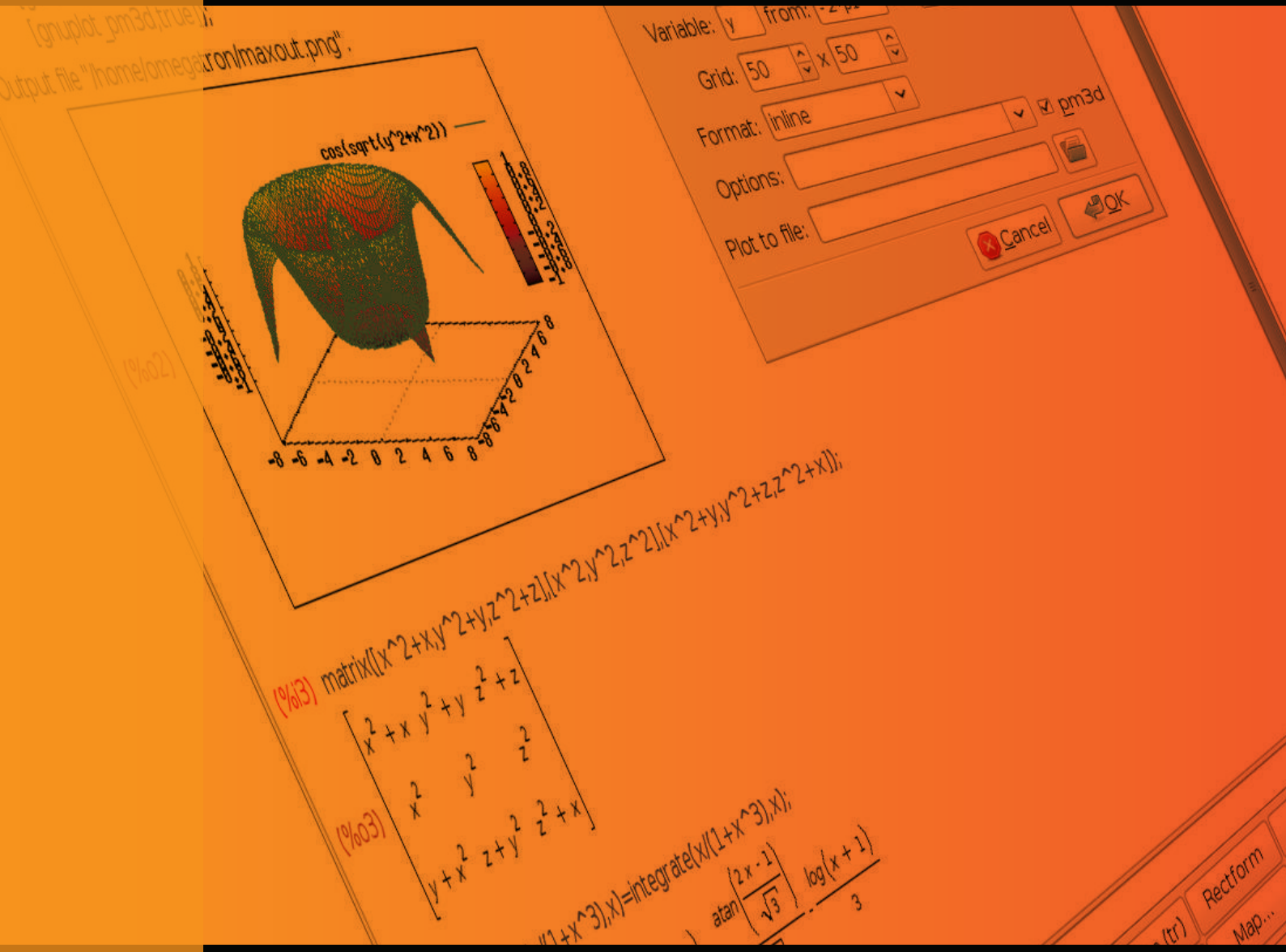
O software **MAXIMA** e aplicações



Cristina Lúcia Dias Vaz

editAedi
Assessoria de Educação a Distância • UFPA

E-book

Cristina Lúcia Dias Vaz

O software Maxima e aplicações

1ª edição

Belém - Pará

editAedi
Assessoria de Educação a Distância • UFPA

2016

Comitê Editorial

Editora da Assessoria da Educação a distância da UFPA -EditAedi

2016

e-ISBN: 978-85-65054-33-1

Diagramação: Cristina Vaz

Revisor: Edilson dos Passos Neri Junior

Editora: EditAedi

Capa: Gioordanna De Gregoriis

Catalogação Bibliográfica

Vaz, Cristina L. D.
O software *Maxima* e aplicações - Belém, PA : EditAedi, 2016, 176 p.

e-ISBN: 978-85-65054-33-1

1. Aplicativo *Maxima* 2. Computação Algébrica
3. Cálculo 4. Fractais
I. Vaz, Cristina L. D. II. Título.

Dedicatória

Dedico este livro ao meu filho Arnaldo, que me ensinou amar incondicionalmente.

Agradecimentos

Gostaria de agradecer a todos alunos que aceitaram, direta ou indiretamente, o desafio de trilhar os caminhos "do software livre", em particular os que trabalharam com o aplicativo `Maxima`.

Um agradecimento carinhoso aos amigos e colegas que contribuíram direta ou indiretamente para a realização deste livro.

Sumário

PREFÁCIO

A computação simbólica ou álgebra computacional é um ramo da Ciência da Computação e da Matemática que trata de cálculos simbólicos que envolvem fatoração de polinômios, resolução de equações algébricas e equações diferenciais, operações e cálculos com matrizes, grupos, tensores, entre outros.

Os cálculos realizados no tratamento simbólico são exatos (isto é, têm precisão infinita) em contraste ao correspondente tratamento numérico. Embora, em determinados problemas, pode-se combinar os dois métodos para gerar fórmulas simbólicas que posteriormente serão tomadas com *input* em programas numéricos, como é, por exemplo, o caso da resolução de equações algébricas.

Nas últimas décadas muitos sistemas de computação simbólica foram desenvolvidos. Os aplicativos mais conhecidos são o Axiom, Derive, Macsyma, Maple, Matlab, Mathematica, Reduce e *Maxima*.

Estes aplicativos são de grande utilidade no processo de ensino e na resolução de problemas. São ferramentas que podem ser usadas para tornar o ensino da Matemática mais experimental, além da abordagem de problemas mais complicados com visualização gráfica, o que, consequentemente, permite ao estudante *aprender experimentando*.

Este livro é um guia especial do aplicativo *Maxima*. Nossa escolha essencialmente considerou o fato do *Maxima* ser um software livre, de acesso gratuito na Internet, análogo ao *Mathematica* e ao *Maple* e compatível com o UNIX, LINUX, MAC e Windows. Por ser um software livre possui as seguintes vantagens: acesso ao código fonte, incentivo ao desenvolvimento de um espírito de comunidade, maior segurança, incentivo ao trabalho colaborativo, criação de patrimônio cultural da humanidade, entre muitas outras.

Para mais detalhes consulte o sítio **http://maxima.sourceforge.net**.

O objetivo principal deste livro é apresentar, e disponibilzar para um público mais amplo, as princinpais ferramentas do aplicativo que a autora usou, e ainda usa, em sua prática docente, na Universidade Federal do Pará. Portanto, trata-se de uma coletânea de tópicos de Matemática que serão apresentados de forma elemen-

tar dando-se ênfase ao uso do aplicativo. A maioria destes tópicos foram tratados pela autora e seus alunos em disciplinas da graduação e orientação de trabalhos de conclusão de curso.

A autora foi inspirada por vários livros, textos e manuais encontrados livremente na internet, os quais, em sua maioria, estão listados na bibliografia do livro.

O livro foi organizado do seguinte modo: no Capítulo 1 apresentaremos os principais comandos básicos do aplicativo *Maxima*: comandos de manipulação algébricas, biblioteca e geração de funções, geração de listas, montagem de gráficos e os principais comandos de programação. No Capítulo 2 usaremos o aplicativo *Maxima* para resolver problemas do Cálculo Diferencial e Integral e no Capítulo 3 usaremos o software na geração e programação de Fractais. No final do texto, anexamos a bibliografica consultada.

Belém, 15 de março de 2016.

Cristina Vaz

1 Principais comandos

1.1 Informações preliminares

Um pouco de história

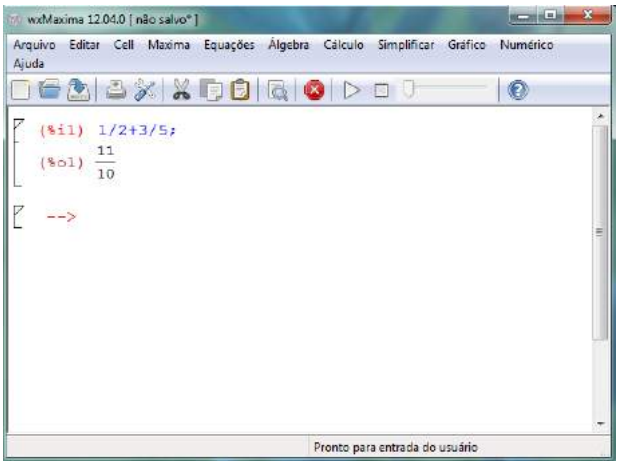
Maxima é um dos sistemas CAS mais antigos. Foi criado pelo grupo MAC no MIT, na década de 1960, e inicialmente chamava-se *Macsyma* (*project MAC's SYmbolic MAnipulator*). Macsyma foi desenvolvido inicialmente para os computadores de grande escala DEC-PDP-10 que eram usados em várias instituições acadêmicas.

Na década de 1980 foi desenvolvido para várias plataformas, e uma das novas versões foi chamada de *Maxima*. Em 1982, o MIT decidiu comercializar Macsyma e, paralelamente, o professor William Schelter da Universidade de Texas continuou a desenvolver o *Maxima*. Na segunda metade da década de 1980 apareceram outros sistemas CAS proprietários, por exemplo, *Maple* e *Mathematica*. Em 1999, a versão proprietária do *Macsyma* foi vendida e retirada do mercado, incapaz de concorrer com os outros sistemas CAS. Em 1998, o professor Schelter obteve autorização do DOE (*Department of Energy*), que tinha os direitos de autor sobre a versão original do *Macsyma*, para distribuir o código fonte do *Maxima*.

Após a morte do professor Schelter, em 2001, formou-se um grupo de voluntários que continuaram a desenvolver e distribuir o *Maxima* como software livre.

Instruções iniciais

O `Maxima` trabalha de modo interativo, isto é, ele exibe uma área de trabalho (chamada **wxMaxima**), na qual você deverá digitar os comandos e visualizar as respostas, como mostra a seguinte figura:



Algumas instruções preliminares são necessárias antes de você começar a usar o aplicativo.

- Cada linha de entrada (input) é designada por (% in), com n o número da linha de entrada e cada linha de saída (output) é designada por (% on), com n o número da linha de saída;
- Cada linha de comando é finalizada com **terminador** *ponto e vírgula* (;)
- Sempre que terminar de digitar uma linha de comando acione a tecla shift+ < enter > ou < enter > para que o *Maxima* efetue as operações;
- Todos os comandos do *Maxima* são escritos com letras minúsculas. O aplicativo faz diferença entre maiúsculas e minúsculas. Por exemplo, **sin(x)** é diferente de **Sin(x)** ou **sin(X)**;
- Os comandos do *Maxima* podem ser usados "clicando-se" diretamente nos respectivos ícones, no menu da área de trabalho.

1.2 Operações básicas

1.2.1 Operadores Aritméticos

Podemos usar o *Maxima* como uma calculadora efetuando operações numéricas com os seguintes operadores aritméticos:

^	-	potência
–	-	subtração
/	-	divisão
*	-	multiplicação
+	-	adição

Exemplo 1.1 Efetue as seguintes operações:

a) $5 \times 8 \times 3$; b) $3{,}2 + 6{,}32$; c) $(8+5)^3 - 6 \times (1+4)$; d) $\dfrac{2{,}65}{4{,}5}$.

Solução:

Maxima

```
(%i1)5*8*3;

(%o1)  120
(%i2)3.2+6.32;

(%o2)  9.52
(%i3)(8+5)^3-6*(1+4);

(%o3)  2167
(%i4)2.65/4.5;

(%o4)  0.58888888888889
```

1.2.2 Resultados aproximados e exatos

Como o *Maxima* é um aplicativo simbólico, frequentemente vai gerar resultados **exatos**. Para obtermos resultados aproximados usamos um dos seguintes comandos:

float(M):	-	resultado decimal do número M.
M, **numer**:	-	resultado decimal do número M.

Exemplo 1.2 Calcule o valor exato e o valor aproximado de $\frac{1}{5}+\frac{5}{7}$.

Solução:

Maxima

```
(%i5)1/5+5/7;
```

(%o5) $\frac{32}{35}$

```
(%i6)float(1/5+5/7);
```

(%o6) 0.91428571428571

```
(%i7)1/5+5/7, numer;
```

(%o7) 0.91428571428571

O aplicativo *Maxima* pode efetuar cálculos com precisão muito alta. Por exemplo, calcular o valor de π com precisão de 40 dígitos:

Maxima

```
(%i8)fpprec:40;
```

(%o8) 40

```
(%i9)bfloat(%pi);
```

(%o9) 3.141592653589793238462643383279502884197*b*0

O modificador **bfloat** foi usado para obter uma representação no formato "big float". A constante **fpprec** controla o número de algarismos significativos usados no formato "big float". O valor de 40 para a variável **fpprec** só tem efeito dentro do bloco "bfloat" onde foi usado; fora do bloco, **fpprec** continua com o seu valor habitual de 16. A letra **b** no fim do resultado representa a ordem de grandeza do número; neste caso b0 representa um fator de $10^0 = 1$.

Constantes e funções

O *Maxima* possui uma biblioteca de constantes e funções que podem ser usadas diretamente.

Constantes usuais

%pi	-	$\pi = 3.1415926...$
%e	-	$e = 2.7182818...$
%i	-	$i = \sqrt{-1}$

Destacamos os símbolos **inf** e **minf** que significam os símbolos matemáticos $+\infty$ e $-\infty$, respectivamente.

Funções usuais

abs(x), sqrt(x)	-	$\lvert x\rvert$, $\sqrt{x}$
exp(x), log(x)	-	e^x, ln(x)
sin(x), cos(x), tan(x)	-	sen(x), cos(x), tg(x)
sec(x), csc(x), cot(x)	-	sec(x), cossec(x), cotg(x).

Destacamos os comandos **n!** e **random(x)** que geram o fatorial de n e um número aleatório entre $[0, x-1]$, respectivamente.

Exemplo 1.3 a) Calcule o valor aproximado de $\log(11)$ e e^{22};
b) Calcule o valor exato de $\text{sen}\left(\frac{\pi}{5}\right)$;
c) Calcule o fatorial de 30.

Maxima

```
(%i10) float(log(11));
(%o10)  2.397895272798371
(%i11) exp(22), numer;
(%o11)  3.5849128461315918*10^9
(%i12) sin(%pi/5);
(%o12)  sin(%pi/5)
(%i13) 30!;
(%o13)  265252859812191058636308480000000
```

Operadores lógicos e relacionais

Podemos usar o `Maxima` para efetuar operações lógicas ou relacionais com os seguintes comandos:

is(expr)	-	verifica se a expressão **expr** é falsa ou verdadeira
assume(expr)	-	assume que a expressão **expr** é verdadeira
forget(expr)	-	apaga o valor lógico da expressão **expr**
and	-	operador lógico **e**
or	-	operador lógico **ou**
=, **notequal**	-	igual, diferente
>, >=	-	maior , maior ou igual
<, <=	-	menor, menor ou igual.

Não podemos considerar várias expressões condicionais simultaneamente, como por exemplo, $6 < 7 < 8$. As desigualdades somente se aplicam aos pares de expressões. Para a resposta verdadeira, o Maxima usa **true** e para falsa, **false**. Quando acontece uma indefinição, usará a palavra **unknown**.

Exemplo 1.4 Efetuando operações lógicas

Maxima

```
(%i14)is(9<3);

(%o14)  false
(%i15)is(6<50 or 6 <7);

(%o15)  true
(%i16)is(x^2 >0);

(%o16)  unknown
(%i17)is(x^2 >=0);

(%o17)  true
```

1.3 Definindo funções

Além das funções da biblioteca do Maxima, podemos também definir (criar) novas funções usando o comando atribuidor **:=**.

$f(x_1,\ldots,x_n)$:=expr(x_1,x_2,..., x_n)	-	define uma função $f:\mathbb{R}^n \to \mathbb{R}$
$f(x_1,\ldots,x_n):=[f_1(\vec{x}),\ldots,f_m(\vec{x})]$	-	define uma função $f:\mathbb{R}^n \to \mathbb{R}^m$
kill(f)	-	remove a definição de f

Os nomes **f** que usamos para definir as funções são símbolos que não devem ser confundidos pelo programa com nomes já existentes. Por isso, é bom evitar o mesmo nome para diferentes funções. Ao terminar de usar o símbolo **f**, apague sua definição usando o comando **kill(f)**.

Exemplo 1.5 Defina as funções dadas abaixo e efetue os cálculos indicados.

a) $f(x) = \sqrt{\dfrac{x^2+1}{3x^4+1}}$; o valor de $f(2.3)$; o valor exato de $f(1)$;

b) $g(x,y,z) = y\cos(x) + z^2$; o valor exato e aproximado de $g(5,3,1)$;

c) $h(t) = (t, t^2)$; o valor de $h(2)$.

Maxima

```
(%i18) f(x):=sqrt((x^2+1)/(3*x^4+1));
```

$$(\%o18)\quad f(x) := \sqrt{\frac{x^2+1}{3x^4+1}}$$

```
(%i19) f(2.3);
```

$$(\%o19)\quad 0.2721057705671$$

```
(%i20) f(1);
```

$$(\%o20)\quad \frac{1}{\sqrt{2}}$$

```
(%i21) g(x,y,z):=y*cos(x)+z^2;
```

$$(\%o21)\quad g(x,y,z) := y*\cos(x) + z^2$$

```
(%i22) g(5,3,1);
```

$$(\%o22)\quad 3*\cos(5) + 1$$

```
(%i23) h(t):=[t,t^2];
```

$$(\%o23)\quad h(t) := [t, t^2]$$

```
(%i24) h(2)
```

$$(\%o24)\quad [2,4]$$

Funções definidas por sentenças

Podemos definir funções dadas por sentenças do tipo

$$f(x) = \begin{cases} f_1(x) & \text{se} \quad x_1 < x \leq x_2 \\ f_2(x) & \text{se} \quad x_2 < x \leq x_3 \\ \cdots & \\ f_m(x) & \text{se} \quad x_{n-1} < x \leq x_n. \end{cases}$$

Para isto, vamos usar a estrutura condicional **if cond then expra else exprb** que avaliará a expressão **expra** se a condição **cond** for verdadeira; se for falsa avaliará a expressão **exprb**.

Exemplo 1.6 Defina a função

$$f(x) = \begin{cases} x^2+1 & \text{se} \quad x \leq 0 \\ \cos(x) & \text{se} \quad 0 < x \leq \pi \\ 2x+3 & \text{se} \quad x \geq \pi \end{cases}$$

e calcule $f(0), f(-1), f(\pi)$.

Soluções:

Maxima

```
(%i25)f(x):=if x<=0 then x^2+1 else  if x < %pi then cos(x) else 2*x+3;
(%o25)  f(x) := ifx <= 0 then x^2 + 1 else if x < π then cos(x) else 2x + 3;
(%i26)[f(0),f(2),f(%pi)];
(%o26)  [1,cos(2),2π + 3]
```

1.4 Manipulando expressões algébricas

Podemos fatorar, expandir e simplificar uma expressão algébrica com os seguintes comandos:

expand(expr)	-	efetua produtos e potências da expressão **expr**
partfrac(expr,var)	-	simplifica as frações na variável **var** de **expr**
factor(expr)	-	fatora a expressão **expr**
ratsimp(expr)	-	simplifica a expressão **expr**
radcan(expr)	-	simplifica expressões **expr** com radicais.

Exemplo 1.7 Simplificando expressões.

Maxima

```
(%i27)p:(x+5)*(x-3);
```

(%o27) $(x-3)*(x+5)$

```
(%i28)q:(x+1)^3;
```

(%o28) $(x+1)^3$

```
(%i29)ratsimp(p);
```

(%o29) $x^2+2x-15$

```
(%i30)expand(q);
```

(%o30) x^3+3x^2+3x+1

```
(%i31)factor(%);
```

(%o31) $(x+1)^3$

```
(%i32)expand(p/q);
```

(%o32) $\frac{x^2}{x^2+2x-15}+\frac{2x}{x^2+2x-15}-\frac{15}{x^2+2x-15}$

Podemos também simplificar expressões trigonométricas com os seguintes comandos:

trigexpand(expr)	-	efetua produtos e potências sem simplicar **expr**
trigsimp(expr)	-	usa a relação $cos^2(x)+\text{sen}^2(x)=1$ para simplificar a expressão trigonométrica **expr**

Exemplo 1.8 Simplificando expressões trigonométricas.

