! cuidado com esses intervalos, eles são fundamentais para uma boa visualização gráfica.

Ex.2 Gráficos de funções trigonométricas

```
> plot(cos(3x+5), x=0..2*Pi);
> plot(tan(x), x=-Pi..Pi,-20..20);
```

Como plotar vários gráficos em um mesmo sistema de eixos

Ex.3 Gráfico das funções  $f(x) = \sin(x)$  e  $g(x) = \cos(x)$  no intervalo  $[-2\pi, 2\pi]$

```
> plot({sin(5*x),cos(7*x)},x=-2*Pi..2*Pi);
```

Ex. 4

```
> plot({exp(x),ln(x),x},x=-5..5,-10..10,color=[red,blue,black]);
```

Ex.5 Um gráfico de uma função racional

$$f(x) = \frac{x+2}{x-1}$$

```
> plot((x+2)/(x-1),x=-5..5,y=-10..10);
```

Observe que o Maple mostra a assíntota vertical, isto é, como a função não está definida para  $x = 1$ , então existe uma assíntota (reta) vertical dividindo o gráfico em duas partes.

Gráficos de funções definidas por mais de uma sentença

Exemplos: Plot os gráficos das funções p, q, s e f dos exemplos do item 4.1

```
> plot(p,x=-2..3,0..4);
```

Observe bem as janelas e obtenha os outros gráficos

O que acontece com o gráfico de uma função quando somamos uma constante?

**Exemplo:**

Plote o o gráfico das funções  $y = x^2$ ,  $y = x^2 + 1$  e  $y = x^2 - 1$  no intervalo  $[-2, 2]$  em um mesmo sistema de eixos.

### Interseção entre curvas

Sejam  $y = f(x)$  e  $y = g(x)$ . Para se determinar o ponto de interseção entre as curvas, deve-se calcular o ponto em que  $f(x) = g(x)$ .

**Exemplo:** Calcule o(s) pontos de interseção entre as funções  $y = x^2 - x - 2$  e a reta  $y = 2x + 1$  e plote o gráfico das duas em um mesmo sistema de eixos

### Gráficos de funções implícitas:

Em geral as funções são da forma  $y = f(x)$ , entretanto, algumas vezes  $y$  não está implícito..

Exemplos:

a)  $x^2 + y^2 = 4$

b)  $3x^2y - x^2 + 2y = 1$ ;

b)  $\sin(x + y) = 2x$

Para obter gráficos de funções implícitas, é necessário chamar o pacote **with(plots)** e usar o comando "**implicitplot**".

**Exemplo:**

obter o gráfico de  $x^2 + y^2 = 4$

A sintaxe é with(plots):

```
> implicitplot(x^2 + y^2 = 4, x = -3..3,y = -3..3);
```

Podemos também obter mais de um gráfico em um mesmo sistema de eixos

Ex.

```
> implicitplot({x^2 - y^2 = 1, y = ln(x), y^2 - exp(2*x) = 0}, x = -Pi..Pi, y = -Pi..Pi);
```

### Importantes considerações

Quando plotamos gráfico com o maple, podemos usar a seguinte sintaxe:

```
> plot( f(x), x = a .. b, opções)
```

Essas opções variam de acordo com a necessidade do gráfico. Por exemplo,

```
> plot([sin(x), x-x^3/6], x=0..2, color=[red,blue], style=[point,line]);
```

Podemos também colocar um título no gráfico. Por exemplo, vamos fazer o gráfico da função de ordem zero com o título "Função Esférica de Bessel", com o nome "j(0,x)" no eixo vertical e controlando o número de marcações no eixo x e no eixo y:

```
> plot(j(0,x), x=0..15, `j(0,x)` = -0.5..1, xtickmarks=4,  
> ytickmarks=6, title=`Funcao Esferica de Bessel`);
```

### Gráficos com animação

Para obter gráficos com animação, devemos chamar o pacote **with(plots)**

#### Exemplos:

Gráfico do seno com animação

Ex.

```
> with(plots):
```

```
> animate( sin(x*t), x = -10..10, t = 1..2, frames = 50);
```

Outro exemplo

A função  $y = e^{(2x)}$

```
> animate(exp(2*x*t), x=-10..10, y=-20..20, t = 1..2, frames=50);
```

## 2.6 Atividades 2

1) Escreva a função  $p(x) = \frac{3^x + 3(x^2 + 10)}{3 - x}$  e calcule o valor dela para  $x = -1/3, 0, 35/23$

e  $\frac{\pi}{2}$ . Para obter o resultado em decimal, use o comando evalf.

2) Escreva a expressão  $L(x) := \ln(x)$ ; e com o comando subs calcule o valor de ln para  $x = 1, x = 3, x = 10$ . O que acontece quando você tenta calcular o valor de  $\ln(0)$ ?

3) Com o comando evalf, calcule o valor de  $\sin(\frac{\pi}{2})$ ,  $\cos(\frac{\pi}{4})$ ,  $\tan(\frac{\pi}{2})$ ;  $\sin(3)$ .

4) Escreva a função exponencial de base 5 e calcule alguns valores para  $x$ .  $x$  pode ser negativo? Se assim for, calcule valores da função para  $x$  negativo. O que você observa?



5) Escreva a função  $f(x) = e^{(-x)} + 3^{(2x)} - \left(\frac{1}{2}\right)^x$  e use o comando subs para calcular o valor da função para  $x = \sqrt{2}$ .