Maxima

```
(%i33)trigsimp(tan(a));
```

(%o33) $\frac{\sin(a)}{\sin(b)}$

```
(%i34)trigexpand(tan(a+b));
```

```
(%o34)  (tan(b)+tan(a))/(1-tan(a)*tan(b))
(%i35)trigsimp(2*sin(x)^2+cos(x)^2);
(%o35)  sin(x)^2+1
```

1.5 Resolvendo equações

Podemos resolver equações algébricas usando o seguinte comando:

solve(eq=0,x) - resolve a equação **eq** para a variável x.

A saída do comando **solve** é uma lista do tipo $[x_1, x_2, x_3, ...]$ com x_i (i= 1,2,3...) as soluções. Se omitirmos a igualdade, o Maxima assumirá o argumento do comando **solve** como uma equação.

Exemplo 1.9 Resolva as seguintes equações:

a) $x^4 - x^2 - 1 = 0$ b) $x^5 - 4x + 2 = 0$ c) $\cos(x) - x = 0$.

Maxima

```
(%i36)solve(x^4-x^2-1=0,x);
(%o36)  [x=-sqrt(sqrt(5)+1)/sqrt(2),x=sqrt(sqrt(5)+1)/sqrt(2),x=-sqrt(sqrt(5)-1)%i/sqrt(2),
        x=sqrt(sqrt(5)-1)%i/sqrt(2)]
(%i37)solve(x^5-4*x+2=0,x);
(%o37)  [0=x^5-4*x+2]
(%i38)solve(cos(x)=x,x);
(%o38)  [x=cos(x)]
```

1.5.1 Resolvendo equações numericamente

Note que, o comando **solve** não resolve as equações dadas nos itens (b) e (c). Podemos resolvê-las numericamente com os seguintes comandos:

find_root(f(x)=0, x, a, b)	-	método da bissecção;
newton(f(x)=0,x, x_0, imax)	-	método de Newton;
mnewton([f_1,..., f_n], [x_1,..., x_n],[x_1^0,..., x_n^0])	-	método de Newton para sistema de equações.

Para o método da bissecção a função deve mudar de sinal nos extremos do intervalo. Se essa condição não for satisfeita, o método será controlado pelo comando **find_root_error**. Se o valor deste comando for **true** ocorreram erros; caso contrário será retornado um valor. Para o método de Newton, x_0 é o chute inicial e o critério de parada é **abs(expr)** < **imax**. Para usarmos o comando **newton** devemos carregar o pacote **newton1** e para usarmos o comando **mnewton** devemos carregar o comando **mnewton**.

Exemplo 1.10 Resolva numericamente as equações dos itens (b) e (c) do exemplo 1.9

Maxima

```
(%i39)find_root(x^5-4*x+2=0,x,-1,1);

(%o39)   0.50849948465733
(%i40)load(newton1)$
(%i41)newton(cos(x)-x,x,1,1/100);

(%o40)   0.73911289091136
```

Exemplo 1.11 Resolva numericamente o seguinte sistema:

$$\begin{cases} x_1 + 3\ln(x_1) - x_2^2 &= 0 \\ 2x_1^2 - x_1 x_2 - 5x_1 + 1 &= 0 \end{cases}$$

Maxima

```
(%i42)load(mnewton)$
(%i43)mnewton([x1+3*log(x1)-x2^2, 2*x1^2-x1*x2-5*x1+1],
      [x1, x2], [5, 5]);
(%o41)  [[x1 = 3.756834008012769, x2 = 2.779849592817898]]
```

1.6 Listas

1.6.1 Gerando listas

Uma *lista* para o Maxima é uma série de objetos entre colchetes e separados por vírgulas. As listas são geradas pelos seguintes comandos:

makelist(expr,i,i_1,i_2)	-	gera uma lista avaliando **expr** para $i_1 \leq i \leq i_2$.
create_list(expr, i_1, L_1,..., i_n, L_n)	-	gera uma lista avaliando **expr** com i_1 associado a cada elemento da lista L_1 e para cada associação indexa i_2 à cada elemento da lista L_2,...

Exemplo 1.12 (a) Gere uma lista com os 30 primeiros inteiros.

(b) Gere uma lista formada pelos elementos x, x^2, x^3, x^4.

(c) Obtenha o 15° elemento da lista do item (a) e o 3° elemento da lista do item (b).

Soluções:

Maxima

```
(%i44)L1:makelist(i,i,1,30)
```

(%o42) $[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]$

```
(%i45)L2:makelist(x^i,i,1,4);
```

(%o43) $[x,x^2,x^3,x^4]$

```
(%i46)L1[15];L2[3]
```

(%o44) 15
$3x^2$

Exemplo 1.13 (a) Gere uma lista formada pelos elementos $(f(1),g(a))$, $(f(1),g(b))$, $(f(2),g(a))$, $(f(2),g(b))$.
(b) Gere uma lista cujos elementos são os gráficos das funções sen(x), sen$(x+1)$, sen$(x+2)$, sen$(x+2)$, sen$(x+4)$ e desenhe esses gráficos com $x \in [0, 2\pi]$, no mesmo sistema de coordenadas.

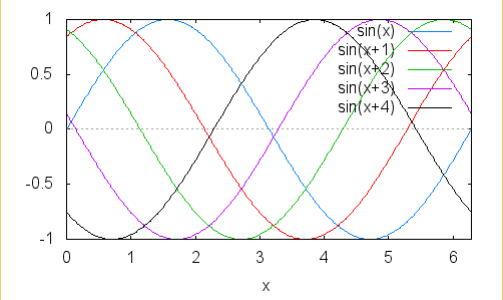
Soluções:

Maxima

```
(%i47)create_list([f(i),g(j)],i,[1,2],j,[a,b]);
```

(%o45) $[[f(1),g(a)],[f(1),g(b)],[f(2),g(a)],[f(2),g(b)]]$

```
(%i48)plot2d(makelist(sin(k+x),k,0,4),[x,0,2*%pi]);
```



(%o46)

1.6.2 Manipulando listas

Para manipularmos listas usaremos os seguintes comandos:

L **[i]**	-	i-ésimo elemento da lista L
apply(opr,L)	-	aplica o operador *opr* na lista L
append(L_1, L_2, ..., L_n)	-	junta as lista L_1, L_2, ..., L_n
map(opr, L)	-	aplica o operador *opr* na lista L.

As operações aritméticas do Maxima são operadores que podem ser usados em lista com o comando **apply**. Por exemplo, **apply("+",L)** aplica o operador *soma* na lista "L".

Exemplo 1.14 (a) Calcule $1^3+2^3+3^3+4^3+5^3+6^3+7^3+8^3+9^3+10^3$

(b) Gere uma lista com os elementos a, x^2, sen(x) e $g(x)$ e calcule o valor de f nessa lista.

(c) Junte as listas $[a,b,c,d,e,f,g,h]$, $[1,2,3]$, $[x^2,y,3z,w]$.

Soluções:

Maxima

```
(%i49)apply("+", makelist(i^3,i,1,10));
(%o47)  3025
(%i50)map(f,[a,x^2, sin(x),g(x)]);
(%o48)  [f(a),f(x^2),f(sin(x)),f(g(x))]
(%i51)append([a,b,c,d,e,f,g,h],[1,2,3],[x^2,y,3*z,w]);
(%o49)  [a,b,c,d,e,f,g,h,1,2,3,x^2,y,3z,w]
```

1.6.3 Vetores & Matrizes

Observe que *listas* podem ser consideradas "vetores" e podemos efetuar as operações usuais dos vetores com os operadores aritméticos do Maxima. Nesse contexto, o produto escalar entre vetores é efetuado com o ponto **.**

Exemplo 1.15 Considere os vetores $u = (x, y, z)$ e $v = (a, b, c)$. Efetue $u + v$, $u - v$, $2u$, $u.v$.

Soluções:

Maxima

```
(%i52)u:[x,y,z]; v:[a,b,c];

(%o50)  [x,y,z][a,b,c]
(%i53)u+v

(%o51)  [x+a,y+b,z+c]
(%i54)u-v

(%o52)  [x-a,y-b,z-c]
(%i55)2*u

(%o53)  [2*x,2*y,2*z]
(%i56)u.v

(%o54)  cz+by+ax
```

Matrizes

Podemos gerar uma matriz com o Maxima usando o seguinte comando:

matrix($[a_{11},a_{12},\ldots,a_{1m}]$, $[a_{21},a_{22},\ldots,a_{2m}]$,...,$[a_{n1},a_{n2},\ldots,a_{nm}]$) - gera uma matriz $n \times m$.

Podemos usar os operadores aritméticos do Maxima para operar com as matrizes. O produto entre matrizes será efetuado com o ponto **.**

Exemplo 1.16 Considere as matrizes

$$A = \begin{bmatrix} 3 & 5 & 1 \\ -2 & 0 & 2 \end{bmatrix}; \quad B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 4 & 1 \end{bmatrix}; \quad C = \begin{bmatrix} 4 & 6 & 1 \\ 2 & 1 & 1 \end{bmatrix}.$$

Efetue $A + C$, $A - C$, $5B$, $A.B$.

Soluções:

Maxima

```
(%i57)A:matrix([3,5,1],[-2,0,2])
```

(%o55) $A = \begin{bmatrix} 3 & 5 & 1 \\ -2 & 0 & 2 \end{bmatrix}$

```
(%i58)B:matrix([2,1],[1,3],[4,1])
```

(%o56) $B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 4 & 1 \end{bmatrix}$

```
(%i59)C:matrix([4,6,1],[2,1,1])
```

(%o57) $C = \begin{bmatrix} 4 & 6 & 1 \\ 2 & 1 & 1 \end{bmatrix}$

```
(%i60)A+C
```

(%o58) $\begin{bmatrix} 7 & 11 & 2 \\ 0 & 1 & 3 \end{bmatrix}$

```
(%i61)A-C
```

(%o59) $\begin{bmatrix} -1 & -1 & 0 \\ -4 & -1 & 1 \end{bmatrix}$

```
(%i62)5*B
                    [ 10   5  ]
                    [         ]
(%o60)              [  5   15 ]
                    [         ]
                    [ 20   5  ]
(%i63)A.B
                    [ 15   19 ]
(%o61)              [         ]
                    [  4   0  ]
```

1.7 Progressões

Para usarmos os comandos referentes as progressões carregamos o pacote **functs**.

arithmetic(a, d, n)	-	retorna o n-ésimo termo da progressão aritmética $a, a+d, a+2d, \ldots, a+(n-1)d$
geometric(a, r, n)	-	retorna o n-ésimo termo da progressão geométrica $a, ar, ar^2, \ldots, ar^{(n-1)}$
geosum(a, r, n)	-	retorna a soma dos elementos da progressão geométrica
arithsum(a, d, n)	-	retorna a soma dos elementos da progressão aritmética de 1 a n. Se n for infinito (**inf**) então a soma será finita se e somente se o valor absoluto de r for menor que 1.

Exemplo 1.17 (a) Determine o décimo termo da PA $(2, 8, 14, \ldots)$.

(b) Qual a soma dos dez primeiros termos da PA $(2, 8, 14, \ldots)$.

(c) Calcule a soma dos n primeiros números ímpares $(1, 3, 5, 7, \ldots, 2n-1, \ldots)$.

(d) Calcule o décimo quinto termo da PG $(1, 3, 9, 27, \ldots)$.

(e) Calcule a soma dos dez primeiros termos da PG $(1, 3, 9, 27, \ldots)$.

(f) Calcule a soma de todos os termos da PG $(1/2, 1/4, 1/8, \ldots)$.

Soluções:

Maxima

```
(%i64)load(functs)$
(%i65)arithmetic(2,6,10)
(%o62)  56
(%i66)arithsum(2,6,10)
(%o63)  290
(%i67)arithsum(1,2,n)
(%o64)  n^2
(%i68)geometric(1,3,15)
(%o65)  4782969
(%i69)geosum(1,3,10)
(%o66)  29524
(%i70)geosum(1/2,1/2,inf)
(%o67)  1
```

1.8 Gráficos

1.8.1 Gráficos 2D

Para desenharmos o gráfico de uma função $f(x)$ (ou mais funções) ou uma curva plana definida(s) no intervalo $[a,b]$, em coordenadas cartesianas, usamos os seguintes comandos:

plot2d(f(x),[x,a,b], [opções])	-	desenha o gráfico da função $f:[a,b]\to\mathbb{R}$
plot2d([discrete,Lx,Ly],[opções])	-	desenha retas ligando os pontos das listas Lx e Ly

plot2d([L_1],[x,a,b], [opções])	-	desenha o gráfico de uma lista L_1 de funções
plot2d ([parametric,x(t),y(t), [t,a,b]],[opções])	-	desenha o traço da curva $(x(t), y(t))$ com $t \in [a,b]$

Podemos usar diretamente o comando gráfico **plot2d** do Maxima acionando a janela **Gráfico 2D** localizada na parte superior da área de trabalho. A opção **padrão**, desenha o gráfico numa janela auxiliar e a opção **embutido**, desenha o gráfico na área de trabalho do Maxima. Também podemos obter o gráfico diretamente na área de trabalho do Maxima digitando **wx** seguida do comando **plot2d**.

Todas as opções do comando **plot2d** são listas, sendo a primeira entrada o nome da opção. Por exemplo, **[nticks, 20]** usa 20 pontos para desenhar o gráfico ou o traço da curva. O Maxima representa os desenhos do comando **plot2d** numa proporção de 4 para 3 entre os eixos horizontal e vertical, portanto para obter a mesma escala nos dois eixos usa-se a opção **"set size square"**.

O Maxima usa os programas auxiliares Gnuplot, Openmath e XMaxima para desenhar os gráficos. No caso do Gnuplot, usa-se o comando **gnuplot_preamble** para implementar as opções do **plot2d**.

Exemplo 1.18 Desenhe o gráfico das seguintes funções:

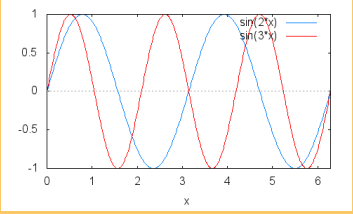
(a) $f(x) = \text{sen}(2x)$ e $g(x) = \text{sen}(3x)$ no intervalo $[0, 2\pi]$ e no mesmo sistema de coordenadas.

(b) $f(x) = \dfrac{1}{x}$ no intervalo [-1, 1], com a variação da imagem de f de -100 a 100.

(c) Desenhe o círculo de centro $(0,0)$ e raio 1.

Maxima

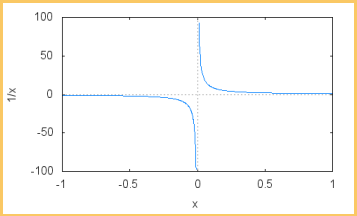
```
(%i71)plot2d([sin(2*x),sin(3*x)],[x,0,2*%pi]);
```



(%o68)

Maxima

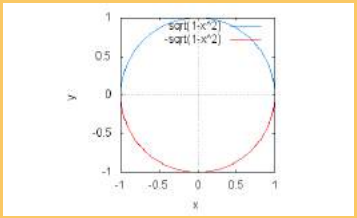
```
(%i72)plot2d(1/x,[x,-1,1],[y, -100,100])
```




```
(%o69)
```

Maxima

```
(%i73)plot2d([sqrt(1-x^2),-sqrt(1-x^2)],[x,-1,1],[y,-1,1],
      [gnuplot_preamble, `set size ratio 1;
      set zeroaxis;"]);
```



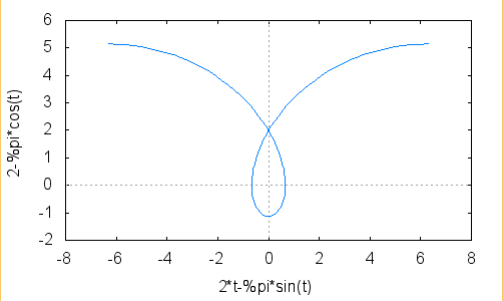

```
(%o70)
```

Exemplo 1.19 Desenhe a curva $(2t - \pi \operatorname{sen}(t), 2 - \pi \cos(t))$ com $t \in [-\pi/2, \pi/2]$.

Solução:

Maxima

```
(%i74)plot2d([parametric,2*t-%pi*sin(t),2-%pi*cos(t),[t,-%pi,%pi]],
      [nticks,100])
```



(%o71)

Exemplo 1.20 Desenhe o diagrama que liga os pontos $L_1 = \{3,5,3,3,5,3,3,3,1\}$ e $L_2 = \{3,3,1,3,3,3,5,3,3\}$ e os pontos $\{(3,1),(5,3),(3,5),(1,3),(3,1)\}$ no mesmo sistema cartesiano.

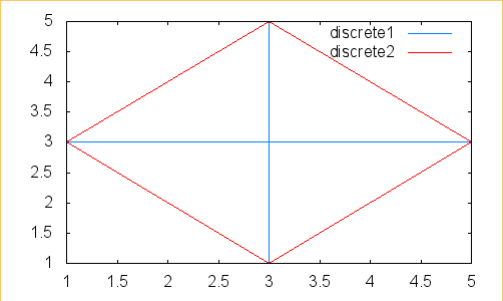
Solução:

Maxima

```
(%i75)Lx:[3,5,3,3,5,3,3,3,1];

(%o72)  [3,5,3,3,5,3,3,3,1]
(%i76)Ly:[3,3,1,3,3,3,5,3,3];

(%o73)  [3,3,1,3,3,3,5,3,3]
(%i77)plot2d([[discrete,Lx,Ly],[discrete,[[3,1],[5,3],
      [3,5],[1,3],[3,1]]]]);
```



(%o74)

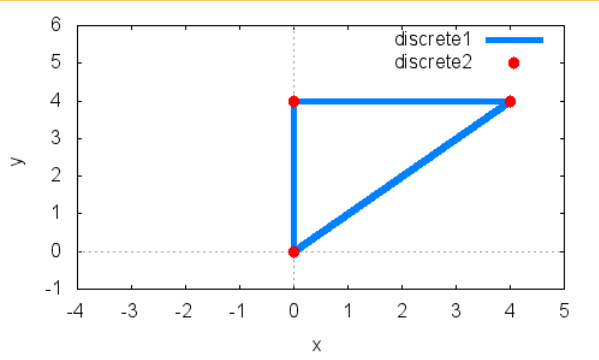
Exemplo 1.21 Desenhe o diagrama que liga os pontos $\{(0,0),(4,0),(4,4)\}$, destacando as retas e os pontos.

Solução:

Maxima

```
(%i78)xy:[[0,0],[0,4],[4,4]];

(%o75)  [[0,0],[0,4],[4,4]]
(%i79)plot2d([[discrete,[0,0,4,0],[0,4,4,0]],[discrete,xy]],
      [x,-4,5],[y,-1,6],[style,[line,5.0],[points,3,2,1]]);
```




```
(%o76)
```

1.8.2 Gráficos 3D

Para desenharmos o gráfico de uma função $f(x,y)$ ou uma superfície definida no retângulo $[a,b]\times[c,d]$, em coordenadas cartesianas, usamos os seguintes comandos:

plot3d(f(x,y),[x,a,b], [y,c,d],[opções])	- desenha o gráfico da função $f:[a,b]\times[c,d]\to\mathbb{R}$
contour_plot(f(x,y),[x,a,b],[y,c,d],[opções])	- desenha as curvas de níveis de $f(x,y)$ com $x\in[a,b]$ e $y\in[c,d]$

plot3d([x(u,v),y(u,v), z(u,v)],x,a,b,y,c,d,[opções]) -

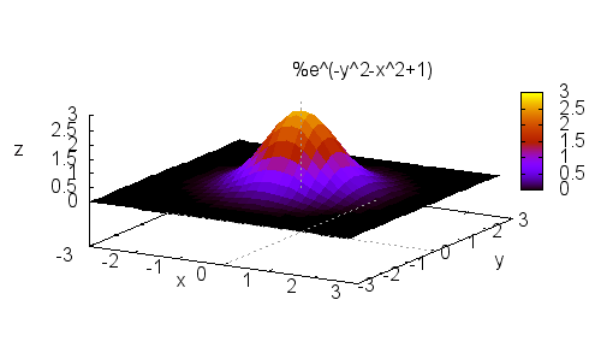
desenha uma superfície parame trizada por u e v com $u \in [a,b]$ e $v \in [c,d]$

Exemplo 1.22 Desenhe o gráfico da função $f(x,y) = e^{1-x^2-y^2}$ no domínio $(x,y) \in [-3,3] \times [-3,3]$.

Solução:

Maxima

```
(%i80)plot3d(exp(1-x^2-y^2),[x,-3,3],[y,-3,3]);
```



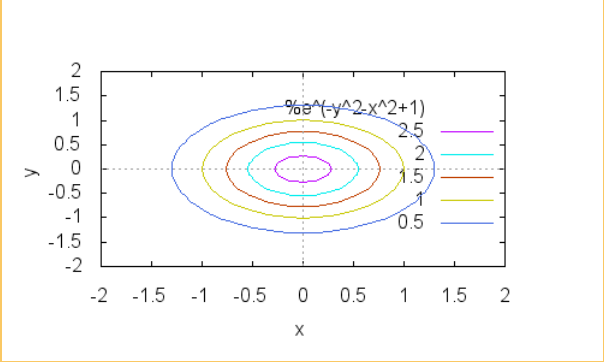

```
(%o77)
```

Exemplo 1.23 Desenhe as curvas de níveis da função $f(x,y) = e^{1-x^2-y^2}$ no domínio $(x,y) \in [-2,2] \times [-2,2]$.

Solução:

Maxima

```
(%i81)contour_plot(exp(1-x^2-y^2),[x,-2,2],[y,-2,2]);
```



(%o78)

1.8.3 Coordenadas polares

A localização de um ponto P do plano no sistema de coordenadas polares é determinada pela distância r de P ao pólo O e um ângulo θ orientado, no sentido anti-horário, formado entre r e a semi-reta horizontal de origem em O, chamada eixo polar. Podemos desenhar gráficos em coordenadas polares com o comando **plot2d** e a opção **set polar** do seguinte modo:

> **plot2d**([$r(\theta)$],[θ,θ_1,θ_2], [gnuplot_preamble,"set polar; opções;"]) - desenha o gráfico em coordenadas polares.

Atenção: para obter a melhor visualização do gráfico, use a opção $[y,y_1,y_2]$ para ajustar a janela de visualização.

Exemplo 1.24 Desenhe o gráfico em coordenada polares das seguintes funções:

a) Círculo: $r = 1$;

b) Cardióide: $r = 1 - \cos(\theta)$

c) Rosácea: $r = \cos(2\theta)$;

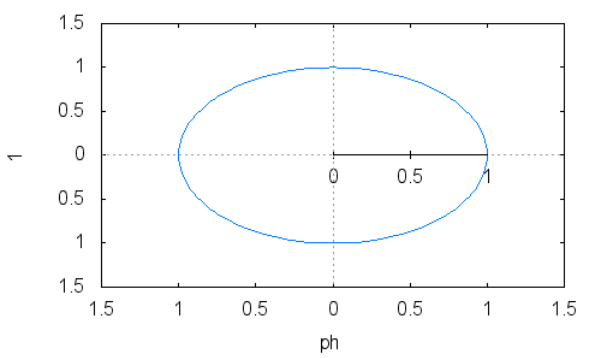
d) Concóide: $r = 2 - \sec(\theta)$

e) Espiral de Arquimedes: $r = \theta$

Solução:

Maxima

```
(%i82)plot2d([1],[ph,0,2*%pi],[y, -2,2], [gnuplot_preamble,
      "set polar"]);
```

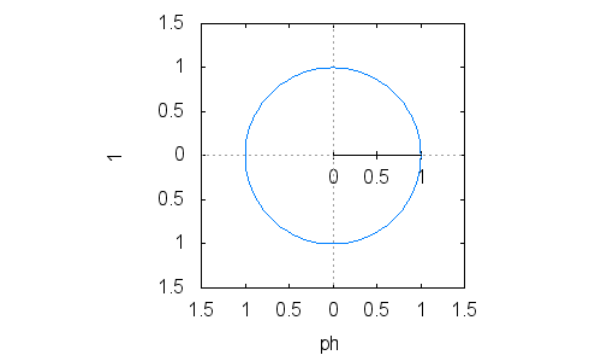



```
(%o79)
```

Observe que o círculo foi desenhado como uma elipse!! Para ajustar a janela de visulaização usaremos a opção **set size square**.

Maxima

```
(%i83)plot2d([1],[ph,0,2*%pi],[y, -2,2], [gnuplot_preamble,
      "set polar;set size square"]);
```

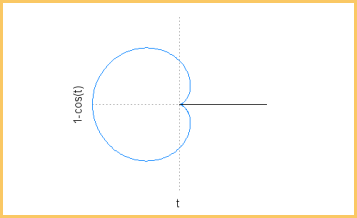



```
(%o80)
```

Nos seguintes gráficos, vamos usar as opções **unset border** e **unset tic** remover a numeração e a borda a implementação gráfica do Maxima.

Maxima

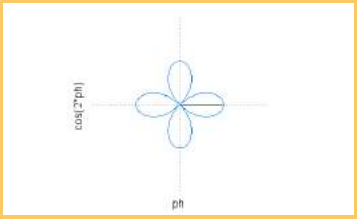
```
(%i84)plot2d([1-cos(ph)],[ph,0,2*%pi],[y, -2,2],
      [gnuplot_preamble, "set polar; set size square;
      unset border;unset tic;"]);
```




```
(%o81)
```

Maxima

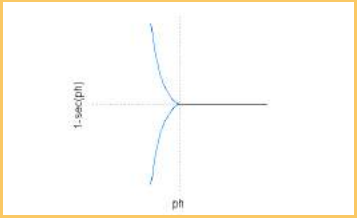
```
(%i85)plot2d([cos(2*ph)],[ph,0,2*%pi],[y, -2,2],
      [gnuplot_preamble,"set polar; set size square;
      unset border;unset tic;"]);
```




```
(%o82)
```

Maxima

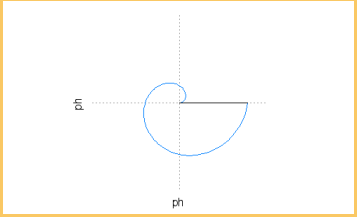
```
(%i86)plot2d([1-sec(ph)],[ph,0,2*%pi],[y, -2,2],
      [gnuplot_preamble, "set polar; set size square;
      unset border;unset tic;"]);
```



(%o83)

Maxima

```
(%i87)plot2d([ph],[ph,0,2*%pi],[y, -8,8],
      [gnuplot_preamble,
      "set polar; set size square;unset border;
      unset tic;"]);
```



(%o84)

Opções do comando plot

No que segue, destacamos algumas opções do comando **plot2d** e **plot3d**. "Opções" são listas contendo dois ou três elementos, sendo o primeiro elemento o nome da opção e os outros os valores associados a esta opção.

[x, ax, bx]	-	amplitude de x
[y, ay, by]	-	amplitude de y

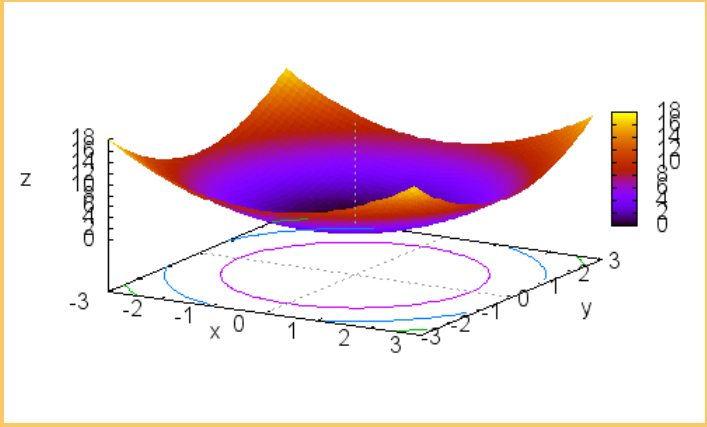
[nticks, n]	-	"n" pontos usados para montagem do gráfico, tem padrão n=10
[grid, n, n]	-	malha da montagem de gráfico 3D
gnuplot_preamble	-	insere comandos de montagem; os comandos devem estar entre aspas e separados por ponto e vírgula
gnuplot_pm3d	-	controla o modo avançado PM3D de montagem de gráfico
unset key	-	omite a fórmula da função.

Exemplo 1.25 Desenhe o gráfico e as curvas de níveis da função $z = x^2 + y^2$ no mesmo sistema cartesiano.

Solução:

Maxima

```
(%i88)meupreambulo:"set pm3d at s;unset surface;set contour;
      unset key";
(%o85)  set pm3d at s; unset surface; set contour; unset key
(%i89)plot3d(x^2+y^2,[x,-3,3],[y,-3,3],[grid,50,50],
      [gnuplot_pm3d, true],
      [gnuplot_preamble,meupreambulo]);
```



(%o86)

1.8.4 Gráficos com draw

O pacote **draw** é um módulo externo à biblioteca básica do `Maxima` que permite desenharmos gráficos 2D e 3D com relativa comodidade. Este pacote foi extensamente desenvolvido por Mário Riotorto [12] e para ser executado precisa ser primeiro carregado na memória do computador com o comando **load**. O pacote é muito versátil e possui vários comandos, destacamos os seguintes:

draw2d(opções, objetos)	-	desenha os gráficos 2D
draw3d(opções, objetos)	-	desenha os gráficos 3D

Existem vários comandos que designam as entradas "objetos gráficos", destacamos os seguintes:

Objetos gráficos 2D

explicit(f(x),x,a,b)	-	gráfico da função $f : [a,b] \to \mathbb{R}$;
parametric(x(t),y(t),t,t_1,t_2)	-	gráfico da curva $(x(t), y(t))$ com $t \in [t_1, t_2]$
implicit(eq,x,x_1,x_2,y,y_1,y_2)	-	traço da curva $(x, y(x))$ dada implicitamente na equação **eq** com $x \in [x_1, x_2]$,$y \in [y_1, y_2]$
polar(r(θ), θ,θ_1,θ_2)	-	gráfico de uma função r(θ) em coordenadas polares, com $\theta \in [\theta_1, \theta_2]$
points(Lx,Ly)	-	pontos das listas Lx e Ly que contém os valores de x e y, respectivamente
region(eq(x,y) $\leq$ 0,x,a,b,y,c,d,)	-	desenha regiões definidas por desigualdades
polygon(Lx,Ly)	-	poligno com vértices dados nas listas Lx eLy
rectangle(p_1,p_2)	-	retângulo com vértices oposto dados em p_1=[px,py] e p_2=[px,py]
ellipse(x_0,y_0,a,b,θ_1,θ_2)	-	elipse com centro em (x_0, y_0) e divisões no eixo de comprimento $\frac{a+b}{2}$; θ_1 ângulo inicial e θ_2 ângulo final
vector($[x_1, y_1]$,$[x_2, y_2]$)	-	vetor com extremidades (x_1, y_1) e (x_2, y_2)

Exemplo 1.26 Desenhe os seguintes objetos gráficos:

(a) $f(x) = 2\,\mathrm{sen}^2(x) + 2\cos(x)$, $x \in [-\pi, \pi]$;

(b) $(x(t), y(t)) = (cos^3(t), 2\,\mathrm{sen}^3(t))$, $t \in [0, 2\pi]$;

(c) $-x^4 + y^4 = 1$ com $y = y(x)$, $x \in [-4, 4]$ e $y \in [-4, 4]$;

(d) $r = \cos(4t)$, $t \in [0, 2\pi]$;

(e) Desenhe todos os objetos gráficos dados acima no mesmo sistema cartesiano;

(f) A região $\frac{4x^2}{y^2} + \frac{y^2}{4} \leq 1$ com $(x, y) \in [-1, 1] \times (0, 2]$.

Soluções:

Maxima

```
(%i90)load(draw)$
```

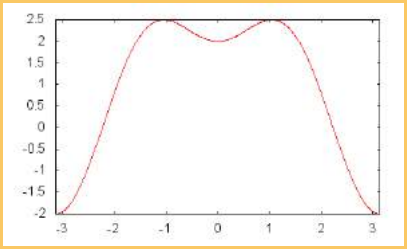
Solução: (a)

Maxima

```
(%i91)g1:explicit(2*sin(x)^2+2*cos(x),x,-%pi,%pi);
```

(%o87) $\text{explicit}(2*\sin(x)^2+2*\cos(x),x,-\%pi,\%pi)$

```
(%i92)draw2d(nticks=200,color=red,g1)
```



(%o88)

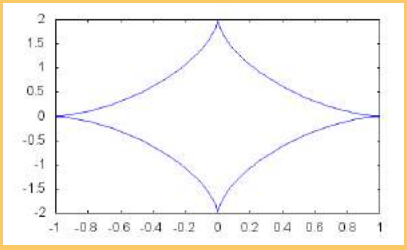
Solução: (b)

Maxima

```
(%i93)g2:parametric(cos(t)^3,2*sin(t)^3,t,0,2*%pi);
```

(%o89) $\text{parametric}(\cos(t)^3,2*\sin(t)^3,t,0,2*\%pi)$

```
(%i94)draw2d(nticks=200,color=blue,g2)
```



(%o90)

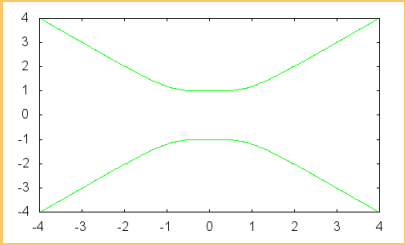
Solução: (c)

Maxima

```
(%i95)g3:implicit(-x^4+y^4=1,x,-4,4,y,-4,4);
```

(%o91) $\text{implicit}(-x^4+y^4=1,x,-4,4,y,-4,4)$

```
(%i96)draw2d(nticks=200,color=green,g3)
```



(%o92)

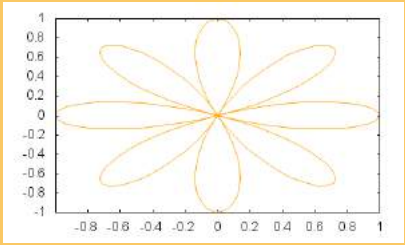
Solução: (d)

Maxima

```
(%i97)g4:polar(cos(4*t),t,0,2*%pi);
```

(%o93) $\text{polar}(\cos(4*t)+1,t,0,2*\%\text{pi})$

```
(%i98)draw2d(nticks=200,color=orange,g4)
```

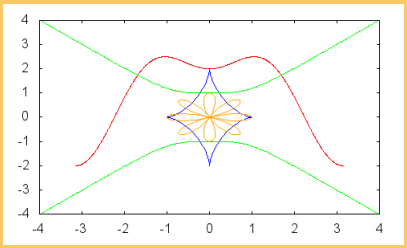


(%o94)

Solução: (e)

Maxima

```
(%i99)draw2d(nticks=200,color=red,g1,color=blue,g2,
      color=green,g3,color=orange,g4)
```

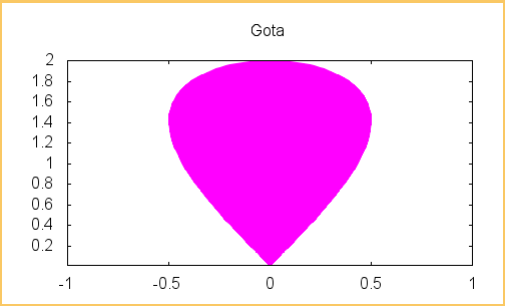



```
(%o95)
```

Solução: (f)

Maxima

```
(%i100)wxdraw2d(
       title = "Gota",nticks=400,fill_color=magenta,
       region(4*x^2/y^2+y^2/4<=1,x,-1,1,y,.1,2));
```




```
(%o96)
```

Objetos gráficos 3D

explicit(f(x,y),x,a,b,y,c,d)	-	gráfico da função $f : [a,b] \times [c,d] \to \mathbb{R}$;
parametric(x(t),y(t),,z(t),t,t_1,t_2)	-	gráfico da curva $(x(t), y(t), z(t))$ com $t \in [t_1, t_2]$
implicit(eq,x,x_1,x_2,y,y_1,y_2,z,z_1,z_2)	-	traço da curva $(x, y, z(x, y))$ dada implicitamente na equação **eq** com $x \in [x_1, x_2]$ e $y \in [y_1, y_2]$
points(Lx,Ly, Lz)	-	pontos das listas Lx, Ly e Lz que contém os valores de x, y e z, respectivamente
cylindrical(r(z,θ),z,z_1,z_2,θ,θ_1,θ_2)	-	superfície em coordenadas cilíndricas com $z \in [z_1, z_2]$; $\theta \in [\theta_1, \theta_2]$
spherical(r(θ,ϕ),θ,θ_1,θ_2,ϕ,ϕ_1,ϕ_2)	-	superfície em coordenadas espéricas com $\theta \in [\theta_1, \theta_2]$, $\phi \in [\phi_1, \phi_2]$
vector($[x_1, y_1, z_1]$,$[x_2, y_2, z_2]$)	-	vetor com extremidades (x_1, y_1, z_1) e (x_2, y_2, z_2)

parametric_surface(x(u,v),y(u,v),,z(u,v),u,u_1,u_2,v,v_1,v_2) - gráfico da superfície parametrizada $(x(u,v), y(u,v), z(u,v))$ com $u \in [u_1, u_2]$; $v \in [v_1, v_2]$

Exemplo 1.27 Desenhe os seguintes objetos gráficos:

(a) $f(x,y) = 2\,\mathrm{sen}(xy)$, $x \in [-2, 2]$;

(b) $\rho = 4\cos(\phi)$, $\phi \in [0, 2\pi]$;

(c) $x^2 + y^2 = z^2$ com $x \in [-1, 1]$, $y \in [-1, 1]$ e $z \in [0, 1]$;

(d) Desenhe os objetos gráficos dos itens (a) e (b) no mesmo sistema cartesiano.

Soluções:

Maxima

```
(%i101)load(draw)$
```

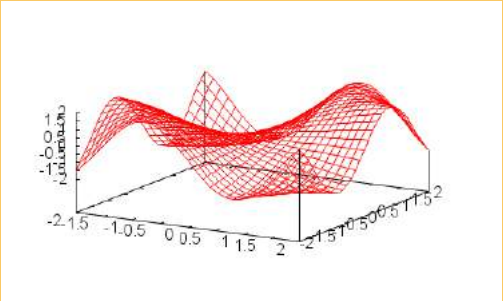
Solução: (a)

Maxima

```
(%i102)gr1:explicit(2*sin(x*y),x,-2,2,y,-2,2);
```

(%o97) $\text{explicit}(2*\sin(x*y),x,-2,2,y,-2,2)$

```
(%i103)draw3d((surface_hide=true,color=red,gr1))
```



(%o98)

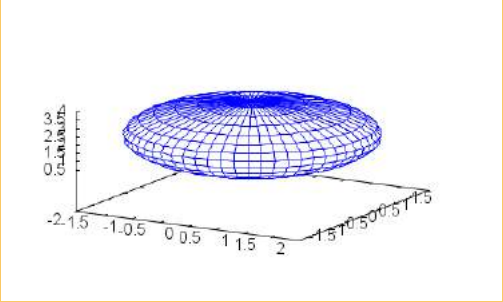
Solução: (b)

Maxima

```
(%i104)gr2:spherical(4*cos(phi),theta,0,2*%pi,phi,-%pi,%pi);
```

(%o99) $\text{spherical}(4*\cos(\text{phi}),\text{theta},0,2*\%\text{pi},\text{phi},-\%\text{pi},\%\text{pi})$;

```
(%i105)draw3d((surface_hide=true,color=blue,gr2))
```



(%o100)

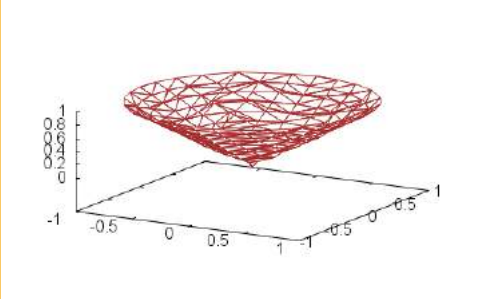
Solução: (c)

Maxima

```
(%i106)gr3:implicit(x^2+y^2=z^2,x,-1,1,y,-1,1,z,0,1);
```

(%o101) $\text{implicit}(x^2+y^2=z^2,x,-1,1,y,-1,1,z,0,1);$

```
(%i107)draw3d((surface_hide=true,color=brown,gr3))
```

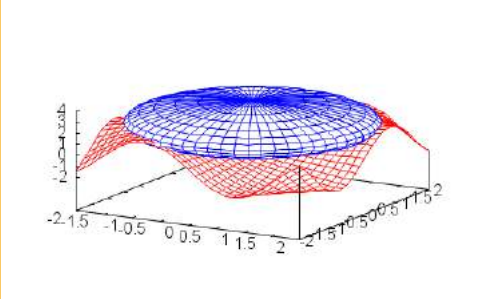


(%o102)

Solução: (d)

Maxima Solução (d)

```
(%i108)wxdraw3d(surface_hide=true,color=red,gr1,color=blue,gr2);
```



(%o103)

Opções dos comandos draw2d e draw3d

No que segue, destacamos algumas opções do comando **draw2d** e **draw3d**. Em geral, "opções" são listas contendo dois ou três elementos, sendo o primeiro elemento o nome da opção e os outros os valores associados a esta opção.