6) Com o comando solve, resolva as equações:

(c)  $7^x = 1/2401$

(d)  $5^{3x-7} = 25$

(e)  $2^{6x+7} = 128$

(f)  $3^{x+2} - 3^{x+1} + 3^{x-2} + 3^{x-3} = 1494$

7) Considere a função  $f(x) = \frac{10x + 5}{x^4 + x^3 + 2x - 4}$

a) Com o comando solve resolva a equação do denominador;

b) que relação tem o resultado acima com o domínio da função?

8) Plote o o gráfico das funções das funções  $y = x^2$ ,  $y = x^2 + 3$  e  $y = x^2 - 3$ , em um mesmo sistema de eixo com x variando de 0 a 2.

9) a) Defina a função  $|x|$  usando o comando **piecewise** e obtenha seu gráfico com maple.

b) faça o mesmo com a função  $|2x + 3|$ .

10) Use o comando **piecewise** para escrever a função definida por  $x^2 - 1$  se  $x < 0$ ,  $x - 1$  se  $0 \leq x$  e  $x < 1$  e 2, se  $1 \leq x$ .

11) Calcule o(s) ponto(s) de interseção entre as retas  $y = 1 - x^2$  e  $y = \sqrt{x + 1}$  e visualize-os com maple.

12) Plot o gráfico das funções tangente, cotangente, secante e cossecante no intervalo de 0 a  $2\pi$ . Exemplo:

**> plot( sec(x), x= 0..2\*Pi, y= -3..3, title='Secante');**

13) Considere a função  $p(x) = -7x^3 - 2x^4 + 2x^2 + 13x - 6$

a) Com o comando factor, fator p(x);

b) Com o comando solve, encontre suas raízes;

c) Encontre os intervalos em que  $p(x) > 0$  e os intervalos em que  $p(x) < 0$ .

14) Com o comando piecewise,

a) escreva uma função definida do seguinte modo:

Para  $x < -1$  e  $-2 \leq x$  ela vale  $-x^2 + 4$ ; para  $-1 \leq x$  e  $x < 0$  ela vale  $-3x$ ; para  $0 \leq x$  e  $x < 1$  ela vale  $3x$ ; para x entre 1 e 2 ela vale  $-x^2 + 4$ .

b) Invente uma função definida por três sentenças. Primeiro escreva-a manualmente, depois escreva-a com o Maple.

15) Com o comando **plot**, obtenha o gráfico

a) do polinômio do exercício 7 em uma janela em que seja possível visualizar as raízes;

b) das funções que você criou no exercício 8 e 9 em janelas convenientes de x e de y.

17) Obtenha o gráfico das funções  $y = x^2$ ,  $y = (x + 1)^2$ ,  $y = (x - 1)^2$ ;

18) Use sua imaginação!!!

a) Encontre duas retas que se interceptam em um ponto e duas que sejam paralelas.

b) Escreva uma função quadrática que não possua raízes reais e cuja parábola tenha concavidade para cima e outra que possua duas raízes reais e que tenha concavidade voltada para baixo.

c) Invente uma função racional cujo domínio seja  $x \neq 1$ .

d) Escreva um polinômio de grau 5 que tenha 0, 1 -2, 3 e -2.5 como raízes e plote seu gráfico

19) Plote o gráfico das funções abaixo em janelas convenientes.

a)  $f(x) = \frac{x^3 + 3x^2 - x - 3}{x^2 + 4x + 3}$  ; b)  $g(x) = \frac{x^2 - 3x - 4}{x^2 - 1}$

20) Considere a função  $f(x) = \cos(x)$

a) Obtenha os gráficos de  $f(x)$ ,  $f(2x)$ ,  $f(3x)$  em um mesmo sistema de eixos;

b) Use o comando `animate` (veja aula 5) para visualizar o gráfico da função cosseno com animação considerando  $x$  variando de -5 até 5,  $t$  de 1 até 3 e frames = 100.

c) Com o comando **`animate`**, obtenha o gráfico da função  $y = 1 + 2^x$  em uma janela conveniente.

21) Use o comando `implicitplot` para plotar os seguintes gráficos:

a) Círculo trigonométrico de raio 3;

b) Elipse de semieixos 2 e 4 e do círculo de raio 2 em um mesmo sistema de eixos.

## Parte 3 : Vetores, Matrizes, Determinantes e Sistemas de Equações Lineares

### 3.1 Vetores

Definindo vetores

Para definirmos vetores utilizamos o comando `array` ou o comando ***vector***:

`E := array (1..5, [1,2,3,4,5]);` define um vetor com 5 elementos; e

`F := vector ([1,2,3,4,5]);` também define um vetor com 5 elementos; e

`G := vector (8, x -> x^3);` define um vetor com 8 elementos onde cada posição recebe o cubo do n° da posição.

#### Exemplos:

`with(linalg):`

`> v := vector([3,0,1]); v[3];`

`> u:=array(1..3,[1,-2,3]); u[2];`

`> w:=linalg[vector](4,[1,x,1/x,sqrt(x)]);`

Definindo um vetor 2D ou 3D dadas as suas coordenadas:

`> a := vector( [3,4] );`

`> b := vector( [2,3,4] );`

Obtendo uma das coordenadas:

`> a[1];`

`> b[3];`

Definindo um vetor 2D ou 3D literal, para definição posterior das coordenadas:

`> b:= vector(3); print(b);`

Obtendo o vetor G de 8 coordenadas, descrito acima

`> G:=vector(8,x->x^3);`

### 3.2 Matrizes

Definindo matriz

Para definir matrizes utilizamos o comando `matrix` ou o comando `array`:

Do mesmo modo que vetores, existe mais de uma maneira de se definir uma matriz com o `maple`

Exemplos:

`A := matrix ([[1,2],[3,4]]);` define uma matriz quadrada de ordem 2 onde a 1ª linha tem os números 1 e 2 e a 2ª linha, os números 3 e 4;

`B := matrix (2, 4, [1,2,3,4,5,6,7,8]);` define uma matriz de duas linhas e quatro colunas;

`C := array (1..3, 1..2, [[1,2],[3,4],[5,6]]);` define uma matriz de três linhas e duas colunas; e

`H := matrix (3, 3, (i,j) -> i * j);` define uma matriz quadrada de ordem 3 onde cada elemento recebe o produto do nº da sua linha com o nº da sua coluna.