Opções gerais

color=nome	-	cor das linhas
fill_color=nome	-	cor de retângulos, polignos e círculos
x(yz)range=[min,max]	-	amplitude da coordenada x(yz)
grid=true/false	-	desenha uma grade,se for "true"
title=nome	-	título principal
x(yz)label=nome	-	título do eixo x(yz)

Opções para objetos 2D

filled_func=true/false	-	se for "true", preenche a área entre o gráfico da função e o eixo x
filled_func= f	-	se for "true", preenche a área entre o gráfico da função e o gráfico de f
borde=true/false	-	se for "true", desenha a fronteira de polígonos
line_width=valor	-	espessura de linhas

Opções para objetos 3D

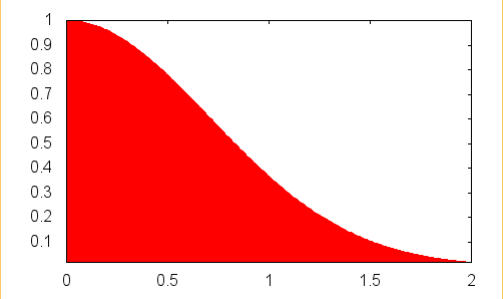
surface_hide=true/false	-	controla a visibilidade de partes ocultas
enhanced3d=true/false	-	se for “true"desenha colorido
contour=valor	-	desenha as curvas de níveis; valor = none, base, surface, both, map
contour_levels=n	-	n curvas de níveis defindas por **contour**
contour_levels=[x_1,h,x_2]	-	desenha curvas de níveis, defindas por **contour**, entre x_1 e x_2 com passo h
contour_levels={x_1,x_2, ...}	-	desenha curvas de níveis,defindas por **contour**, em x_1, x_2, ...

Exemplo 1.28 Desenhe em vermelho a área limitada pelo eixo x e a função e^{-x^2} para $x \in [0, 2]$.

Solução:

Maxima

```
(%i109)load(draw)$
(%i110)draw2d(color=red,filled_func=true,explicit(exp(-x^2),x,0,2));
```



(%o104)

Exemplo 1.29 (a) Desenhe o gráfico e as curvas de níveis da função $z = x^2 + y^2$ no mesmo sistema cartesiano;

(b) Desenhe as curvas de níveis da função $z = 3\,\text{sen}(|x|) + |y|$

(c) Desenhe uma pirâmide.

(d) Desenhe um cilidro de raio 1 e altura 6.

Soluções:

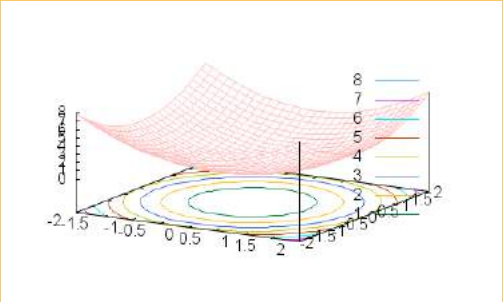
Maxima

```
(%i111)load(draw)$
```

Solução: (a)

Maxima

```
(%i112)draw3d(surface_hide=true,contour_levels=10,contour=base,
       color=pink,explicit(x^2+y^2,x,-2,2,y,-2,2));
```

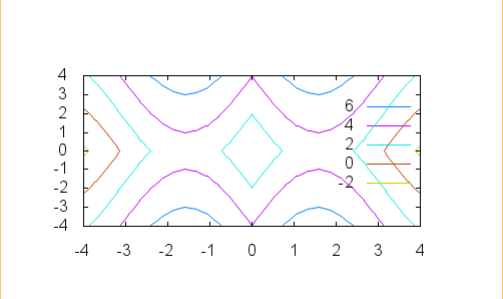


```
(%o105)
```

Solução: (b)

Maxima

```
(%i113)draw3d(surface_hide=true,contour=map,
       color=purple,explicit(3*sin(abs(x))+abs(y),x,-4,4,y,-4,4));
```

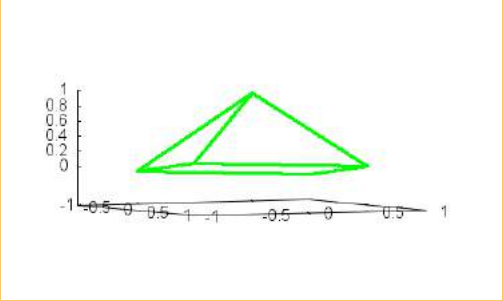


(%o106)

Solução: (c)

Maxima

```
(%i114)opcoes: [view=[96,27],surface_hide = true,color = green,
       line_width = 3];
(%o107)   [view = [96, 27], surface_hide = true, color = green, line_width = 3]
(%i115)triangulo(p1,p2,p3) := mesh([p1,p2], [p3,p3]);
(%o108)   triangulo(p1, p2, p3) := mesh([p1, p2], [p3, p3])
(%i116)quadrado(p1,p2,p3,p4):=mesh([p1,p2],[p3,p4])
(%o109)   quadrado(p1, p2, p3, p4) := mesh([p1, p2], [p3, p4])
(%i117)piramide :
       [triangulo([1,0,0],[0,1,0],[0,0,1]),
       triangulo([1,0,0],[0,-1,0],[0,0,1]),
       triangulo([-1,0,0],[0,1,0],[0,0,1]),
       triangulo([-1,0,0],[0,-1,0],[0,0,1]),
       quadrado([-1,0,0],[0,1,0],[0,-1,0],[1,0,0])]$
(%i118)draw3d(opcoes,piramide);
```

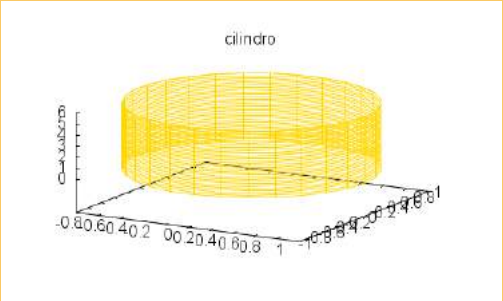


(%o110)

Solução (d)

Maxima

```
(%i119)wxdraw3d(surface_hide=false,color=gold,title="cilindro",
       cylindrical(1,z,0,6,t,0,2*%pi));
```



(%o111)

1.9 Comandos de programação

É sempre útil fazermos nossos próprios programas, pois otimiza vários processos principalmente processos iterativos. Podemos usar a linguagem de programação do aplicativo `Maxima` para implementar algoritmos e resolver vários problemas. No que segue, faremos uma breve introdução aos principais comandos de programação.

if cond then expr1 else expr2	- avalia a expressão **expr1** se a condição **cond** for verdadeira; caso contrário avalia **expr2**
block ([v_1, ...,v_m], $expr_1$, ... $expr_n$)	- avalia **expr**$_1$, ... **expr**$_n$ e retorna o valor de **expr**$_n$
return(valor)	- comando que antecipa uma saída
/* comentários */	- comentários

var:i_0, **while** cond **do** (expr(var(i)))	- avalia a expressão **expr** para **var(i)** com **i** iniciando em i_0 e finaliza quando a condição **cond** for falsa
var:i_0, **unless** cond **do** (expr(var(i)))	- avalia a expressão **expr** para **var(i)** com **i** iniciando em i_0 e finaliza quando a condição **cond** for verdadeira
for var(i) in L1 **do** expr(var(i))	- avalia a expressão **expr** para **var(i)** nos elementos da lista **L1**

for var(i): i_0 thru i_n step m do (expr(var(i))) -
avalia a expressão **expr** para **var(i)** com **i** variando de i_0 à i_n com passo **m**, finaliza quando **i** atinge o valor $\mathbf{i}_n$

for var(i): i_0 while cond step m do (expr(var(i))) -
avalia a expressão **expr** para **var(i)** com **i** iniciando i_0, com passo **m**, finaliza quando a condição **cond** for falsa

for var(i): i_0 unless cond step m do (expr(var(i))) -
avalia a expressão **expr** para **var(i)** com **i** iniciando i_0, com passo **m**, finaliza quando a condição **cond** for verdadeira

As declarações **do** e **for** são usadas em processo iterativos. As declarações **thru**, **while** e **unless** são critérios de parada para os comandos **do** e **for**. O processo iterativo é executado em ciclos: começa com o valor $\mathbf{i}_0$ para variável **var(i)** e avalia a expressão **expr(var(i))**; depois um incremento é adicionado ao parâmetro de iteração, conforme o passo **m** (se for o caso). Para **m=1**, o comando **step** pode ser omitido. Em seguida, avalia novamente a expressão **expr(var(i))** até que um critério de parada seja satisfeito. Para a declaração **for**, a saída de finalização do `Maxima` para um processo iterativo é **done**. Em geral, a "programação" com o `Maxima` gera uma nova função.

Exemplo 1.30 Faça um programa para calcular n! e calcule 6!

Solução:

Maxima

```
(%i120)fatorial(n):= if n < 1 then 1 else n*fatorial(n-1)$
(%i121)fatorial(6)

(%o112)  720
```

Agora, faremos um programa usando os comandos **block** e **do**:

Maxima

```
(%i122)fatorial1(n):=block ([var],var:1,while n> 1 do (var:n*var,
       n:n-1),var)$
(%i123)fatorial1(6)

(%o113)  720
```

Exemplo 1.31 (a) Some os dez primeiros números naturais;

(b) Faça um programa para obter o polinômio $4x^4+7x^3+9x^2+10x$.

Solução (a):

Maxima

```
(%i124)block([s],s:0,
       for i: 1 while i <= 10 do (s: s+i),s);
(%o114)  55
```

Solução (b):

Maxima

```
(%i125)block([p],p:0,
       for i: 1 thru 4 do
       for j: 1 thru i do (p:p+i*x^j),p);
(%o115)  4x^4+7x^3+9x^2+10x
```

Exemplo 1.32 Faça um programa para calcular os polinônios de Legendre $p_0(x) = 1$, $p_1(x) = x$, $n\,p_n(x) = (2n-1)\,x\,p_{n-1}(x) - (n-1)\,p_{n-2}(x)$. Calcule $p_3(x)$.

Solução:

Maxima

```
(%i126)Legendre(n,x):=block([],if n=0 then 1 else if n=1 then x else
       ((2*n-1)*x*Legendre(n-1,x)-(n-1)*Legendre(n-2,x))/n)$
(%i127)ratsimp(Legendre(3,x));
```

$$(\%o116) \quad \frac{5x^3 - 3x}{2}$$

1.9.1 Aplicações no Cálculo Numérico

Nesta seção apresentaremos implementações de certos tópicos do Cálculo Numérico.

Exemplo 1.33 (aproximação do π) Faça um programa para calcular o valor de π, com precisão dada, usando a série

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots\right)$$

Maxima

```
(%i128)aproxPI(n):=block([valor],
       valor:4*sum((-1)^(k+1)/(2*k-1),k,1,n),float(valor))$
(%i129)aproxPI(10000);
```

$$(\%o117) \quad 3.141492653590044$$

Exemplo 1.34 (norma) Faça um programa para calcular a norma euclidiana do vetor $\vec{x} = (x_1, x_2, \cdots, x_n)$, dada por

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Maxima

```
(%i130)norma(lista):=block([n:length(lista),norma],
       soma:sum(lista[i]^2,i,1,n), norma:float(sqrt(soma)))$
(%i131)x:[1,3,5,7,9,11,13,15,17,19,21,23,25]$
(%i132)norma(x);

(%o118)   50.64582904840241
```

Exemplo 1.35 (Método da Bissecção) Faça um programa para calcular uma raiz da equação $f(x) = x - \cos(x)$, usando o método da Bissecção.

Solução:

Maxima /* Método da Bissecção */

```
(%i133)bissecao(f,a,b,tol,imax):=(x:a,
       for i:1 thru imax do
       (solx[0]:x, x:float((a+b)/2),solx[i]:x,
       erro:float(abs((solx[i]-solx[0])/solx[i])),
       if f(a)*f(solx[i]) < 0 then b:solx[i] else a: solx[i],
       if erro <= tol then
       (print("erro relativo=",erro),
       print("solução com a precisão desejada=",x), kill(f),return(fim))
       else
       if i >= imax then (print("iteração=",i),print("erro relativo=",
       erro),
       print("solução aproximada:",x),
       kill(f), return(fim))  )) $
```

Maxima /* Função */

```
(%i134)f(x):= x-cos(x)$
```

*/ Verificação se a função muda de sinal nos extremos do intervalo */

```
(%i135)f(-1)*f(2)

(%o119)   -3.721596543649575
```

*/ Raíz aproximada */

```
(%i136)bissecao(f,-1,2,0.001,20);
```

```
  erro relativo=9.90425883129745841 10^-4
  solução com a precisão desejada=0.739501953125
(%o120)  fim
```

Exemplo 1.36 (Método de Newton) Faça um programa para calcular uma raiz da equação $f(x) = x - \cos(x)$, usando o método de Newton.

Solução:

```
Maxima /* Método de Newton */
(%i137)newton(f,x,tol,imax):= ([y, df, dfx], df: diff (f('x), 'x),
       for i: 1 thru imax do
       (solx[0]:x, y: ev(df), x: float(x - f(x)/y), solx[i]:x,
       erro:abs((solx[i]-solx[0])/solx[i]),
       if erro < tol then (print("erro relativo=",erro),
       print("solução com a precisão desejada:",x),
       kill(f),return(fim))))$
```

```
Maxima /* Função */
(%i138)f(x):= x-cos(x)$
  */ Raíz aproximada */
(%i139)newton(f,0.5,0.0001,10);
  erro relativo=7.6489468503479832 10^-5
  solução com a precisão desejada=0.739501953125
(%o121)  fim
```

Exemplo 1.37 (Eliminação Gaussiana) Faça um programa para resolver o sistema abaixo, usando o método de Eliminação gaussiana.

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 10 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 7 \\ 3x_1 + 2x_2 + x_3 + 2x_4 = 6 \\ 4x_1 + 3x_2 + 2x_3 + x_4 = 5 \end{cases}$$

Solução:

Maxima /* Método de Elininação gaussiana */

```
(%i140)EliminaGauss(A,n):=(M:triangularize(A),
        b:makelist(M[i,n+1],i,1,n),matrix(b),
        eq[i]:=sum(M[i,j]*x[j],j,1,n)=b[i], kill(A,M),
        xx:linsolve(makelist(eq[i],i,1,n),makelist(x[i],i,1,n)))$
```

Maxima

```
(%i141)A:matrix([1,2,3,4,10],[2,1,2,3,7],[3,2,1,2,6],[4,3,2,1,5]);
```

(%o122)
$$\begin{bmatrix} 1 & 2 & 3 & 4 & 10 \\ 2 & 1 & 2 & 3 & 7 \\ 3 & 2 & 1 & 2 & 6 \\ 4 & 3 & 2 & 1 & 5 \end{bmatrix}$$

```
(%i142)EliminaGauss(A,4);
```

(%o123) $[x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 2]$

2 Cálculo Diferencial e Integral

2.1 Somatório e limite

Podemos efetuar o cálculo de somatórios e limites com os seguintes comandos:

sum(f(i), i, 1, n)	-	$\sum_{i=1}^{n} f(i)$
sum(**sum**(f(i,j), i, 1, n),j,1,m)	-	$\sum_{j=1}^{m}\sum_{i=1}^{n} f(i,j)$
limit(f(x),x,a,plus)	-	$\lim_{x\to a^+} f(x)$
limit(f(x),x,a,minus)	-	$\lim_{x\to a^-} f(x)$
limit(f(x),x,a,inf)	-	$\lim_{x\to+\infty} f(x)$
limit(f(x),x,a,minf)	-	$\lim_{x\to-\infty} f(x)$.

Destacamos os seguintes comandos de saída **und** e **ind** para limite *indefinido* e *indefinido*, mas limitado.

Exemplo 2.1 Calcule os seguintes limites:

a) $\lim_{n\to+\infty} \frac{n}{n+1}$; b) $\lim_{x\to 0} \frac{\text{sen}(x)}{x}$; c) $\lim_{x\to+\infty} \cos(x)$.

Maxima

```
(%i143)limit(n/(n+1),n,inf);

(%o124)  1
(%i144)limit(sin(x)/x,x,0);
```

```
(%o125)  1
(%i145)limit(cos(x),x,inf);
(%o126)  ind
```

Exemplo 2.2 Calcule os seguintes somatórios:

a) $\sum_{i=1}^{10}(i^3+1)$; b) $\sum_{i=1}^{n} i^2$; c) $\sum_{j=1}^{m}\sum_{i=1}^{n}(i+j)$.

Maxima

```
(%i146)sum(i^3+1,i,1,10);
(%o127)  3035
(%i147)sum(i^2,i,1,n)
(%o128)  sum(i^2,i,1,n)
(%i148)sum(sum(i+j,i,1,n),j,1,m);
(%o129)  sum(sum(i+j,i,1,n),j,1,m)
```

Para obtermos o valor simbólico dos somatório (quando houver) com índice máximo não numerico, usamos o comando **simpsum** para simplificar o resultado.

Maxima

```
(%i149)sum(i^2,i,1,n), simpsum;
(%o130)  (2*n^3+3*n^2+n)/6
(%i150)sum(sum(i+j,i,1,n),j,1,m), simpsum;
(%o131)  m*n^2/2+(m^2*n+m*n)/2+m*n/2
```

2.2 Derivação e integração

Para calcularmos a derivada e a integral de uma função f usamos os seguintes comandos:

diff(f(x),x)	-	$f'(x)$
diff(f(x),x,n)	-	$f^{(n)}(x)$
diff(f(x,y),x,n,y,m)	-	$\frac{\partial^n}{\partial x^n}\left(\frac{\partial^m f}{\partial y^m}\right)$

integrate(f(x),x,a,b)	-	$\int_a^b f(x)\,dx$
integrate(**integrate**(f(x),x,a,b),y,c,d)	-	$\int_a^b \int_c^d f(x,y)\,dx\,dy$

Exemplo 2.3 (Cálculo direto de derivadas) Calcule as seguintes derivadas:

a) $f(x) = \sqrt{x+1}$; $f'(x)$; b) $f(x,y) = \text{sen}(xy^2)$; $\frac{\partial^2 f}{\partial y \partial x}$.

Maxima

```
(%i151)assume(x+1>0)
(%o132)  [x>-1]
(%i152)diff(sqrt(x+1),x);
(%o133)  1/(2*sqrt(x+1))
(%i153)diff(sin(x*y^2),y,1,x,1);
(%o134)  2*y*cos(x*y^2)-2*x*y^3*sin(x*y^2)
```

Podemos definir a função derivada usando o comando **define** ou a dupla aspas, como mostra o seguinte exemplo:

Exemplo 2.4 (Função derivada) Calcule a derivada da função $f(x) = \cos(x^2) + e^{-x^2}$ e o valor desta derivada em $x = 1$.

Solução:

Maxima

```
(%i154)f(x):=cos(x^2)+exp(-x^2)$
(%i155)define(g(x),diff(f(x),x));
```

(%o135) $g(x) := -2x\,\mathrm{sen}(x^2) - 2x\,\%e^{-x^2}$

```
(%i156)diff(f(x),x)$
(%i157)df(x):=''(%)
(%i158)df(1)
```

(%o136) $-2\sin(1) - 2\,\%e^{-1}$

```
(%i159)g(1)
```

(%o137) $-2\sin(1) - 2\,\%e^{-1}$

Podemos calcular o valor da derivada sem precisarmos definir a função derivada:

Maxima

```
(%i160)''(diff(f(x),x)), x=1;
```

(%o138) $-2\sin(1) - 2\,\%e^{-1}$

```
(%i161)ev(%,x=1);
```

(%o139) $-2\sin(1) - 2\,\%e^{-1}$

Operadores apóstrofo e duplo apóstrofo

Vamos destacar o uso dos operadores **apóstrofo '** e **duplo apóstrofo "**. O operador apóstrofo aplicado a um símbolo não efetua as operações, o operador duplo apóstrofo efetua as operações. Veja os seguintes exemplos:

Maxima

```
(%i162)A:200$
(%i163)A^2
```

(%o140) 40000

```
(%i164)'A^2
```

(%o141) A^2

```
(%i165)''A^2
```

(%o142) 40000

```
(%i166)'(sqrt(A) + 2*A)
```

(%o143) $2A + \sqrt{A}$

```
(%i167)''%
```

(%o144) $52^{3/2} + 400$

```
(%i168)'diff(f(x),x) = ''(diff(f(x),x));
```

(%o145) $\frac{d}{dx}\left(\cos(x^2) + \%e^{-x^2}\right) = -2x\,\mathrm{sen}(x^2) - 2x\,\%e^{-x^2}$

Exemplo 2.5 (Cálculo direto de integrais) Calcule as seguintes integrais:

a) $\int_0^{\pi/3} \mathrm{sen}^4(x)\cos(x)\,dx$; b) $\int \frac{1}{x\ln(x)^2}$; c) $\int_0^x s^3\,ds$.