Lembrar que: Para trabalharmos com vetores e matrizes, devemos sempre chamar o pacote **linalg**.

Sendo A uma matriz qualquer, denotaremos um elemento da matriz A por  $a_{i,j}$  onde i corresponde a linha e j a coluna da matriz.

`> with(linalg):`

`> A:=matrix([[a[1,1],a[1,2]],[a[2,1],a[2,2]]]);`

`> A[2,1];` retorna o elemento que fica na linha 2 e coluna 1

`A := matrix ([[a,b],[c,d]]);` define uma matriz quadrada de ordem 2

Exemplo:

`> with(linalg):`

`> B:=matrix([[1,2],[3,5]]);`

ou

`> C:=array(1..2, 1..2, [[2,0], [3,5]]);`

`C:= matrix ([[a,b,c],[d,e,f],[g,h,i]]);` define um matriz quadrada de ordem 3

## Tipos de Matrizes

### Matriz qualquer (ordem m x n)

Uma matriz de ordem 3 x 2

`> R:=array(1..3,1..2,[[1,3],[0,-2],[2,1]]);`

Uma matriz de ordem 4 x 3

`> J:=array(1..4,1..3,[[1,3,0],[0,1,6],[5,-3,2],[0,1,-1]]);`

### Matriz zero

Matriz cujos elementos  $a_{i,j}$  são todos nulos.

Exemplo

`> Z:=matrix([[0,0,0],[0,0,0]]);`

### Matriz identidade

Matriz quadrada que tem todos os elementos da diagonal igual a 1.

`> L:=matrix([[1,0],[0,1]]);`

```
> R:=array(1..3,1..3,[[1,0,0],[0,1,0],[0,0,1]]);

# O exemplo da matriz dada por H := matrix (3, 3, (i,j) -> i * j);
> H:=matrix(3,3,(i,j)->i*j);
```

### 3.3 Determinante de uma matriz

**Definição:** Chama-se **determinante de uma matriz quadrada** a soma algébrica dos produtos que se obtém efetuando todas as permutações dos segundos índices do termo principal, fixado os primeiros índices e, fazendo-se preceder os produtos do sinal + ou -, conforme a permutação dos segundos índices seja de classe par ou ímpar.

A **ordem** de um determinante é igual a ordem da matriz que se deseja calcular o determinante. Por exemplo se a matriz é a matriz quadrada de ordem 3 então a ordem do determinante também é 3.

A **notação** para se calcular um determinante de uma matriz A, em linguagem matemática em geral é  $\det(A)$ . No maple também usa-se o comando  $\det(A)$ .

O determinante de uma matriz de ordem 2

```
> with(linalg):
> A:=matrix([[a[1,1],a[1,2]],[a[2,1],a[2,2]]]);
> det(A)
>
```

Exemplo

```
> B:=matrix([[1,2],[3,5]]);
> det(B);
```

O determinante de uma matriz de ordem 3

```
> C:=matrix(3,3);
> det(C)
>
```

Exemplo

```
> E:=matrix([[1,0,2],[3,5,1],[2,-2,3]]);
> det(E);
```

Exemplo de uma matriz de ordem 4

```
> F:=matrix([[1,1,2,1],[0,5,-1,1],[-1,-2,3,4],[1,0,1,2]]);
```

### Exercício resolvido:

Dada a matriz

```
> G:=matrix([[x,3,2],[5,x,1],[1,3,1]]);
```

Resolva a equação dada por  $\det(G) = 12$

Solução:

```
> det(G)=12;
> solve(x^2-5*x+6);
```

Resposta:  $x = 2$  ou  $x = 3$

### 3.5 Sistema de Equações Lineares

Um problema básico de Álgebra Linear é a resolução de sistemas de equações lineares.

Uma *Equação Linear* é uma equação da forma

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n = b$$

No qual  $x_1, x_2, \dots, x_n$  são as variáveis e  $a_1, a_2, \dots, a_n$  são os respectivos coeficientes das variáveis e  $b$  é o termo independente.

Os valores das variáveis que transformam uma equação linear em identidade, isto é, que satisfazem à equação, constituem sua **solução**.

A um conjunto de equações lineares, dá-se o nome de **sistema de equações lineares**.

Um sistema de equações lineares é da forma

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m$$

Considerando um sistema de  $m$  equações lineares e  $n$  **incógnitas com coeficientes reais**, **podemos escrevê-la** sob forma matricial como  $\mathbf{AX} = \mathbf{B}$ . Resolver o sistema significa discutir a existência de soluções e **obter o conjunto solução quando for possível**.

Algumas informações importantes:

- 1 - Um sistema de equações lineares é dito compatível quando admite solução.
- 2 - Um sistema compatível é determinado quando admite uma única solução.
- 3 - Um sistema compatível é indeterminado quando admite infinitas soluções.
- 4 - Um sistema de equações lineares é dito incompatível quando não admite solução.