Solução:

Maxima

```
(%i169)integrate(cos(x)*sin(x)^4,x,0, %pi/3);
```

(%o146) $\frac{3^{5/2}}{360}$

```
(%i170)integrate(1/(x*log(x)^2),x);
```

(%o147) $-\frac{1}{\log(x)}$

```
(%i171)integrate(s^3,s,0,x);
```

(%o148) $\frac{x^4}{4}$

2.2.1 Cálculo de uma variável com o Maxima

Exemplo 2.6 (Reta tangente e reta normal) Calcule a reta tangente e a reta normal à função $f(x) = x^2$ no ponto $(1,1)$. Desenhe a função, as retas e o ponto no mesmo sistema cartesiano.

Solução: Para calcularmos a reta tangente vamos derivar a função e obter o coeficiente angular "m" da tangente; o da normal sabendo que é dado por $m_1 = -1/m$. Portanto, com o Maxima tem-se:

Maxima

```
(%i172)diff(x^2,x);
```

(%o149) $2x$

```
(%i173)df(x):=''%;
```

(%o150) $\mathrm{df}(x) := 2x$

```
(%i174)m:df(1);
```

(%o151) 2

```
(%i175)reta1:1+m*(x-1);
```

(%o152) $2(x-1)+1$

```
(%i176)retatg(x):=''%;
```

(%o153) $\mathrm{retatg}(x) := 2(x-1)+1$

```
(%i177)reta2:1-(1/m)*(x-1);
```

(%o154) $1-\frac{x-1}{2}$

```
(%i178)retanormal(x):=''%;
```

(%o155) $\mathrm{retanormal}(x) := 1-(x-1)/2$

Desenhando no mesmo sistema cartesiano:

Maxima

```
(%i179)load(draw)$
(%i180)grfun:explicit(x^2,x,-1.4,2.8);
```

(%o156) $\mathrm{explicit}(x^2, x, -1.4, 2.8)$

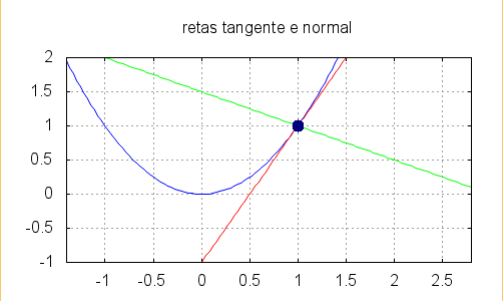
```
(%i181)grtg:explicit(retatg(x),x,-1.4,2.8);
```

(%o157) $\text{explicit}(2(x-1)+1, x, -1.4, 2.8)$

```
(%i182)grnormal:explicit(retanormal(x),x,-1.4,2.8);
```

(%o158) $\text{explicit}\left(1-\frac{x-1}{2}, x, -1.4, 2.8\right)$

```
(%i183)draw2d(yrange=[-1,2],color=blue,grfun,color=red,grtg,
       color=green,grnormal,color=navy,point_size=2,point_type=7,
       points([1],[1]),grid=true,title="retas tangente e normal");
```



(%o159)

Exemplo 2.7 (Integral definida) Usando a definição, calcule as seguintes integrais:

a) $\int_0^1 x^2\,dx$; b) $\int_{-1}^0 (x^2-x^3)\,dx$; c) $\int_0^1 \sqrt{x}\,dx$.

Solução: (a) Devemos calcular o limite:

$$\lim_{n\to\infty}\sum_{i=1}^{n}\xi_i^2\,(x_i-x_{i-1}).$$

Como a função $f(x)=x^2$ é contínua em $[0,1]$, podemos considerar a partição uniforme $x_i-x_{i-1}=\frac{1}{n}$ e os pontos $\xi_i=x_i=\frac{i}{n}$. Assim, o limite acima torna-se

$$\lim_{n\to\infty}\sum_{i=1}^{n}\left(\frac{i}{n}\right)^2\frac{1}{n}\Rightarrow\lim_{n\to\infty}\frac{1}{n^3}\left(\sum_{i=1}^{n}i^2\right).$$

Agora, vamos calcular soma dos quadrados dos n primeiros naturais $\sum_{i=1}^{n}i^2$.

Maxima

```
(%i184)sum(i^2,i,1,n), simpsum;
```

(%o160) $\dfrac{2n^3+3n^2+n}{6}$.

Calculando o limite:

$$\lim_{n\to\infty}\frac{1}{n^3}\left(\frac{2n^3+3n^2+n}{6}\right)=\lim_{n\to\infty}\left(\frac{1}{3}+\frac{1}{2n}+\frac{1}{6n^2}\right)=\frac{1}{3}.$$

Podemos calcular mais diretamente combinando todos os comandos do seguinte modo:

Maxima

```
(%i185)limit(sum(i^2/n^3,i,1,n),n,inf), simpsum;
```

(%o161) $\dfrac{1}{3}$

Solução: (b) Devemos calcular o limite:

$$\lim_{n\to\infty}\sum_{i=1}^{n}(\xi_i^2-\xi_i^3)(x_i-x_{i-1}).$$

Como a função $f(x)=x^2-x^3$ é contínua em $[-1,0]$, podemos considerar a partição uniforme $x_i-x_{i-1}=\dfrac{1}{n}$ e os pontos $\xi_i=x_i=-1+\dfrac{i}{n}$. Assim, o limite acima torna-se

$$\begin{aligned}
&\lim_{n\to\infty}\sum_{i=1}^{n}\left[\left(-1+\frac{i}{n}\right)^2-\left(-1+\frac{i}{n}\right)^3\right]\frac{1}{n}=\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left(-1+\frac{i}{n}\right)^2\left[1-\left(-1+\frac{i}{n}\right)\right]\\
&=\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left(2-\frac{5i}{n}+\frac{4i^2}{n^2}-\frac{i^3}{n^3}\right)\\
&=\lim_{n\to\infty}\left(\frac{2}{n}\sum_{i=1}^{n}1-\frac{5}{n^2}\sum_{i=1}^{n}i+\frac{4}{n^3}\sum_{i=1}^{n}i^2-\frac{1}{n^4}\sum_{i=1}^{n}i^3\right)
\end{aligned}\tag{2.1}$$

Sabemos que

$$\sum_{i=1}^{n}1=n,\qquad\sum_{i=1}^{n}i^2=\frac{2n^3+3n^2+n}{6}.$$

Resta calcularmos soma dos n primeiros naturais $\sum_{i=1}^{n} i$ e a soma dos cubos dos n primeiros naturais $\sum_{i=1}^{n} i^3$:

Maxima

```
(%i186)sum(i,i,1,n), simpsum;
```

$$(\%o162)\quad \frac{n^2+n}{2}$$

```
(%i187)sum(i^3,i,1,n), simpsum;
```

$$(\%o163)\quad \frac{n^4+2n^3+n^2}{4}.$$

Portanto, o limite (2.1) torna-se

$$\begin{aligned}
&\lim_{n\to\infty}\left(\frac{2}{n}\sum_{i=1}^{n}1-\frac{5}{n^2}\sum_{i=1}^{n}i+\frac{4}{n^3}\sum_{i=1}^{n}i^2-\frac{1}{n^4}\sum_{i=1}^{n}i^3\right)\\
&=\lim_{n\to\infty}\left[2-\frac{5}{2}\left(1+\frac{1}{n}\right)+\frac{2}{3}\left(2+\frac{3}{n}+\frac{1}{n^2}\right)-\frac{1}{4}\left(1+\frac{2}{n}+\frac{1}{n^2}\right)\right]\\
&=\lim_{n\to\infty}\left[2-\frac{5}{2}+\frac{2}{3}-\frac{1}{4}+\frac{5}{2n}+\frac{6}{3n}+\frac{2}{3n^2}-\frac{2}{4n}+\frac{1}{4n^2}\right]\\
&=\left(2-\frac{5}{2}+\frac{2}{3}-\frac{1}{4}\right)=\frac{7}{12}.
\end{aligned}$$

Podemos calcular mais diretamente combinando todos os comandos do seguinte modo:

Maxima

```
(%i188)limit(sum(((-1+i/n)^2-(-1+i/n)^3)*1/n,i,1,n),n,inf),
       simpsum;
```

$$(\%o164)\quad \frac{7}{12}$$

Solução: (c) Devemos calcular o limite:

$$\lim_{n\to\infty}\sum_{i=1}^{n}\sqrt{\xi_i}\,(x_i-x_{i-1}).$$

Como a função $f(x)=\sqrt{x}$ é contínua em $[0,1]$ podemos considerar a partição uniforme $x_i-x_{i-1}=\frac{1}{n}$ e os pontos $\xi_i=x_i=\frac{i}{n}$. Porém, esta escolha não é a mais

adequada, pois não temos uma fórmula para a série

$$\sum_{i=1}^{n} \frac{\sqrt{i}}{n\sqrt{n}}$$

em termos de "n". Para evitar este problema, vamos escolher a partição não uniforme $x_i - x_{i-1} = \frac{i^2}{n^2} - \frac{(i-1)^2}{n^2} = \frac{2i-1}{n^2}$ e os pontos $\xi_i = x_i = \frac{i^2}{n^2}$.

Assim, o limite acima torna-se

$$\lim_{n\to\infty} \sum_{i=1}^{n} \sqrt{\frac{i^2}{n^2}} \left(\frac{2i-1}{n^2}\right) = \lim_{n\to\infty} \sum_{i=1}^{n} \frac{i}{n} \left(\frac{2i-1}{n^2}\right)$$
$$= \lim_{n\to\infty} \frac{1}{n^3} \sum_{i=1}^{n} (2i^2 - 1) = \lim_{n\to\infty} \frac{2}{n^3} \sum_{i=1}^{n} i^2 - \lim_{n\to\infty} \frac{1}{n^3} \sum_{i=1}^{n} 1$$

Sabemos que

$$\sum_{i=1}^{n} 1 = n, \qquad \sum_{i=1}^{n} i^2 = \frac{2n^3 + 3n^2 + n}{6}.$$

Logo,

$$\lim_{n\to\infty} \sum_{i=1}^{n} \sqrt{\xi_i}\,(x_i - x_{i-1}) = \lim_{n\to\infty} \frac{2}{n^3} \left(\frac{2n^3 + 3n^2 + n}{6}\right) - \lim_{n\to\infty} \frac{1}{n^2}$$
$$= \lim_{n\to\infty} \left(\frac{2}{3} + \frac{1}{n} + \frac{1}{3n^2}\right) = \frac{2}{3}.$$

Podemos calcular mais diretamente combinando todos os comandos do seguinte modo:

Maxima

```
(%i189)limit(sum(sqrt(i^2/n^2)*((2*i-1)/n^2),i,1,n),n,inf),
       simpsum;
(%o165)  2/3
```

Exemplo 2.8 Mostre que

$$\int_a^b x\,dx = \frac{b^2 - a^2}{2}.$$

Solução: Usando a definição, devemos calcular o limite:

$$\lim_{n\to\infty} \sum_{i=1}^{n} \xi_i\,(x_i - x_{i-1}).$$

Como a função $f(x) = x$ é contínua em $[a,b]$ podemos considerar a partição uniforme $x_i - x_{i-1} = \dfrac{b-a}{n}$ e os pontos $\xi_i = x_i = a + \dfrac{(b-a)i}{n}$. Assim, o limite acima torna-se

$$\lim_{n\to\infty} \sum_{i=1}^{n} \xi_i \, (x_i - x_{i-1}) = \lim_{n\to\infty} \sum_{i=1}^{n} \left(a + \frac{(b-a)i}{n}\right)\left(\frac{b-a}{n}\right).$$

Calculando com o `Maxima` obtemos

Maxima

```
(%i190)limit(sum((a+(b-a)*i/n)*((b-a)/n),i,1,n),n,inf),
       simpsum;
```

(%o166) $\dfrac{b^2 - a^2}{2}$

Outra solução para este exemplo é usar diretamente o comando **integrate**:

Maxima

```
(%i191)integrate(x,x,a,b);
```

(%o167) $\dfrac{b^2}{2} - \dfrac{a^2}{2}$

Exemplo 2.9 (Integral indefinida) Desenhe o gráfico da função

$$F(x) = \int_0^x \cos(t)\, dt.$$

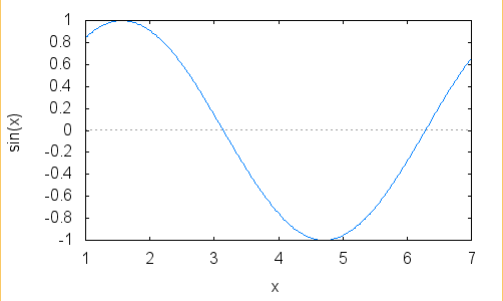
Você reconhece este gráfico? Explique.

Solução:

Maxima

```
(%i192)F(x):= integrate(cos(t),t,0,x)
(%i193)assume(x>0);
```

(%o168) $[x > 0]$



(%o169)

O gráfico é uma senóide, ou seja, é o gráfico da função $f(x) = \text{sen}(x)$.

Exemplo 2.10 (Integral indefinida) Usando o Teorema Fundamental do Cálculo, calcule a derivada das seguintes integrais indefinidas:

a) $F(x) = \int_0^x \sqrt{t^2+1}\,dt$; b) $F(x) = \int_{\pi/2}^{x^3} \cos(t)\,dt$.

Solução: (a)

Maxima

```
(%i194)assume(x>0)
```

(%o170) [x>0]

```
(%i195)assume(x>0)
(%i196)ratsimp(diff(integrate(sqrt(t^2+1),t,0,x),x));
```

(%o171) $\sqrt{x^2+1}$

Solução: (b)

Maxima

```
(%i197)diff(integrate(cos(t),t,%pi/2,x^3),x);
```

(%o172) $3x^2 cos(x^3)$

Exemplo 2.11 (Cálculo de área) Calcule a área limitada pelo gráfico das funções $f(x) = 2 - x^2$ e $g(x) = x$. Desenhe a região.

Solução: Primeiro calcularemos os pontos de intersecção entre as funções:

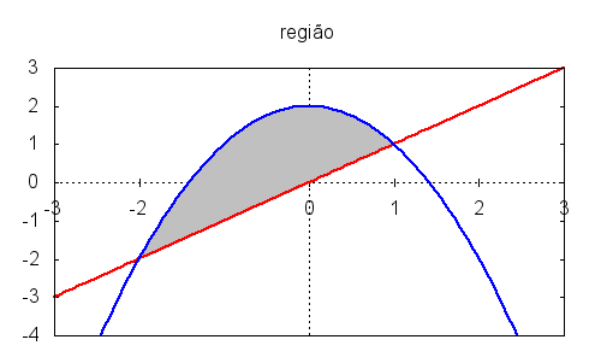
Maxima

```
(%i198)f1(x):=x$ f2(x):=2-x^2$
       [x1,x2]: map('rhs, solve(f1(x)=f2(x)));
(%o173)  [1,-2]
```

Agora, desenharemos a região de integração:

Maxima

```
(%i199)wxdraw2d(title = "região",fill_color = grey,
       filled_func = f2,explicit(f1,x,x2,x1),filled_func = false,
       xaxis = true,xtics_axis = true,yaxis = true,line_width = 2,
       key = "",color=red,explicit(f1,x,-3,3),key = "",color = blue,
       explicit(f2,x,-3,3),yrange=[-4,3]);
```




```
(%o174)
```

Como $g(x) \leq f(x)$ em $[-2,1]$ temos que a área é $\int_{-2}^{1} f(x) - g(x)\,dx$. Calculando com o `Maxima` obtemos

Maxima

```
(%i200)integrate(f2-f1,x,-2,1);
(%o175)  9/2
```

Exemplo 2.12 (Cálculo de área) Calcule a área limitada pelo gráfico das funções $f(x) = \text{sen}(x)$ e $g(x) = \cos(x)$. Desenhe a região.

Solução: O comando **solve** do aplicativo Maxima não resolve equações do tipo $\text{sen}(x) = \cos(x)$. Para $x \in [0, 2\pi]$ temos que o seno e o cosseno são iguais para $x = \frac{\pi}{4}$ e $x = \frac{5\pi}{4}$. Deste modo, a região de integração é dada por

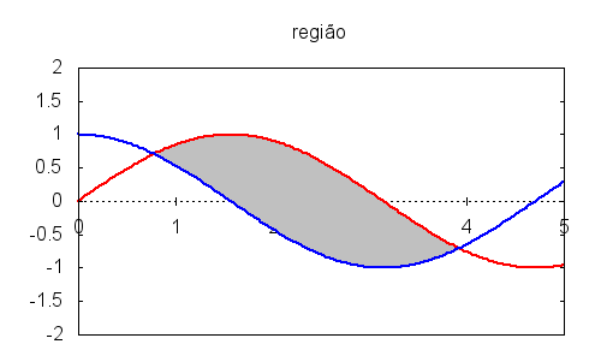
Maxima

```
(%i201)f1(x):=sin(x)$ f2(x):=cos(x)$
       [x1,x2]:[%pi/4,5*%pi/4];
```

(%o176) $[\frac{\pi}{4}, \frac{5\pi}{4}]$

Maxima

```
(%i202)wxdraw2d(title = "região",fill_color = grey,filled_func = f2,
       explicit(f1,x,x1,x2),filled_func = false,xaxis = true,
       xtics_axis = true,yaxis = true,line_width = 2,key = "",
       color=red,
       explicit(f1,x,0,5),key="",color = blue,explicit(f2,x,0,5),
       yrange=[-2,2] );
```



(%o177)

Como $\text{sen}(x) \geq \cos(x)$ para $x \in [\pi/4, 5\pi/4]$ temos que a área é $\int_{\pi/4}^{5\pi/4} (\text{sen}(x) - \cos(x))\,dx$. Calculando com o Maxima tem-se

Maxima

```
(%i203)integrate(f1-f2,x,%pi/4,5*%pi/4);
```

(%o178) $2^{3/2}$

Exemplo 2.13 (Cálculo de área) Calcule a área limitada pelo gráfico das funções $f(x) = 3x^3 - x^2 - 10x$ e $g(x) = -x^2 + 2x$. Desenhe a região.

Solução: Primeiro calcularemos os pontos de intersecção entre as funções:

Maxima

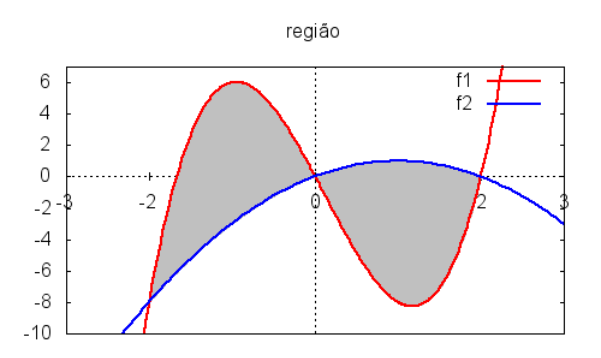
```
(%i204)f1(x):=3*x^3-x^2-10*x$ f2(x):=-x^2+2*x$
       [x1,x2,x3]: map('rhs, solve(f1(x)=f2(x)));
```

(%o179) $[-2, 2, 0]$

Agora, desenharemos a região de integração:

Maxima

```
(%i205)wxdraw2d(title = "região",fill_color = grey,
       filled_func = f2,explicit(f1,x,x1,x2),
       filled_func = false,xaxis = true,xtics_axis = true,
       yaxis = true,line_width = 2,key = "f1",color=red,
       explicit(f1,x,-3,3),  key="f2",color = blue,
       explicit(f2,x,-3,3),yrange=[-10,7] );
```



(%o180)

Como $g(x) \leq f(x)$ em $[-2,0]$ e $f(x) \leq g(x)$ em $[0,2]$ temos que a área é

$$\int_{-2}^{0} f(x) - g(x)\,dx + \int_{0}^{2} g(x) - f(x)\,dx.$$

Calculando com o Maxima temos

Maxima

```
(%i206)integrate(f1-f2,x,-2,0)+integrate(f2-f1,x,0,2);
(%o181)  24
```

Observação 2.1 Observe que no exemplo 2.13 a integração de -2 a 2 daria um resultado incorreto. De fato,

Maxima

```
(%i207)integrate(f1-f2,x,-2,2);
(%o182)  0
```

Exemplo 2.14 (Cálculo de área) Sejam $a > 0$ e $b > 0$. Mostre que a área limitada pela elipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

é $\pi\, a\, b$.