Para determinar a solução de um sistema de equações lineares, de um modo geral utiliza-se o comando solve.

Exemplos:

- 1 - Vamos iniciar com um exemplo simples de um sistema de duas equações lineares

$$3x - 4y = 5$$

$$-2x + y = 3$$

Solução:

```
> S_1:={x-4*y=5,2*x-y=3};
```

```
> solve(S_1);
```

Para visualizarmos o ponto encontrado, plote o gráfico das equações em uma janela conveniente

Para plotar o gráfico use o comando implicitplot.

```
>
```

2 - Consideremos agora um sistema de três equações lineares.

$$2x + 4y - 6z = 10$$

$$4x + 2y + 2z = 16$$

$$2x + 8y - 4z = 24$$

```
> S_2:={2*x+4*y-6*z=10,4*x+2*y+2*z=16,2*x+8*y-4*z=24};
```

```
> solve(S_2);
```

É possível a visualização gráfica de S<sub>2</sub>?

Sim é possível, entretanto, observe que cada equação do sistema acima representa um plano no espaço, portanto a visualização gráfica deve ser em 3d (espaço tridimensional).

Para obter o gráfico com o maple, utiliza-se o comando `implicitplot3d` (equações, variação das variáveis x, y, z). No caso do sistema S<sub>2</sub>, proceda do seguinte modo:

```
> with(plots):
```

```
> implicitplot3d(S_2, x = x_1 .. x_2, y = y_1 .. y_2, z = z_1 .. z_2 );
```

3 - Um sistema de quatro equações lineares.

$$x + 3y - z + 2w = 3$$

$$2x + 2y - 2z - w = 6$$

$$3x - 2y + 2z + w = 2$$

```
> S_3:={x+3*y-z+2*w=3,2*x+2*y-2*z-w=6,3*x-2*y+2*z+w=2};
```

```
> solve(S_3);
```

É possível a visualização gráfica de um sistema de 04 equações lineares com quatro variáveis?

Se você acha que é tente obter esse gráfico.

Uma outra maneira de obter a solução de um sistema de equações lineares, é procedendo do seguinte modo:

Considerando o sistema do exemplo 2.

Primeiro vamos obter a matriz dos coeficientes do sistema

```
> with(linalg):
```

```
> A:=genmatrix(S_2,{x,y,z},B);
```

```
>
```

Agora, com o comando `evalm`, vamos obter a matriz B (Que matriz é essa?)

```
> evalm(B);
```

Observe o sistema e descubra quem é essa matriz.

Agora vamos ao próximo passo;

Usar o comando ***linsolve*** ( consulte o ***Help*** para saber sobre esse comando), para obter finalmente a solução.

```
> linsolve(A,B);
```

O resultado é a matriz solução acima que significa  $x = 2$ ,  $y = 3$  e  $z = 1$ .

O processo utilizado para a resolução do sistema foi, encontrar X na equação  $A X = B$ , onde A é a matriz dos coeficientes,

B a matriz (coluna) dos termos independentes e X a matriz solução  $X = [x, y, z]$ .

Um outro modo de encontrar a solução de um sistema é considerando a matriz  $A^{(-1)}$  inversa da matriz A dos coeficientes e resolvendo a equação:  $X = A^{(-1)} B$  onde B é a matriz dos termos independentes.

Obs. Uma matriz coluna é uma matriz da forma:

> **matrix(3,1,[x,y,z]);**

Exemplo: Usar o método da matriz inversa para resolver o sistema dado pelas equações:

$$2x + 3y + 7z = 25$$

$$x + 3y + 2z = -11$$

$$5x + 3y + 4z = -5$$

Solução:

Seja C a matriz dos coeficientes que é dada por:

> **C:=matrix([[2,3,7],[1,3,2],[5,3,4]]);**

Calcular agora a inversa de C

> **inverse(C);**

Considerando a matriz dos termos independentes B, resolver a equação  $X = C B$

> **B:=matrix(3,1,[25,-11,-5]);**

O resultado é matriz

> **X:=evalm(C&\*B);**

A solução é  $x = -18$ ,  $y = -18$  e  $z = 72$

Um exemplo de um sistema compatível e indeterminado

Considere o sistema de equações lineares dado por:

$$2x + 4y + 6z = -6$$

$$3x - 2y - 4z = -38$$

$$x + 2y + 3z = -3$$

Encontre sua solução.

> **S\_4:={2\*x+4\*y+6\*z=-6,3\*x-2\*y-4\*z=-38,x+2\*y+3\*z=-3};**

> **solve(S\_4);**

Observe que as variáveis x e y são dependentes da variável z. A variável z assume infinitos valores, portanto o sistema possui infinitas soluções (compatível e indeterminado).

### 3.6 Atividades 3

1) Com o Maple, obtenha os seguintes vetores:

a)  $m = [-2, 3, 5]$ ;  $n = [1, 0, -1, 3]$ ; Determine a segunda coordenada de m e a terceiro de n;

b) um vetor 3D e escreva-o de duas maneiras: Usando o comando array; usando o comando vector.

2) Escreva uma matriz de ordem 3 x 5 e retorne o elemento da linha 3 e coluna 2;

3) Defina com o maple a matriz quadrada L de ordem 4, de modo que seus elementos sejam da forma  $i * j$

4) Dada a matrix

> **H:=matrix([[2,x-2,1],[1,x+3,4],[3,x+1,5]]);**

Resolva a equação dada por:  $2\det(H) = 56$   $2\det(H) = 56$

5) Dada as matrizes

> **A:=matrix([[1,5,2,-3],[-3,9,-1,10],[0,7,-2,3]]);**

**B:=matrix([[8,10,-5,12],[-7,4,5,1],[6,2,4,11]]);**



Calcule

a) verifique se  $\det(A + B) = \det(A) + \det(B)$ ;

b) Verifique se  $\det(AB) = \det(A)\det(B)$ ;

c) Calcule  $\det(AB^t)$

6) Considere a matriz A dada por:

**> A:=matrix([[4,5,2\*x],[3,9,-x],[5,-2,3\*x]]);**

Resolva a equação dada por:  $3 \det(A) = 145$

7) A condição para que uma matriz A *admita inversa* é que  $\det(A) \neq 0$ .

Verifique se as matrizes abaixo admitem inversa, calculando seus determinantes. Caso elas admitam inversas, determine-as.

**> L:=matrix([[2,1,3],[4,2,2],[2,5,3]]); S:=matrix([[1,3,1,0],[-2,1,0,0],[0,0,3,0],[2,0,1,0]]);**

8) Use a condição do exercício 6, para resolver o seguinte exercício:

Determine o valor de  $x$  para que a matriz dada abaixo não tenha inversa

**> C:=matrix([[2,3,1],[6,x,2],[1,2\*x,3]]);**

9) Defina com o maple a matriz quadrada L de ordem 4, de modo que seus elementos sejam da forma  $i / j$ .

10) Dê exemplo de uma matriz linha e de uma matriz coluna;

11) Encontre a inversa da matriz A (chame-a de C) do exercício 4 e mostre que o produto da matriz  $A C = I$ .

12) dada as matrizes

$E := \text{array}([ [2, 3, 1], [-1, 3, -2], [2, 0, 4] ])$ ;

$F := \text{array}([ [0, 1, 2], [3, 2, 5], [1, 2, 3] ])$  ;

b) Calcule  $(E + F)^2$

d) Verifique se  $E F = F E$  ;

e) Mostre que  $(E - F)^t = E^t - F^t$  ;

f) Verifique se  $(E F)^t = E^t F^t$  ;

g) A inversa da matriz E e a inversa da matriz F;

11) Resolva o sistema de equações lineares dado por:

$$\begin{aligned} x - y + 2z &= -3 \\ 5x + 3y - 4z &= 8 \\ 7x + 2y + z &= 10 \end{aligned}$$

a) Usando o comando solve;

b) resolvendo a equação da forma  $AX = B$ ;

c) Pelo método da matriz inversa.

12) Resolva e classifique os sistemas dados abaixo:

a)  $2x - 5y - z = -8$

b)  $x + 4y + 6z = 11$

$x - 2y - 4z = -11$

$2x + 3y + 4z = 9$

$-5x + y + z = -9$

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

## Parte 4: Introdução a Programação com Maple

O Maple possui linguagem de programação de alto nível, disponível *on-line* e compatível com os recursos de sua própria estrutura. Assim, é possível construir procedimentos nos quais o usuário tenha um interesse especial, e utilizar nessa construção todas as estruturas e funções do Maple, ou mesmo estendê-las ou modificá-las, de acordo com suas necessidades particulares. Procedimentos e estruturas modificadas podem ser inicializadas em um único pacote, e um conjunto de procedimentos pode ser inicializado em uma biblioteca.

Um dos fatos que tornam o Maple um programa bastante elogiado e reconhecido no mundo inteiro é a facilidade que ele dá ao programador que objetiva lidar com computação algébrica. A grande quantidade de funções disponíveis para os mais diversos fins na área de Matemática, Estatística, etc, poupam muitas linhas de código (e consequentemente, muito tempo e erro) quando comparamos o Maple a outras linguagens (C, C++, Pascal, etc).

Antes de iniciarmos a introdução a programação com o maple, vamos conhecer um pouco sobre a estrutura de dados do maple.

### 4.1 Manipulando Estruturas de Dados Básicas

#### **nops, op, [], subsop, NULL**

Expressões ou estruturas de dados, tais como expressões algébricas, funções, sequências, listas e conjuntos, geralmente envolvem operadores, como  $+$   $-$   $*$   $/$   $^$   $.$   $,$

**nops** dá o número de operandos (ou elementos):

```
> nops(a+b+c+d);
> nops(f(a,b));
> nops([a,b,c]);
```

**op** fornece os operandos (ou elementos) de uma sequência:

```
> op(a+b+c+d);
> op(f(a,b));
> op([a,b,c]);
```

**op** aceita um índice numérico ou domínio de valores como primeiro argumento opcional e retorna uma subsequência de operandos:

```
> op(2, [a,b,c]);
> op(2..3, [a,b,c]);
> op(2,[a,{3,4},c]);
```

O Operando zero é o operador ou tipo de estrutura:

```
> op(0, a+b);
> op(0, [a,b,c]);
> op(0,[a,{3,4},c]);
```

[] é como **op** , mas preserva a estrutura de dados:  
 > **[a,b,c][2..3];**

lembrando que  
 > **op(2..3, [a,b,c]);**  
 exceto quando não há índices:  
 > **[a,b,c][];**

Somente [] pode ser usado diretamente com uma sequência:  
 > **(a,b,c)[2];**  
 > **op(2,(a,b,c));**

**subsop** combina **subs** e **op** :  
 > **subsop(2=B, [a,b,c]);**

**NULL** representa a sequência vazia, que é útil para inicializar sequências e deletar elementos:  
 > **S := NULL;**  
 > **S := S,a,b;**  
 > **subsop(2=NULL, [a,b,c]);**

Note que **NULL** é utilizado somente para entrada e nunca aparece na saída.