Solução: Calculando com o Maxima obtemos:

Maxima

```
(%i208)f1:b/a*sqrt(a^2-x^2)$f2:0$
       assume(a>0,b>0)$ assume(-a<x and x<a)$
       2*integrate(f1-f2,x,-a,a);
(%o183)  π a b
```

Exemplo 2.15 (Cálculo de área) Calcule a área limitada pela da curva abaixo e desenhe a região.

$$y^2 = x^2(1-x^2).$$

Solução: Primeiro calcularemos os pontos de intersecção entre as funções $y_1 = x\sqrt{1-x^2}$ e $y_2 = -x\sqrt{1-x^2}$

Maxima

```
(%i209)assume(-1<x and x<1);
```

(%o184) $[x > -1, x < 1]$

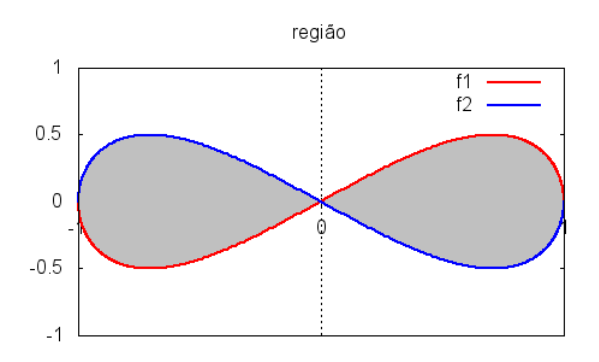
```
(%i210)f1:x*sqrt(1-x^2)$f2:-x*sqrt(1-x^2)$
       [x1,x2,x3]: map('rhs, solve(f1=f2));
```

(%o185) $[-1, 1, 0]$

Agora, desenharemos a região de integração:

Maxima

```
(%i211)wxdraw2d(title = "região",fill_color = grey,
       filled_func = f2,explicit(f1,x,-1,1),
       filled_func = false,xaxis = true,xtics_axis = true,
       yaxis = true,line_width = 2,key = "f1",color=red,
       explicit(f1,x,-1,1),  key="f2",color = blue,
       explicit(f2,x,-1,1),yrange=[-1,1]);
```



(%o186)

Como $y_2 \leq y_1$ em $[-1,0]$ e $y_1 - y_2 \leq g(x)$ em $[0,1]$ temos que a área é

$$\int_{-1}^{0} y_2 - y_1\, dx + \int_{0}^{1} y_1 - y_2\, dx.$$

Calculando com o `Maxima` temos

Maxima

```
(%i212)integrate(f2-f1,x,-1,0)+integrate(f1-f2,x,0,1);
```

$$(\%o187) \quad \frac{4}{3}$$

Nos seguintes exemplos usaremos o comando **integrate** para resolver integrais indicando qual a técnica de integração usada.

Exemplo 2.16 (Integração por partes) Calcule as seguintes integrais:

(a) $\int x^2 \ln(x)\, dx$; (b) $\int_0^1 \text{arcsen(x)}\, \text{dx}$ (c) $\int x^2 \text{sen}(x)\, dx$.

Solução: Usando o comando **integrate** obtemos

Maxima

```
(%i213)integrate(x^2*log(x),x);
```

$$(\%o188) \quad \frac{x^3 \log(x)}{3} - \frac{x^3}{9}$$

```
(%i214)integrate(asin(x),x,0,1);
```

$$(\%o189) \quad \frac{\pi}{2} - 1$$

```
(%i215)integrate(x^2*sin(x),x);
```

$$(\%o190) \quad 2x\sin(x) + (2 - x^2)\cos(x)$$

Exemplo 2.17 (Integrais trigonométricas) Calcule as seguintes integrais:

(a) $\int \text{sen}(x)^3 \cos(x)^4\, dx$; (b) $\int_{\pi/6}^{\pi/3} \frac{\cos(x)^3}{\sqrt{\text{sen}(x)}}\, dx$ (c) $\int \cos(x)^4\, dx$.

Solução: Usando o comando **integrate** obtemos

Maxima

```
(%i216)integrate(sin(x)^3*cos(x)^4,x);
```

$$(\%o191)\quad \frac{5\cos(x)^7 - 7\cos(x)^5}{35}$$

```
(%i217)integrate(cos(x)^3/sqrt(sin(x)),x,%\pi/6,%\pi/3);
```

$$(\%o192)\quad \frac{17\,3^{1/4}}{5\,2^{3/2}} - \frac{19}{5\,2^{3/2}}$$

```
(%i218)integrate(cos(x)^4,x),trigreduce;
```

$$(\%o193)\quad \frac{\sin(4x) + 8\sin(2x) + 12x}{32}$$

Exemplo 2.18 (Substituição trigonométrica) Calcule as seguintes integrais:

$$\text{(a)} \int \frac{dx}{x^2\sqrt{9-x^2}}; \qquad \text{(b)} \int_{\sqrt{3}}^{2} \frac{\sqrt{x^2-3}}{x}\,dx \qquad \text{(c)} \int \frac{dx}{(x^2+1)^{3/2}}.$$

Solução: Usando o comando **integrate** obtemos

Maxima

```
(%i219)integrate(1/(x^2*sqrt(9-x^2)),x);
```

$$(\%o194)\quad -\frac{\sqrt{9-x^2}}{9x}$$

```
(%i220)integrate(sqrt(x^2-3)/x,x,sqrt(3),2),ratsimp;
```

$$(\%o195)\quad -\frac{\sqrt{3}\pi - 6}{6}$$

```
(%i221)integrate(1/(x^2+1)^(3/2),x);
```

$$(\%o196)\quad \frac{x}{\sqrt{x^2+1}}$$

Exemplo 2.19 (Frações parciais) Calcule as seguintes integrais:

$$\text{(a)} \int \frac{dx}{x^2-5x+6}; \qquad \text{(b)} \int \frac{5x^2+20x+6}{x^3+2x^2+x}\,dx \qquad \text{(c)} \int \frac{8x^3+13x}{(x^2+2)^2}.$$

Solução: Usando o comando **integrate** obtemos

Maxima

```
(%i222)integrate(1/(x^2-5*x+6),x);
```

$$(\%o197)\quad -\log(x-2) + \log(x-3)$$

```
(%i223)integrate((5*x^2+20*x+6)/(x^3+2*x^2+x),x);
```

$$(\%o198)\quad -\log(x+1)+6\log(x)-\frac{9}{x+1}$$

```
(%i224)integrate((5*x^3+13*x)/(x^2+2)^2,x);
```

$$(\%o199)\quad \frac{5\log(x^2+2)}{2}-\frac{3}{2x^2+4}$$

2.2.2 Curvas com o Maxima

Exemplo 2.20 (Curvas) Seja $\vec{r}(t) = 4cos(t)\vec{i} + 4\text{sen}(t)\vec{j} + t\vec{k}$. Calcule:

(a) $\lim_{t\to\pi}\vec{r}(t)$; (b) $\frac{d\vec{r}}{dt}$; (c) $\int \vec{r}(t)\,dt$; (d) $\int_0^{\pi}\vec{r}(t)\,dt$.

Desenhe o traço da curva.

Soluções:

Maxima

```
(%i225)r(t):=[4*cos(t),4*sin(t),t];
```

$$(\%o200)\quad r(t) := [4\cos(t), 4\sin(t), t]$$

```
(%i226)limit(r(t),t,%pi);
```

$$(\%o201)\quad [-4, 0, \pi]$$

```
(%i227)diff(r(t),t);
```

$$(\%o202)\quad [-4\sin(t), 4\cos(t), 1]$$

```
(%i228)integrate(r(t),t);
```

$$(\%o203)\quad [4\sin(t), -4\cos(t), \frac{t^2}{2}]$$

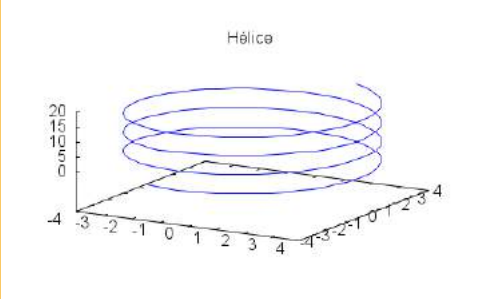
```
(%i229)integrate(r(t),t,0,%pi);
```

$$(\%o204)\quad [0, 8, \frac{\pi^2}{2}]$$

Desenhando a curva:

Maxima

```
(%i230)load(draw)$
(%i231)wxdraw3d(nticks=200,parametric(4*cos(t),4*sin(t),t,t,-2,20),
       title="Hélice");
```



(%o205)

Exemplo 2.21 Considere a hélice $\vec{r}(t) = 2cos(t)\vec{i} + 2\text{sen}(t)\vec{j} + t\vec{k}$. Calcule o vetor $\vec{T}$ tangente unitário e o vetor normal unitário $\vec{N}$ a $\vec{r}(t)$. Desenhe a curva e os vetores.

Solução: O vetor $\vec{T}$ tangente unitário é definido como $\vec{T} = \dfrac{r'(t)}{\|r'(t)\|}$ e vetor normal unitário $\vec{N}$ é definido como $\vec{T} = \dfrac{T'(t)}{\|T'(t)\|}$.

Calculando o vetor tangente unitário:

Maxima

```
(%i232)r(t):=[2*cos(t),2*sin(t),t];
```

(%o206) $\text{r}(t) := [2\cos(t), 2\sin(t), t]$

```
(%i233)diff(r(t),t);
```

(%o207) $[-2\sin(t), 2\cos(t), 1]$

```
(%i234)define(dr(t),%);
```

(%o208) $\text{dr}(t) := [-2\sin(t), 2\cos(t), 1]$

```
(%i235)normadr(t):=sqrt(dr(t)[1]^2+dr(t)[2]^2+dr(t)[3]^2);
```

(%o209) $\text{normadr}(t) := \sqrt{(\text{dr}(t))_1^2 + (\text{dr}(t)_2^2 + (\text{dr}(t))_3^2}$

```
(%i236)normadr(t);
```

(%o210) $\sqrt{4\sin(t)^2 + 4\cos(t)^2 + 1}$

```
(%i237)Tr(t):=trigsimp(dr(t)/normadr(t));
```

(%o211) $\text{Tr}(t) := \text{trigsimp}\left(\dfrac{\text{dr}(t)}{\text{normadr}(t)}\right)$

```
(%i238)Tr(t);
```

(%o212) $[-\frac{2\sin(t)}{\sqrt{5}}, \frac{2\cos(t)}{\sqrt{5}}, \frac{1}{\sqrt{5}}]$

Calculando o vetor normal unitário:

Maxima

```
(%i239)diff(Tr(t),t);
```

(%o213) $[-\frac{2\cos(t)}{\sqrt{5}}, -\frac{2\sin(t)}{\sqrt{5}}, 0]$

```
(%i240)define(dTr(t),%);
```

(%o214) $\mathrm{dTr}(t) := [-\frac{2\cos(t)}{\sqrt{5}}, -\frac{2\sin(t)}{\sqrt{5}}, 0]$

```
(%i241)normadTr(t):=sqrt(dTr(t)[1]^2+dTr(t)[2]^2+dTr(t)[3]^2);
```

(%o215) $\mathrm{normadTr}(t) := \sqrt{(\mathrm{dTr}(t))_1^2 + (\mathrm{dTr}(t)_2^2 + (\mathrm{dTr}(t))_3^2}$

```
(%i242)normadTr(t);
```

(%o216) $\sqrt{\frac{4\sin(t)^2}{5} + \frac{4\cos(t)^2}{5}}$

```
(%i243)Nr(t):=trigsimp(dTr(t)/normadTr(t));
```

(%o217) $\mathrm{Nr}(t) := \mathrm{trigsimp}\left(\frac{\mathrm{dTr}(t)}{\mathrm{normadTr}(t)}\right)$

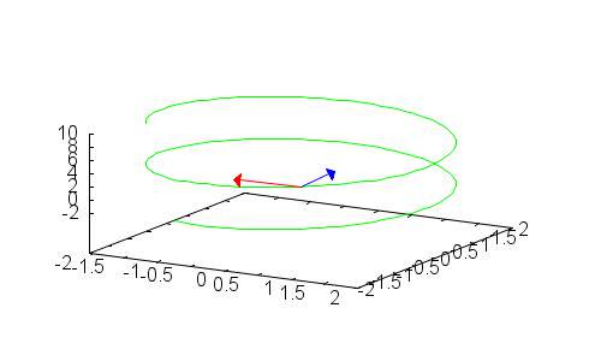
```
(%i244)Nr(t);
```

(%o218) $[-\cos(t), -\sin(t), 0]$

Desenhando a curva e os vetores:

Maxima

```
(%i245)load(draw)$
(%i246)grcurvar:parametric(r(t)[1],r(t)[2],r(t)[3],t,-2,10)$
(%i247)grTr:vector([0,0,0],[Tr(0)[1],Tr(0)[2],Tr(0)[3]])$
(%i248)grNr:vector([0,0,0],[Nr(0)[1],Nr(0)[2],Nr(0)[3]])$
(%i249)wxdraw3d(nticks=200,color=green,grcurvar,head_length=0.1,
       line_width=0.2,color=blue,grTr,color=red,grNr);
```



(%o219)

Exemplo 2.22 Calcule o comprimento de arco da hélice

$$\vec{r}(t) = b\,cos(t)\,\vec{i} + b\,\mathrm{sen}(t)\,\vec{j} + \sqrt{1-b^2}\,t\,\vec{k}$$

Solução: Sabemos que o comprimento de arco é dado por $s = \int_0^{2\pi} \|r'(t)\|\,dt$.

Maxima

```
(%i250)r(t):=[b*cos(t),b*sin(t),sqrt(1-b^2)*t];
```

(%o220) $\mathrm{r}(t) := [2\cos(t), 2\sin(t), \sqrt{1-\mathrm{b}^2}\,t]$

```
(%i251)diff(r(t),t);
```

(%o221) $[-2\sin(t), 2\cos(t), \sqrt{1-\mathrm{b}^2}]$

```
(%i252)define(dr(t),%);
```

(%o222) $\mathrm{dr}(t) := [-2\sin(t), 2\cos(t), \sqrt{1-\mathrm{b}^2}]$

```
(%i253)normadr(t):=sqrt(dr(t)[1]^2+dr(t)[2]^2+dr(t)[3]^2);
```

(%o223) $\mathrm{normadr}(t) := \sqrt{(\mathrm{dr}(t))_1^2 + (\mathrm{dr}(t)_2^2 + (\mathrm{dr}(t))_3^2}$

```
(%i254)normadr(t);
```

(%o224) 1

```
(%i255)s:integrate(normadr(t),t,0,2*%pi);
```

(%o225) 2π

2.2.3 Cálculo de várias variáveis com o Maxima

Exemplo 2.23 (Regra da cadeia) Seja $f(x,y) = x^2y - y^2$ com $x = \text{sen}(t)$ e $y = e^t$. Calcule $\frac{df}{dt}(0)$.

Solução:

Maxima

```
(%i256)f(x,y):=x^2*y -y^2;
```

(%o226) $f(x,y) := x^2y - y^2$

```
(%i257)[x,y]:[sin(t),exp(t)];
```

(%o227) $[\sin(t), \%e^t]$

```
(%i258)diff(f(x,y),t);
```

(%o228) $\%e^t \,\text{sen}(t)^2 + 2\,\%e^t \cos(t)\,\text{sen}(t) - 2\,\%e^{2t}$

```
(%i259)define(df(t),%);
```

(%o229) $df(t) := \%e^t \,\text{sen}(t)^2 + 2\,\%e^t \cos(t)\,\text{sen}(t) - 2\,\%e^{2t}$

```
(%i260)df(0)
```

(%o230) -2

Exemplo 2.24 (Derivação implícita) Calcule $\frac{dy}{dx}$ para $y(x)$ dada implicitamente em $y^3 + y^2 - 5y - x^2 + 4 = 0$.

Solução:

Maxima

```
(%i261)F:y^3+y^2-5*y-x^2+4;
```

(%o231) $F : y^3 + y^2 - 5y - x^2 + 4;$

```
(%i262)Fx:diff(F,x);
```

(%o232) $-2x$

```
(%i263)Fy:diff(F,y);
```

(%o233) $3y^2 + 2y - 5$

```
(%i264)dy:-Fx/Fy
```

(%o234) $$\frac{2x}{3y^2 + 2y - 5}$$

Exemplo 2.25 (Derivada direcional) Calcule a derivada direcional da função $f(x,y) = 3x^2 - 2y^2$ em $\left(-\frac{3}{4}, 0\right)$ na direção do vetor $\vec{v} = \frac{3}{4}\vec{i} + \vec{j}$

Solução: Sabemos que $D_u f(x,y) = \nabla f(x,y).\vec{u}$. Usaremos o produto escalar, representado pelo ponto **.**

Maxima

```
(%i265)v:[3/4,1];
```

(%o235) $[\frac{3}{4}, 1]$

```
(%i266)f(x,y):=3*x^2-2*y^2;
```

(%o236) $f(x,y) := 3x^2 - 2y^2$

```
(%i267)df:[diff(f(x,y),x),diff(f(x,y),y)];
```

(%o237) $[6x, -4y]$

```
(%i268)define(gradf(x,y),%);
```

(%o238) $gradf(x,y) := [6x, -4y]$

```
(%i269)(gradf(-3/4,0).v)/sqrt(v.v);
```

(%o239) $-\frac{27}{10}$

Exemplo 2.26 (Integral dupla) Usando a definição, calcule as seguinte integrais duplas:

(a) $\int_0^1\int_0^1 xy\,dx\,dy$; (b) $\int_{-1}^1\int_1^2 (3x-y)\,dx\,dy$; (c) $\int_{-1}^1\int_0^1 x^3y^2\,dx\,dy$.