Somas, produtos, sequências, funções, funções, listas e conjuntos são estruturas de dados muito similares.

Listas, funções e conjuntos podem ser manipuladas via suas sequências, por exemplo,  
**op(lista) -> sequência, [sequência] -> lista**

para se ter eficiência:

- escreva explicitamente e comente
- use **add, mul, seq**
- use um loop **do** [com todo o código relacionado no mesmo grupo de execução] . Por exemplo, suponha que  $x_{n+1} = f(x_n)$ ,  $x_1 = a$  e queremos computar  $\sum_{i=1}^n x_i$ ,  $\prod_{i=1}^n x_i$ , e a sequência  $x_i, i = 1 .. n$  for  $n = 5$  :

```
> restart:
> # Começa o gpo de execucao
# Inicialização:
S := 0:   # soma
P := 1:   # produto
Q := NULL: # sequência
```

```

x := a: # primeiro elemento
n := 5: # número de elementos
# Loop principal:
to n do
  S := S + x; # soma
  P := P * x; # produto
  Q := Q , x; # sequência
  x := f(x) # próximo elemento
end do:
# Resultados:
'S' = S;
'P' = P;
'Q' = Q;
# Fim do grupo de execução

```

#### **Juntando e estendendo listas**

```

> L1 := [a,b,c]: L2 := [d,e,f]:
> [op(L1),op(L2)]; # junta
> [op(L1),d]; # estende

```

#### **Inserindo e deletando elementos de lista**

```

> L := [a,b,c,d];
> [op(1..2,L), m, op(3..4,L)]; # insere depois do segundo elemento
> [op(1..2,L), op(4,L)]; # deleta o terceiro elemento
> subsop(3=NULL, L); # deleta o terceiro elemento

```

#### **Somando e multiplicando elementos de listas ou sequências; operadores como funções**

A maioria dos operadores pode ser utilizada como função (mapping or function), se protegidos por aspas agudas (backward quotes):

```

> `+(a,b,c);
> L := [a,b,c]:
> `*(op(L)); # op dá uma sequência de elementos
> `*(a,b,c);

```

Note que

```

> mul(el, el=L);
> add(el, el=L);
> seq(e1,e1=L);

```

#### **Comando map**

```

> restart:

```

Aplica uma função a cada operando de uma expressão:

```

> map(f, [a,b,c]);
> map(f, {a,b,c});
> map(x -> x^2, x+y+z);
> map(x -> x^2, {a,b,c});

```

```
> f := (x,y) -> cos(x*y);
```

## 4.2 Simulação de Gambling

João e Maria estão jogando uma moeda.

Toda vez que sai Coroa, João paga à Maria um centavo; se sai Cara, Maria paga ao João também um centavo. O jogo só acaba quando um dos jogadores perde todo o seu dinheiro.

João começa jogando com 4 centavos e Maria com 5 centavos.

Simule o jogo e faça uma estimativa do valor significativo do jogo.

- **Configurando a Moeda**

*Cara: 1      Coroa: -1*

```
> ZeroOne := rand(0..1);
> die := proc( ) 2*ZeroOne( ) - 1 end;
> die( );
```

*No começo do jogo :*

```
> João := 4; Maria := 5;
```

*Uma vitória:*

```
> p := die( );
> João := João + p;
> Maria := Maria - p;
```

## Iteração

Nós repetimos os passos anteriores até o jogo acabar

*Começo:*

```
> João := 4; Maria := 5;
> counter := 0;
>
```

*O loop:*

```
> while (João <> 0 and Maria <> 0) do
>   p := die( );
>   João := João + p;
>   Maria := Maria - p;
>   counter := counter + 1;
> od;
>
```

O jogo como um Procedimento:

```
> game :=
> proc(João, Maria)
```

```

> local j, s, p, counter;
> counter := 0;
> j := João;
> s := Maria;
> while (j <> 0 and s <> 0) do
>   p := die( );
>   j := j + p;
>   s := s - p;
>   counter := counter + 1;
> od;
> end:
>

```

O que acabamos de ver foi um exemplo de um simples programa usando o maple, onde aparecem alguns comandos usuais de programação. Agora, vamos ver mais especificamente o uso desses comandos. Começando como o mais simples deles que é o “proc”.

### 4.3 Procedimentos:

É um programa em Maple na sua forma mais simples.

Exemplo 1

```

f:=proc(< >)
  <comandos>
end

```

```

> ?proc;
> metade:=proc(x)
>   evalf(x/2);
> end;
metade(3);

```

```

> divi:=proc(p,q)
>   p/q;
> end;
divi(2,3);
> divi(2,3.0);
> divi(2-3*I,2+I);
> divi(2-3*I,2.+I);

```

### 4.4 Variáveis locais e globais:

Em linguagem de programação, algumas vezes demos declarar as variáveis como Locais e Globais:

Quais variáveis são globais dentro dos procedimentos?

Aquelas que são declaradas globais são usadas no corpo do procedimento, mas não são um valor associado. Todas as outras são locais, se necessário, é feita local pelo próprio Maple.

Um exemplo: a implementação de uma chave de on/off.

```
> on := proc( )
>   global switch;
>   switch := true
> end:
> state := proc( )
>   global switch;
>   switch
> end:
> on():
> state();
>
```

O que aconteceria se você esquecesse da declaração na definição do procedimento? Veja abaixo.

```
> off := proc( )
>   switch := false
> end;
> on(): off():
> state();
>
```

#### 4.5 Comandos de Repetição: for-do-od

O looping gerado pelo for, na sua versão mais básica, se escreve:

```
for i from <i0> to <ifim> do
  <comandos>
od
```

Exemplo 1.

Soma dos n primeiros números:

```
> soma:=proc(n)
  local total,i;
>   total:=0;
>   for i from 1 to n do
>     total:=total+i
>   od;
>   total;
> end:
> soma(10^6);time(soma(10^6));
```

```
> add(i,i=1..10^6);time(add(i,i=1..10^6));
```

Exemplo 2.

Sequência de Fibonacci, com condições iniciais  $f_0$  e  $f_1$ :  $f(n) = f(n-2) + f(n-1)$

```
> fibon:=proc(n,f0,f1)
  f(0):=f0;
  f(1):=f1;
  > for i from 2 to n do
    > f(i):=f(i-1)+f(i-2)
  > od;
> end:
>
> fibon(5,1,0);fibon(400,1,0);
```

**Obs. 1** Existem três formas particulares do comando [for](#) que são [seq](#), [add](#) e [mul](#). A sintaxe desses comandos são iguais, de forma que vamos apenas descrever o comando [add](#) que executa um somatório. A sua sintaxe é uma das seguintes formas

```
add(f, i = a .. b)
add(f, i = L)
```

onde  $f$  é uma expressão que geralmente depende do índice  $i$ . Na primeira forma, o índice  $i$  assume valores inteiros de  $a$  até  $b$ , onde  $a$  e  $b$  tem que ser numéricos. Na segunda forma o índice assume valores da expressão ou estrutura de dados  $L$ . Por exemplo

```
> restart:
> add( a[i]*x^i, i=0..5 );
> add( sin(i), i=[theta,omega,tau]);
> seq( i^2, i=1..5 );
> mul(x*i,i=1..5);
> L := [seq(i, i=1..6)];
> add(i,i=L);
> mul( x-i, i=L );
```

**Obs.2** Para incremento: a partir de um valor inicial de variável, incrementa (decrementa) a mesma com o número de unidades especificado em incremento (se o incremento não for especificado, será considerado como uma unidade) e executa o bloco de comandos até que a variável atinja um valor limite ou que a condição em while se torne falsa.

Exemplo 3

O exemplo que se segue calcula o quociente de uma divisão com resto  $o$ . Consideraremos inicialmente a variável  $i$  sem especificar o incremento

```
> for i from 1 to 17 do
  > if i mod 3=0 then
  > print(i);
  > fi;
> od;
```



Agora o mesmo exemplo com a variável incrementada de 2 unidades

```
> for i from 1 by 2 to 17 do
> if i mod 3=0 then
> print(i);
> fi;
> od;
```

O exemplo a seguir considera um decremento à variável

Exemplo 4:

```
> j:=0:
> for i from 35 by -5 while i > 0 do
> if i+j mod 3 = 1 then
> print('i = `i.` e j = `j');
> j:=j+1:
> fi;
> od;
```

#### 4.6 O comando while com procedimento

Exemplo 1

Definindo a divisão euclidiana:

```
> div := proc(a::posint, b::posint)
> local i, c, d, e;
> i:=0: c:=a: d:=b: e:=a;
> while c>=d do e:=c-d: c:=e: i:=i+1: od:
> print('o quociente de `a.` por `b.` é `i.` e o resto é `e');
> end;
```

Usando o div

```
> div(9,5);
> div(17,3);
> div(73,12);
> div(5.75,3.2);
> div(-3,7);
> div(2,7);
```

Exercício especial

A função rand( ) do Maple gera números aleatórios (teste e verifique no help do Maple como ela o faz).

Escreva um programa que gera números aleatoriamente com rand( ), para quando encontrar um número menor que k devolver a quantidade de números gerados com rand( ) até obter um número menor que k .

```
> randk:=proc(k::integer)
> local n,conta;
> n:=k+1;
> conta:=0;
> while n >= k do
```

```

    n:=rand();
    conta:=conta+1;
od;
> print('O número obtido foi n =', n, ' e foram necessários gerar', conta, '
números');
> end:
randk(10000000000);

```

#### 4.7 Condicional: if - then [else] - fi.

É o clássico:

se-então-(senão-)fim

e tem várias versões. O Feijão com arroz é:

if < condições > then <comandos> || else <comandos> || fi

```

> ? if
> divi:=proc(p,q)
if q=0 then
    ERROR('Denominador nulo');
fi;
> p/q;
> end;

> divi(2,0);

```

**Função chapéu:**

```

> chap:=proc(x)
> if x<=0 then 0
else
    if 0<x and x<=2 then
        x
    else
        if x>2 and x<=4 then
            4 - x
        else
            0
        fi;
    fi;
fi;
> end:
> chap(1);
Alternativamente:

```

```

> chap:=proc(x)
> if x<=0 then 0
    elif 0<x and x<=2 then x
    elif x>2 and x<=4 then 4 - x

```

```

    else 0
  fi;
> end;

```

**Exercício 3** - Construa um gráfico para a função chapéu

```
> plot(chap,-10..10);
```

**Exercício 4** - Programe e teste a função módulo (valor absoluto) -  $ABS(x) := |x|$

```

> ABS:=proc(x)
>   if x >= 0 then x
>   else -x
>   fi;
> end;
plot(ABS,-5..5);

```

#### 4.8 While - do - od - (enquanto - faça)

while <condições> do <comandos> od

```

> fator2:=proc(n)
  local contafator,m;
  m:=n;
  contafator:=0;
>   while is(m/2,integer) do
    m:=m/2;
    contafator:=contafator+1;
  od;
end:
> fator2(36);

```

Exercício n. 6: Faça o exercício n.1 usando o comando while-do, em vez de for-do.