Solução: (a) Devemos calcular o limite:

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i y_j (x_i - x_{i-1})(y_j - y_{j-1})\right).$$

Como a função $f(x,y) = xy$ é contínua no retângulo $[0,1]\times[0,1]$, podemos considerar a seguinte partição:

$$x_i = \frac{i}{n}, \quad y_j = \frac{j}{n}.$$

Assim, o limite acima torna-se

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{ij}{n^4}\right)\Rightarrow\lim_{n\to\infty}\frac{1}{n^4}\left(\sum_{i=1}^{n}i\right)\left(\sum_{j=1}^{n}j\right)$$

Agora, vamos calcular soma dos n primeiros naturais:

Maxima

```
(%i270)somax:sum(i,i,1,n), simpsum;
```

(%o240) $\dfrac{n^2+n}{2}$

Assim, devemos calcular o limite:

$$\lim_{n\to\infty}\frac{1}{n^4}\left(\frac{n^2+n}{2}\right)^2.$$

Maxima

```
(%i271)limit(((n^2+n)/2)^2*1/n^4,n,inf);
```

(%o241) $\dfrac{1}{4}$

Também podemos calcular diretamente:

Maxima

```
(%i272)limit(sum(sum(i*j/n^4,i,1,n),j,1,n),n,inf), simpsum;
```

(%o242) $\dfrac{1}{4}$

Solução: (b) Devemos calcular o limite:

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(3x_i-y_j)(x_i-x_{i-1})(y_j-y_{j-1})\right).$$

Como a função $f(x,y)=3x-y$ é contínua no retângulo $[1,2]\times[-1,1]$, podemos considerar a seguinte partição:

$$x_i=-1+\frac{2i}{n},\quad y_j=1+\frac{j}{n}.$$

Assim, devemos calcular o seguinte limite:

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(3\left(-1+\frac{2i}{n}\right)-1+\frac{j}{n}\right)\left(\frac{2}{n}\right)\left(\frac{1}{n}\right)\right)=\lim_{n\to\infty}\left(-\frac{8}{n^2}+\frac{12i}{n^3}+\frac{2j}{n^3}\right).$$

Calculando diretamente com o `Maxima` tem-se:

Maxima

```
(%i273)limit(sum(sum(-8/n^2+12*i/n^3-2*j/n^3,i,1,n),j,1,n),
       n,inf), simpsum;
(%o243)  -3
```

Solução: (c) Devemos calcular o limite:

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^3y_j^2)(x_i-x_{i-1})(y_j-y_{j-1})\right).$$

Como a função $f(x,y)=x^3y^2$ é contínua no retângulo $[-1,1]\times[0,1]$, podemos considerar a seguinte partição:

$$x_i=\frac{i}{n},\quad y_j=-1+\frac{2j}{n}.$$

Assim,

$$\begin{aligned}
&\lim_{n\to\infty}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{i^3}{n^3}\right)\left(-1+\frac{2j}{n}\right)^2\left(\frac{1}{n}\right)\left(\frac{2}{n}\right)\right)\\
&=\lim_{n\to\infty}\frac{2}{n^2}\left(\sum_{i=1}^{n}\left(\frac{i^3}{n^3}\right)\right)\left(\sum_{j=1}^{n}\left(-1+\frac{2j}{n}\right)^2\right)\\
&=\lim_{n\to\infty}\frac{2}{n^2}\left(\frac{1}{n^3}\sum_{i=1}^{n}i^3\right)\left(n-\frac{4}{n}\sum_{j=1}^{n}j+\frac{4}{n^2}\sum_{j=1}^{n}j^2\right)
\end{aligned}$$

Agora, vamos calcular soma dos quadrados e dos cubos dos n primeiros naturais

Maxima

```
(%i274)somaA:sum(i^2,i,1,n), simpsum;
(%o244)  (2n^3+3n^2+n)/6
(%i275)somaB:sum(i^3,i,1,n), simpsum;
```

(%o245) $\dfrac{n^4+2n^3+n^2}{4}$

Assim, devemos calcular o seguinte limite:

$$\lim_{n\to\infty}\frac{2}{n^2}\left(\frac{1}{n^3}\sum_{i=1}^{n}i^3\right)\left(n-\frac{4}{n}\sum_{j=1}^{n}j+\frac{4}{n^2}\sum_{j=1}^{n}j^2\right)$$
$$=\lim_{n\to\infty}\frac{2}{n^2}\left(\frac{1}{n^3}\left(\frac{n^4+2n^3+n^2}{4}\right)\right)\left(1-n+\frac{4}{n^2}\left(\frac{2n^3+3n^2+n}{6}\right)\right)$$

Maxima

```
(%i276)define(pA(n),%o202)$
(%i277)define(pB(n),%o203)$
(%i278)s(n):=(2*pB(n)/n^5)*(1-n+4*pA(n)/n^2)$
(%i279)limit(expand(s(n)),n,inf);
```

(%o246) $\dfrac{1}{6}$

Calculando diretamente com o `Maxima` tem-se:

Maxima

```
(%i280)limit(sum(sum((i/n)^3*(-1+2*j/n)^2*(2/n^2),i,1,n),
       j,1,n),n,inf), simpsum;
```

(%o247) $\dfrac{1}{6}$

Exemplo 2.27 (Cálculo de volume) Calcule o volume do sólido W limitado abaixo pelo retângulo $[0,1]\times[0,1]$ e acima pelo plano $x+y+z=2$ (veja figura 2.1).

Solução: O volume de W é dado pela integral dupla

$$V=\int_0^1\int_0^1 2-x-y\,dx\,dy.$$

Aplicando o Teorema de Fubini obtemos

Maxima

```
(%i281)integrate(2-x-y,x,0,1);
```

$$(\%o248) \quad -\frac{y-3}{2}$$

```
(%i282)g(y):=''%;
```

$$(\%o249) \quad g(y) := -\frac{y-3}{2}$$

```
(%i283)integrate(g(y),y,0,1);
```

$$(\%o250) \quad 1$$

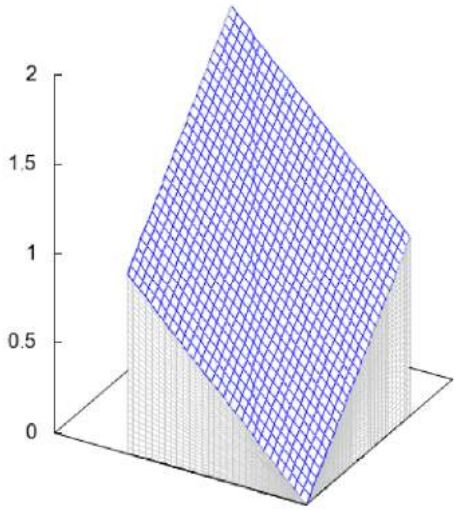


Figura 2.1: Sólido

Calculando diretamente com o Maxima tem-se:

Maxima

```
(%i284)integrate(integrate(2-x-y,x,0,1),y,0,1);
```

$$(\%o251) \quad 1$$

Exemplo 2.28 (Cálculo de volume) Calcule o volume do sólido limitado pelo parabolóide $z = 4 - x^2 - 2y^2$ e pelo plano xy.

Solução: Fazendo $z = 0$ temos que a região no plano xy é a elipse $x^2 + 2y^2 = 4$, como mostra a figura (2.2).

Neste caso, $-2 \leq x \leq 2$ e $\varphi_1(x) = -\sqrt{(4-x^2)/2}$ e $\varphi_2(x) = \sqrt{(4-x^2)/2}$, ou seja,

$$D = \left\{(x,y) \in \mathbb{R}^2 ; -2 \leq x \leq 2 ; -\sqrt{\frac{4-x^2}{2}} \leq y \leq \sqrt{\frac{4-x^2}{2}}\right\}.$$

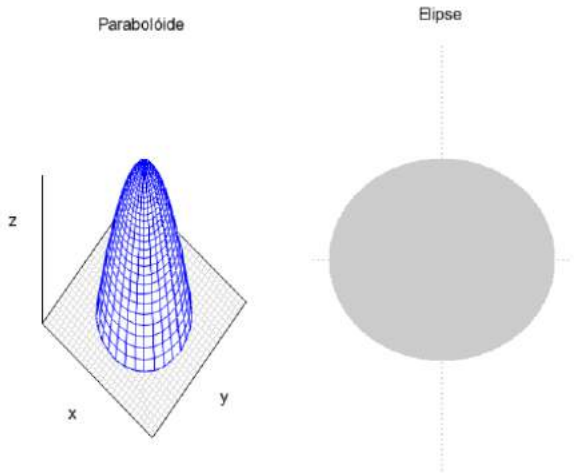


Figura 2.2: Parabolóide elíptico

Portanto, volume é dado por

Maxima

```
(%i285)assume(-2<=x and x<=2);
(%o252)   [x >= -2,x <= 2]
(%i286)integrate(integrate(4-x^2-2*y^2,y,
       -sqrt((4-x^2)/2),sqrt((4-x^2)/2)),x,-2,2);
(%o253)   2^(5/2) %pi
```

Exemplo 2.29 (Integral dupla) Desenhe a região de integração D e calcule a integral

$$\int_0^1 \int_{-\sqrt{1-y^2}}^{1-y} f(x,y)\,dx\,dy.$$

Solução: Para desenharmos vamos usar o pacote **draw**.

Maxima

```
(%i287)load(draw)$
(%i288)f1:1-x;
```

(%o254) $1-x$

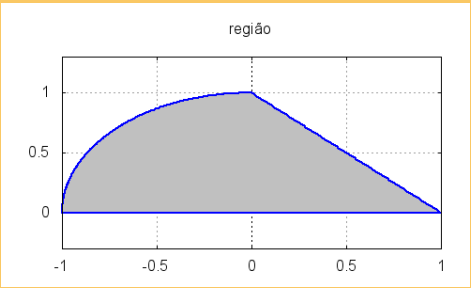
```
(%i289)f2: sqrt(1-x^2);
```

(%o255) $\sqrt{1-x^2}$

```
(%i290)f3:0;
```

(%o256) 0

```
(%i291)wxdraw2d(title = "região",fill_color = grey,
       filled_func = f3,explicit(f2,x,-1,0),
       filled_func = false,xaxis = true,
       yaxis = true,line_width = 2,key = "",color=blue,
       explicit(f3,x,-1,0),
       key = "",color = blue,
       explicit(f2,x,-1,0),fill_color = grey,
       filled_func = f3,explicit(f1,x,0,1),
       filled_func = false,xtics=true,
       ytics=1/2,lin_width = 2,key = "",color=blue,
       explicit(f3,x,0,1),key = "",color = blue,
       explicit(f1,x,0,1),yrange=[-.3,1.3],grid = true)
```



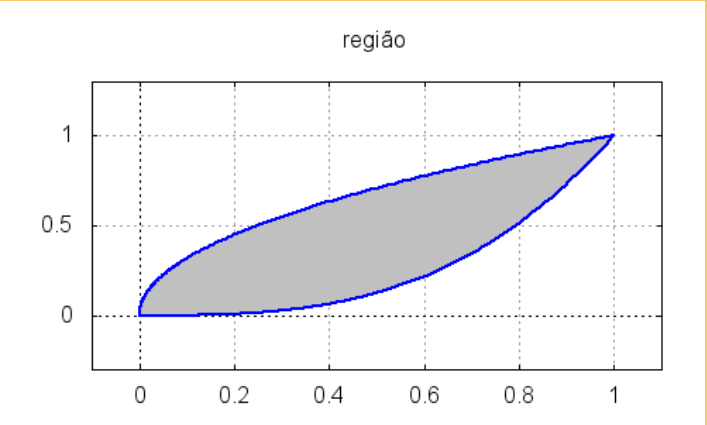
(%o257)

Exemplo 2.30 (Invertendo a ordem de integração) Desenhe a região de integração D e troque a ordem de integração da integral $\int_0^1\int_{x^3}^{\sqrt{x}} f(x,y)\,dy\,dx$.

Solução: Para desenharmos vamos usar o pacote **draw**.

Maxima

```
(%i292)load(draw)$
(%i293)f1:x^3$
       f2:sqrt(x)$
(%i294)wxdraw2d(title = "região",fill_color = grey,
       filled_func = f1,explicit(f2,x,0,1),
       filled_func = false,xaxis = true,
       yaxis = true,line_width = 2,key = "",color=blue,
       xtics=true,xrange=[-.1,1.1],
       ytics=1/2,li_width = 2,key = "",color=blue,
       explicit(f1,x,0,1),key = "",color = blue,
       explicit(f2,x,0,1),yrange=[-.3,1.3],grid = true);
```



(%o258)

Trocando a ordem de integração: $\int_0^1\int_{y^{1/3}}^{y^2} f(x,y)\,dx\,dy$.

Exemplo 2.31 (Invertendo a ordem de integração) Inverta a ordem de integração e depois calcule as seguintes integrais:

(a) $\int_0^1\int_{\sqrt{x}}^1 \sqrt{1+y^3}\,dy\,dx$; (b) $\int_0^1\int_y^1 \sqrt{1-x^2}\,dx\,dy$

Solução: (a) Invertendo a ordem tem-se $\int_0^1\int_0^{y^2} \sqrt{1+y^3}\,dx\,dy$.

Vamos calcular a integral dupla aplicando o *Teorema de Fubini*:

Maxima

```
(%i295)integrate(sqrt(1-y^3),x,0,y^2);
```

(%o259) $y^2\sqrt{1-y^3}$

```
(%i296)f(y):=''%;
```

(%o260) $f(y) := y^2\sqrt{1-y^3}$

Usar a substituição: $u = 1+y^3$; $du = 3y^2\,dy$

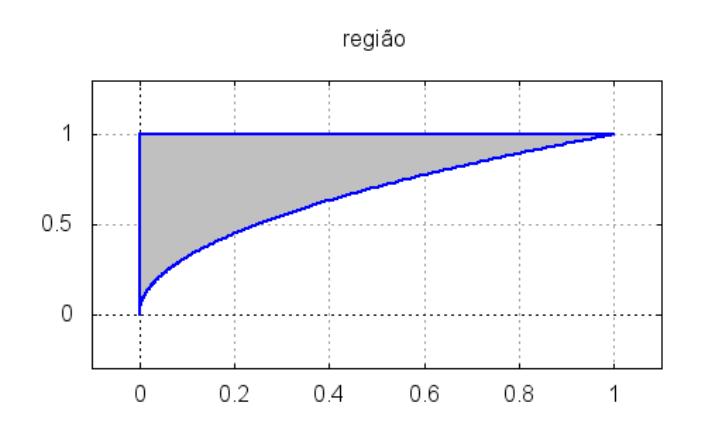
```
(%i297)integrate(f(y),y,0,1);
```

(%o261) $\frac{2}{9}$

Desenhando a região de integração:

Maxima

```
(%i298)f1:1$
       f2:sqrt(x)$
       R6:points([0,0],[0,1])$
(%i299)wxdraw(gr2d(title = "região",fill_color = grey,
       filled_func = f1,explicit(f2,x,0,1),
       filled_func = false,xaxis = true,yrange=[-.3,1.3],grid = true,
       yaxis = true,line_width = 2,key = "",color=blue,
       xtics=true,xrange=[-.1,1.1],
       ytics=1/2,line_width = 2,key = "",color=blue,
       explicit(f1,x,0,1),key = "",color = blue,
       explicit(f2,x,0,1),
       color=blue,point_type=0,points_joined=true,R6));
```



(%o262)

Solução: (b) Invertendo a ordem tem-se $\int_0^1 \int_0^x \sqrt{1-x^2}\, dy\, dx$.

Vamos calcular a integral dupla aplicando o *Teorema de Fubini*:

Maxima

```
(%i300)integrate(sqrt(1-x^2),y,0,x);
```

(%o263) $x\sqrt{1-x^2}$

```
(%i301)g(x):=''%;
```

(%o264) $g(x) := x\sqrt{1-x^2}$

Usar a substituição: $u = 1+x^2$; $du = 2x\,dx$.

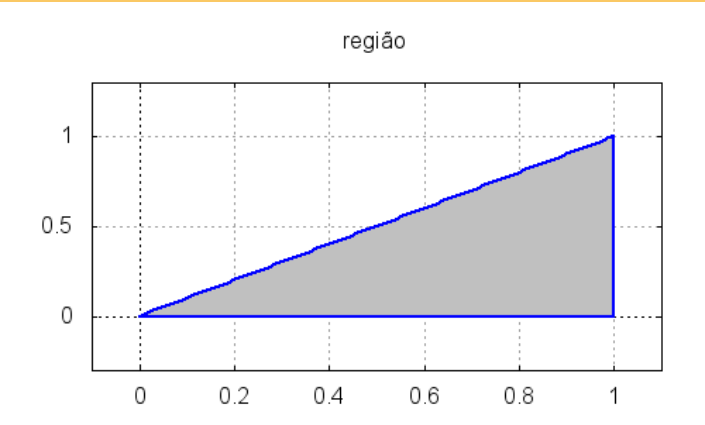
```
(%i302)integrate(g(x),x,0,1);
```

(%o265) $\frac{1}{3}$

Desenhando a região de integração:

Maxima

```
(%i303)f1:0
       $f2:x
       $R7:points([1,1],[1,0])$
(%i304)wxdraw(gr2d(title = "região",fill_color = grey,
       filled_func = f1,explicit(f2,x,0,1),
       filled_func = false,xaxis = true,yrange=[-.3,1.3],grid = true,
       yaxis = true,line_width = 2,key = "",color=blue,
       xtics=true,xrange=[-.1,1.1],
       ytics=1/2,line_width = 2,key = "",color=blue,
       explicit(f1,x,0,1),key = "",color = blue,
       explicit(f2,x,0,1),
       color=blue,point_type=0,points_joined=true,R7));
```



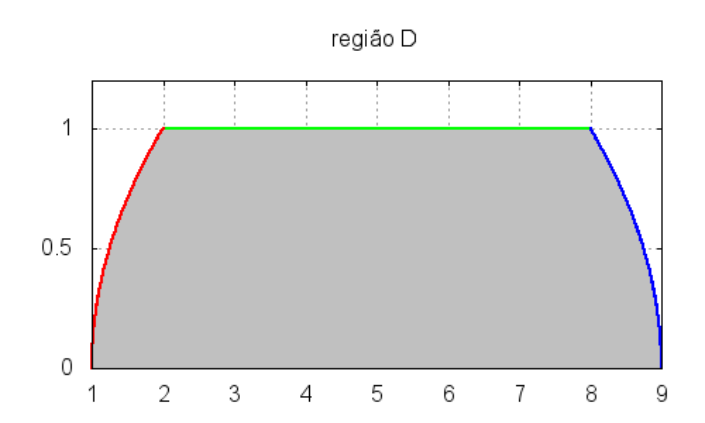
(%o266)

Exemplo 2.32 (Cálculo de volume) Calcule o volume do sólido no primeiro octante limitado pelas superfícies $z = 1-y^2$, $x = y^2+1$ e $x = -y^2+9$. Faça um esboço do sólido.

Solução: O volume é dado pela integral da função $z = 1 - y^2$ na região D a ser determinada. Primeiro, vamos desenhar a região D.

Maxima

```
(%i305)f1:sqrt(x-1)$
       f2:sqrt(9-x)$
(%i306)f3:1$
       R8:points([2,2],[0,1])$
       R9:points([8,8],[0,1])$
(%i307)wxdraw2d(title = "região D",fill_color = grey,
       filled_func = 0,explicit(f1,x,1,2),
       fill_color = grey,filled_func = 0,explicit(f3,x,2,8),
       filled_func = false,xaxis = true,
       yaxis = true,line_width = 2,key = "",color=blue,
       explicit(f3,x,2,8),
       key = "",color = blue,
       explicit(f1,x,1,2),fill_color = grey,
       filled_func = 0,explicit(f2,x,8,9),
       filled_func = false,xtics=true,
       ytics=1/2,line_width = 2,key = "",color=green,
       explicit(f3,x,2,8),
       key = "",color = red, explicit(f1,x,1,2),
       key = "",color = blue,
       explicit(f2,x,8,9),yrange=[0,1.2],grid = true);
```



(%o267)

A região de integração D é composta de 3 regiões. Portanto, a integral será dada por 3 integrais: $I_1 + I_2 + I_3$;

$$I_1 = \int_0^1 \int_{y^2+1}^2 1 - y^2 \, dx\, dy; \quad I_2 = \int_0^1 \int_2^8 1 - y^2 \, dx\, dy; \quad I_3 = \int_0^1 \int_8^{-y^2+9} 1 - y^2 \, dx\, dy.$$

Maxima

```
(%i308)I1:integrate(integrate(1-y^2,x,y^2+1,2),y,0,1);
```

$$(\%o268) \quad \frac{8}{15}$$

```
(%i309)I2: integrate(integrate(1-y^2,x,2,8),y,0,1);
```

$$(\%o269) \quad 4$$

```
(%i310)I3:integrate(integrate(1-y^2,x,8,-y^2+9),y,0,1);
```

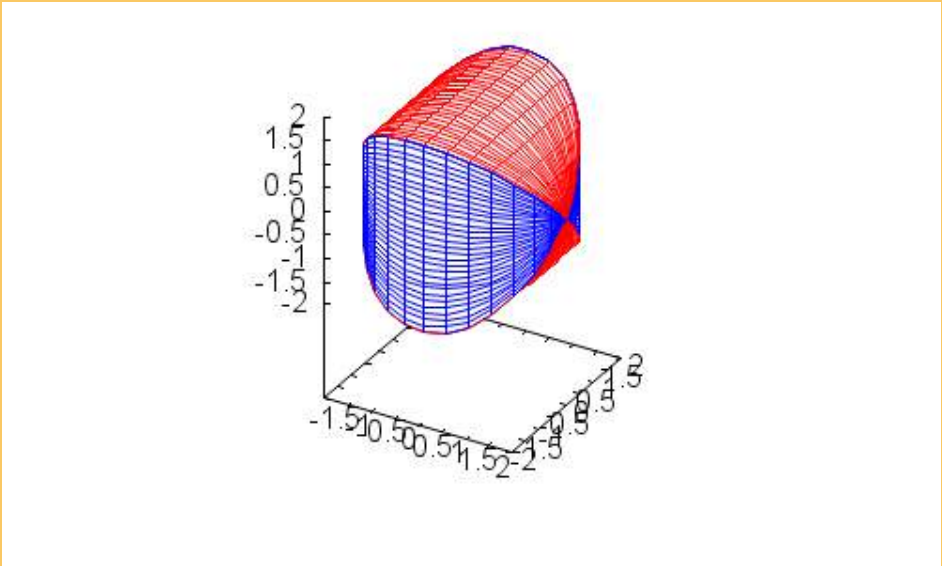
$$(\%o270) \quad \frac{8}{15}$$

Portanto, o volume é $V = \frac{76}{15}$.

O sólido é a intersecção de dois cilindros, o que não é trivial desenharmos com o `Maxima`. Para desenharmos este sólido faremos uma reparametrização das superfícies.