**Exercício 7:** Certifique-se que a função rand( ) do Maple gera números aleatórios (teste e verifique no help do Maple como ela o faz).

Escreva um programa que gera números aleatoriamente com rand( ), para quando encontra um número menor que k e devolve a quantidade de números gerados com rand( ) até obter um número menor que k .

```

> randk:=proc(k::integer)
  local n,conta;
>   n:=k+1;
  conta:=0;
  while n >= k do
    n:=rand();
    conta:=conta+1;
  od;
>   print('O número obtido foi n =', n, ' e foram necessários gerar', conta, ' números');

```

```
> end:
randk(10000000000);
```

Exemplo: Um Programa que usa o algoritmo de Euclides para calcular o máximo divisor comum (MDC) entre dois números naturais  $m$  e  $n$ . (O método de Euclides, basicamente, divide  $m$  por  $n$  (se  $m > n$ ), encontrando um resto  $r_1$ . Se  $r_1 = 0$ ,  $n$  é o m.d.c. Senão, divide  $n$  por  $r_1$  encontrando  $r_2$ . Se  $r_2 = 0$ ,  $r_1$  é o mdc. Senão, divide  $r_1$  por  $r_2$ , dando  $r_3$ . Se  $r_3 = 0$ , então  $r_2$  é o mdc e assim por diante, até dar divisão exata. Demonstre que o método acima sempre funciona, escreva o algoritmo correspondente, bem como um programa para Maple que calcula o mdc. Calcule o mdc entre 44559431 e 4619832

```
> restart;
mdc:=proc(m,n)
  local resto,div,aux;
> resto:=irem(m,n);
  div:=n;
  while resto>0 do
    aux:=resto;
    resto:=irem(div,resto);
    div:=aux;
  od;
> div;
end:
mdc(12,8);
```

#### 4.9 Atividades 4

1. Atribua a lista  $[a,b,c]$  à variável **L1** e  $[d,e,f]$  à variável **L2**. Então, usando somente as variáveis **L1** and **L2**, atribua à variável **L** a lista  $[a,b,c,d,e,f]$ , ou seja, junte as listas **L1** e **L2**. Sugestão: Use o comando **op**.
2. Delete de **L** o segundo e o terceiro elemento, inclusive, com e sem **subsop**. [Veja help para mais detalhes de **subsop**.]
3. Utilizando somente a variável **L**, adicione seus elemento eficientemente usando **+** como uma função e sem usar **+**.
4. Use o comando **map** para aplicar
  - a)  $2x$  a  $a, b$  e  $c$ ; b)  $x^3$  a  $x, y$  e  $z$ ;
5. Mapei a função  $g = \exp(x)/\sin(y)$ ;
- 6 - Use o comando **for** pra gerar os elementos pares de 2 até 20.
- 7 – Faça um procedimento que calcule a o inverso de um número.

8 – Repita o exercício 2 para calcular a raiz cúbica de um número.

9 – Faça um procedimento que calcule a diferença entre dois números dividido por 2,  $c = (a - b) / 2$ . Declare  $c$  como variável "local" e chame o procedimento de dif2. Calcule  $c$  para:  $a = 12$  e  $b = 18$ ;  $a = 56$  e  $b = 17$ ;  $a = -32$  e  $b = 45$ ;  $a = 259$  e  $b = -123$

10 – Escreva e teste um programa para a soma dos  $n$  primeiros termos de uma P.G. de termo inicial  $a_0$  e razão  $r$ .

11 – Como se define uma iteração usando o comando **for do od** que executa a mesma tarefa que o comando **seq 1:=seq(termo(a+i\*n), i=0..(b-a)/n)**; onde  $a$ ,  $b$  e  $n$  são números.

12 – Usando o condicional **if - fi** construa a seguinte função:

Se  $-3 \leq x$  e  $x \leq 0 \Rightarrow f(x) = x^2 + 1$ ,  $x > 0$  e  $x \leq 3 \Rightarrow f(x) = x + 1$

## **Parte 6 – Referências Bibliográficas**

- [1] Renato Portugal, Introdução ao Maple, Coordenação de Ciência da Computação Laboratório Nacional de Computação Científica - Petrópolis-RJ - Copyright, Renato Portugal, 2002, Copyright, Renato Portugal, 2002.
- [2] Marcia M. C. Cruz, Notas de aulas de Laboratório de Apoio Computacional, Departamento de Matemática – CCET – UFRN.
- [3] Heal, K. M., Hansen, M., Rickard, K., Maple V Learning Guide, Springer-Verlag, New York, 1997.
- [4] Monagan, M., Geddes, K., Heal, K., Labahn, G., Vorkoetter, S., Maple V Programming Guide, Springer-Verlag, New York, 1997.
- [5] Heck, A., Introduction to Maple, Second edition, Springer-Verlag, New York, 1996.
- [6] Wright, F., Computing with Maple, Chapman & Hall/CRC, 2001.
- [7] Koer, M., Maple: An Introduction and Reference, Addison-Wesley, 1997.
- [8] Introdução à Programação em Maple, III Semana da Física, Fernando Deeke Sasse, UDESC-Joinville, 25-29/8/2003