Maxima

```
(%i311)reparametrize(f1,f2,f3,iv,iv0,iv1,dv,dv0,dv1) :=
       apply(parametric_surface, append(subst([ iv = u ,
       dv = (1-v)*subst([iv=u],dv0) + v * subst([iv=u],dv1)],
       [f1,f2,f3]),[u, iv0, iv1, v, 0, 1]));
(%o271)
(%i312)wxdraw3d(
       surface_hide = true,
       user_preamble = "set view equal xyz;",
       reparametrize(2*cos(t), 2*sin(t), z, t, 0, 2*%pi, z,
       -2*abs(sin(t)), 2*abs(sin(t))),
       color = red,
       reparametrize(2*cos(t), z, 2*sin(t), t, 0, 2*%pi, z,
       -2*abs(sin(t)), 2*abs(sin(t))))
```



(%o272)

Exemplo 2.33 (Integral em coordenadas polares) Use integral dupla para calcular a área limitada pela rosácea $r = 4\cos(4t)$. Desenhe a região de integração.

Solução: Sabemos que $area = \iint_R dx\,dy = \int_\alpha^\beta \int_{s_1(t)}^{s_2(t)} r\,dr\,dt.$

Seja A a área de uma pétala da rosacea (veja figura 2.3). Então, a região de integração é dada por $0 \le r \le 2\cos(2,t)$ e $-\pi/4 \le t \le \pi/4$ e a área da pétala é

$$\frac{A}{4} = \iint_R dx\,dy = \int_{-\pi/4}^{\pi/4} \int_0^{2\cos(2t)} r\,dr\,dt.$$

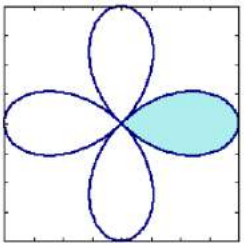


Figura 2.3: Rosacea

Calculado com o Maxima obtemos

Maxima

```
(%i313)4*integrate(integrate(r,r,0,2*cos(2*t)),t,-%pi/4,%pi/4);
```

(%o273) 2π

Exemplo 2.34 (Integral em coordenadas polares) Seja R a região anular limitada pelos círculos $x^2+y^2=1$ e $x^2+y^2=4$. Calcule a integral $\iint_R (x^2+y)\,dx\,dy$. Desenhe a região de integração.

Solução: Os extremos polares são $1 \leq r \leq 2$ e $0 \leq t \leq 2\pi$. Faremos a substituição $x = r\,cos(t)$ e $y = r\,\mathrm{sen}(t)$. Assim, integral é dada por:

Maxima

```
(%i314)f(x,y):=x^2+y;
```

(%o274) $f(x,y) := x^2 + y$

```
(%i315)[x,y]:[r*cos(t),r*sin(t)];
```

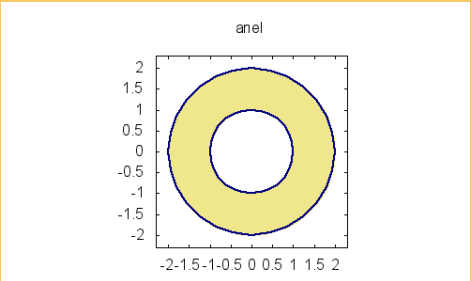
(%o275) $[r\cos(t), r\,\mathrm{sen}(t)]$

```
(%i316)integrate(integrate(f(x,y)*r,r,1,2),t,0,2*%pi);
```

(%o276) $\frac{15\pi}{4}$

Desenhando a região de integração:

Maxima



(%o277)

Exemplo 2.35 (Integral tripla) Calcule a integral tripla

$$\int_0^2 \int_0^x \int_0^{x+y} e^x(y+2z)\,dz\,dy\,dx$$

Solução:

Maxima

```
(%i317)integrate(integrate(integrate(exp(x)*(y+2*z),z,0,x+y),y,0,x),
       x,0,2);
```

$$(\%o278) \quad \frac{19(2\%e^2+6)}{6}$$

Exemplo 2.36 (Coordenadas cilíndricas) Calcule o volume do elipsóide $4x^2+4y^2+z^2=16$

Solução: Sabemos que o volume é dado por

$$V = \iiint_S dV = 8\int_0^2 \int_0^{\sqrt{4-x^2}} \int_0^{2\sqrt{4-x^2-y^2}} dz\,dy\,dx.$$

Em coordenadas cilíndricas tem-se

$$V = 8\int_0^{\pi} \int_0^2 \int_0^{2\sqrt{4-r^2}} r\,dz\,dr\,dt.$$

Maxima

```
(%i318)assume(r>=0 and r<=2)$
(%i319)8*integrate(integrate(integrate(r,z,0,sqrt(4-r^2)),r,0,2),
       t,0,%pi);
```

$$(\%o279) \quad \frac{64\pi}{3}$$

Exemplo 2.37 (Coordenadas cilíndricas) Considere o sólido S limitado pelo paraboloide $z=x^2+y^2$ e pelo plano $z=4$. Calcule

$$\iiint_S (x^2+y^2)\sqrt{x^2+y^2}\,dz\,dy\,dx$$

Solução: Usando coordendas cilíndricas tem-se

$$\iiint_S (x^2+y^2)\sqrt{x^2+y^2}\,dz\,dy\,dx = \int_0^4 \int_0^{2\pi} \int_0^{\sqrt{z}} r^4\,dr\,dt\,dz.$$

Maxima

```
(%i320)assume(z>=0)$
(%i321)f(x,y,z):=(x^2+y^2)*sqrt(x^2+y^2)$
(%i322)[x,y,z]:[r*cos(t),r*sin(t),z]$
(%i323)integrate(integrate(integrate(trigsimp(f(x,y,z))*r,r,0,
       sqrt(z)),t,0,2*%pi),z,0,4);
(%o280)  512*%pi/35
```

Exemplo 2.38 (Coordenadas esféricas) Calcule em coordanadas esféricas a seguinte integral:

$$\int_{-2}^{2} \int_0^{\sqrt{4-x^2}} \int_0^{\sqrt{4-x^2-y^2}} y\,z\,dz\,dy\,dx$$

Solução: Usando coordenadas esféricas tem-se

$$\int_{-2}^{2} \int_0^{\sqrt{4-x^2}} \int_0^{\sqrt{4-x^2-y^2}} y\,z\,dz\,dy\,dx =$$
$$= \int_0^{\pi/2} \int_0^{\pi} \int_0^{2} (\rho\,\mathrm{sen}(\phi)\mathrm{sen}(\theta)\rho\cos(\phi))\rho^2\,\mathrm{sen}(\phi)\,d\rho\,d\theta\,d\phi.$$

Maxima

```
(%i324)f(x,y,z):=y*z$
(%i325)[x,y,z]:[rho*sen(phi)*cos(theta),rho*sin(phi)*sin(theta),
       rho*cos(phi)]$
(%i326)integrate(integrate(integrate(f(x,y,z)*rho^2*sin(phi),rho,0,2),
       theta,0,%pi),phi,0,%pi/2);
(%o281)  64/15
```

Exemplo 2.39 (Coordenadas esféricas) Calcule o volume do sólido S limitado por baixo pelo cone $z^2 = x^2 + y^2$, $z \geq 0$ e por cima pela esfera $x^2 + y^2 + z^2 = 9$

Solução: Usando coordenadas esféricas tem-se

$$V = \int_0^{\pi/4} \int_0^{2\pi} \int_0^3 \rho^2 \operatorname{sen}(\phi)\, d\rho\, d\theta\, d\phi.$$

Maxima

```
(%i327)integrate(integrate(integrate(rho^2*sin(phi),rho,0,3),
       theta,0,2*%pi),phi,0,%pi/4);
```

(%o282) $18\left(1 - \frac{1}{\sqrt{2}}\right)$

Exemplo 2.40 (Jacobiano em coordenadas polares) Calcule o jacobiano da mudança de variável $x = r\operatorname{sen}(t)$ e $y = r\cos(t)$.

Solução:

Maxima

```
(%i328)x(r,t):=r*cos(t)$
(%i329)y(r,t):=r*sin(t)$
(%i330)Jpolar:matrix([diff(x(r,t),r),diff(x(r,t),t)],[diff(y(r,t),r),
       diff(y(r,t),t)]);
```

(%o283) $\begin{bmatrix} \cos(t) & -r\operatorname{sen}(t) \\ \operatorname{sen}(t) & r\cos(t) \end{bmatrix}$

```
(%i331)JaconianoPolar:trigsimp(determinant(Jpolar));
```

(%o284) r

Exemplo 2.41 (Jacobiano) Seja R a região limitada por $x-2y = 0$, $x-2y = 4$, $x+y = 4$ e $x+y = 1$. Calcule a integral $\iint_R 3xy\, dx\, dy$. Desenhe a região de integração.

Solução: Sabemos que

$$\iint_R 3xy\, dx\, dy = \iint_Q f(g(u,v), h(u,v))|J|\, du\, dv.$$

Maxima

```
(%i332)f(x,y)=3*x*y$
(%i333)x(u,v):=(2*u+v)/3$
(%i334)y(u,v):=(u-v)/3$
(%i335)J1:matrix([diff(x(u,v),u),diff(x(u,v),v)],[diff(y(u,v),u),
       diff(y(u,v),v)]);
```

$$(\%o285)\quad \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} \end{bmatrix}$$

```
(%i336)J01:trigsimp(determinant(J1));
```

$$(\%o286)\quad -\frac{1}{3}$$

```
(%i337)integrate(integrate(f(x,y)*abs(J01),v,-4,0),u,1,4);
```

$$(\%o287)\quad \frac{164}{9}$$

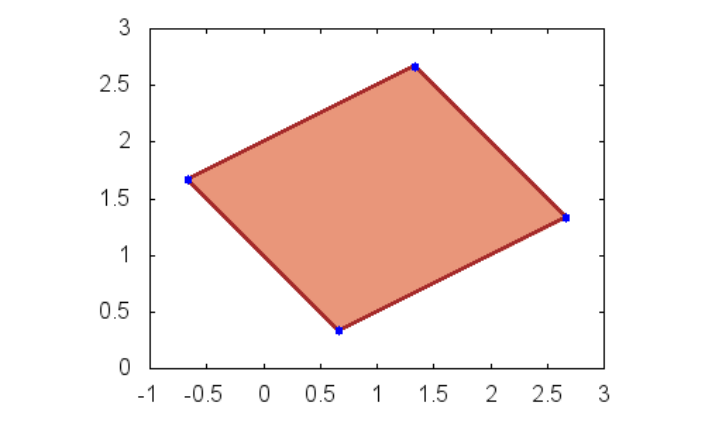
Desenhando a região de integração:

Maxima

```
(%i338)reg1:region(x-2*y<0 and x-2*y>-4and x+y<4 and
       x+y>1,x,-1,3,y,0,3)$
(%i339)reta1:explicit(x/2,x,2/3,8/3)$
(%i340)reta2:explicit((x+4)/2,x,-2/3,4/3)$
(%i341)reta3:explicit(4-x,x,4/3,8/3)$
(%i342)reta4:explicit(1-x,x,-2/3,2/3)$
```

Maxima

```
(%i343)wxdraw2d(proportional_axes=xy,x_voxel=40,y_voxel=40,nticks=450,
       line_width=2,fill_color=dark-salmon,reg1,color=brown,
       reta1,color=brown,reta2,color=brown,reta3,color=brown,reta4,
       color=blue, point_type=7,point_size=1,A1,B1,C1,D1);
```



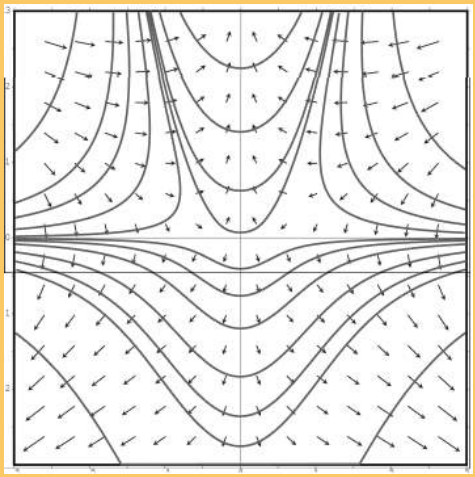
(%o288)

Exemplo 2.42 (Campo vetorial conservativo) Esboce o campo vetorial gerado pelo gradiente da função potencial $f(x,y) = x^2 y - \frac{y^2}{2}$.

Solução: Usaremos o comando **ploteq**(f(x,y),[x,y],[x,a,b],[y,c,d]). Para obter o campo clique na grade gerada pelo `Maxima`.

Maxima

```
(%i344)ploteq(x^2*y-y^2/2,[x,y],[x,-3,3],[y,-3,3],[vectors,"blue"]);
```



(%o289)

Exemplo 2.43 (Integral de linha) Calcule $\int_C (x^2 - y + 3z)\,ds$ com C o segmento de reta de extremidades $(0,0,0)$ e $(1,2,1)$.

Solução: Sabemos que a forma parametrizada do segmento de reta é $\vec{r} = (t, 2t, t)$ com $0 \leq t \leq 1$ e

$$\int_C f(x,y,z)\,ds = \int_a^b f(\vec{r}(t)) \left\| \frac{d\vec{r}}{dt} \right\| dt$$

com $f(x,y,z) = x^2 - y + 3z$.

Maxima

```
(%i345)f(x,y,z):=x^2-y+3*z$
(%i346)[x,y,z]:[t,2*t,t]$
(%i347)dx:diff([x,y,z],t)$
(%i348)linha:integrate(f(x,y,z)*sqrt(dx.dx),t,0,1);
```

$$(\%o290) \quad \frac{5}{\sqrt{6}}$$

Exemplo 2.44 (Trabalho) Calcule o trabalho realizado pelo campo de forças

$$\vec{F}(x,y,z) = \left(-\frac{x}{2}, -\frac{y}{2}, \frac{1}{4}\right)$$

sobre uma partícula que se move ao longo de uma hélice dada por $\vec{r}(t) = (cos(t), sen(t), t)$.

Solução:

Maxima

```
(%i349)F(x,y,z):=[-x/2,-y/2,1/4]$
(%i350)[x,y,z]:[cos(t),sin(t),t]$
(%i351)dr:diff([x,y,z],t)$
(%i352)W:integrate(F(x,y,z).dr,t,0,3*%pi);
```

$$(\%o291) \quad \frac{3\pi}{4}$$

Exemplo 2.45 (Rotacional) O campo vetorial $\vec{F}(x,y,z) = (2xy, x^2+z^2, 2yz)$ é irrotacional?

Solução: Mostraremos que **rot F** $= 0$. Para isso, usaremos o pacote **vect**.

Maxima

```
(%i353)load(vect)$
(%i354)F(x,y,z):=[2*x*y,x^2+z^2,2*y*z]$
(%i355)curl(F(x,y,z))$
(%i356)express(%);
```

$$(\%o292)\quad \frac{d}{dy}(2yz)-\frac{d}{dz}(z^2+x^2),\frac{d}{dz}(2xy)-\frac{d}{dx}(2yz),\frac{d}{dx}(z^2+x^2)-\frac{d}{dy}(2xx)$$

```
(%i357)ev(%,diff);
```

$$(\%o293)\quad [0,0,0]$$

Exemplo 2.46 (Divergente) Determine o divergente em $(2,1,-1)$ do campo vetorial $\vec{F}(x,y,z) = (x^3y^2z, x^2z, x^2y)$.

Solução:

Maxima

```
(%i358)load(vect)$
(%i359)F(x,y,z):=[x^3*y^2*z,x^2*z+z^2,x^2*y]$
(%i360)divergente:diff(F(x,y,z)[1],x)+diff(F(x,y,z)[2],y)+
       diff(F(x,y,z)[3],z);
```

$$(\%o294)\quad 3x^2y^2z$$

```
(%i361)ev(%,x=2,y=1,z=-1);
```

$$(\%o295)\quad -12$$

2.3 Desenhando curvas

Exemplo 2.47 Desenhe as seguintes curvas:

(a) $(x,y) = (4\,\text{sen}(2\,t), 2\,\cos(2\,t))$; (b) $(x,y) = (\cos(t) + t\,\text{sen}(t), \text{sen}(t) - t\,\cos(t))$;

(c) $(x,y) = (|t-1|, t+2)$; (d) $(x,y) = \left(\dfrac{3\,t}{1+t^3}, \dfrac{3\,t^2}{1+t^3}\right)$.

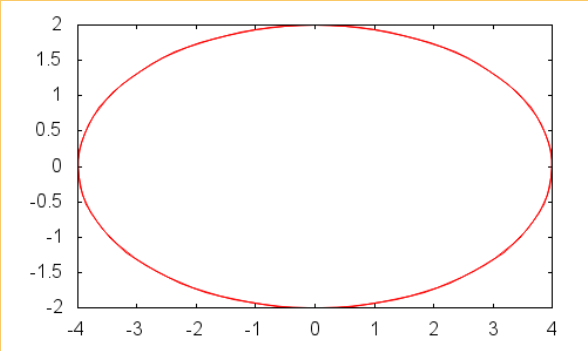
Solução: (a)

Maxima

```
(%i362)load(draw)$
(%i363)curva1:parametric(4*sin(2*t),2*cos(2*t),t,0,2*%pi);
```

(%o296) parametric(4 sin(2 t), 2 cos(2 t), t, 0, 2 π)

```
(%i364)draw2d(nticks=100,color=red,curva1);
```



(%o297)

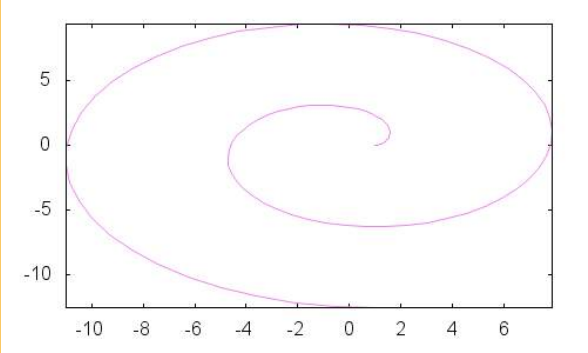
Solução: (b)

Maxima

```
(%i365)curva2:parametric(cos(t)+t*sin(t),sin(t)-t*cos(t),t,0,4*%pi);
```

(%o298) parametric(t sen(t) + cos(t), sin(t) − t cos(t), t, 0, 4 ∗ π)

```
(%i366)draw2d(nticks=100,color=violet,curva2);
```



(%o299)

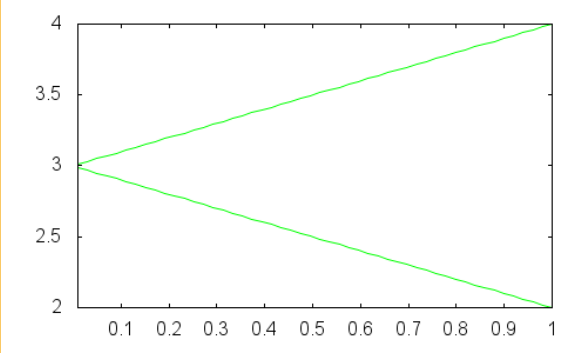
Solução: (c)

Maxima

```
(%i367)curva3:parametric(abs(t-1),t+2,t,0,2);
```

(%o300) $\text{parametric}(|t-1|, t+2, t, 0, 2)$

```
(%i368)draw2d(nticks=100,color=green,curva3);
```



(%o301)

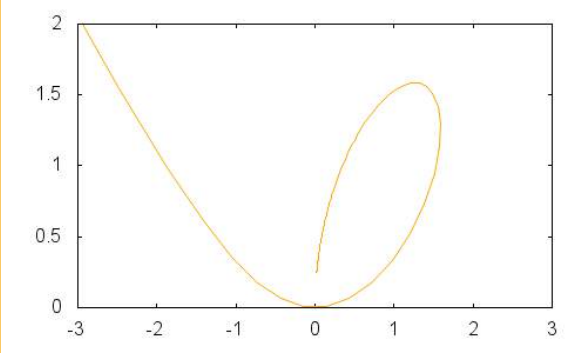
Solução: (d)

Maxima

```
(%i369)curva4:parametric(3*t/(1+t^3),3*t^2/(1+t^3),t,-12,12);
```

(%o302) $\text{parametric}\left(\frac{3t}{t^3+1}, \frac{3t^2}{t^3+1}, t, -12, 12\right)$

```
(%i370)draw2d(nticks=250,xrange=[-3,3],yrange=[0,2],color=orange,
       curva4);
```



(%o303)

Exemplo 2.48 A ciclóide é a curva descrita por um ponto P sobre um círculo de raio "a" rolando ao longo de uma reta num plano. As equações paramétricas da ciclóide são

$$(x, y) = a(t - \text{sen}(t)), a(1 - \cos(t)).$$

Desenhe a ciclóide $(x, y) = 2(t - \text{sen}(t)), 2(1 - \cos(t))$

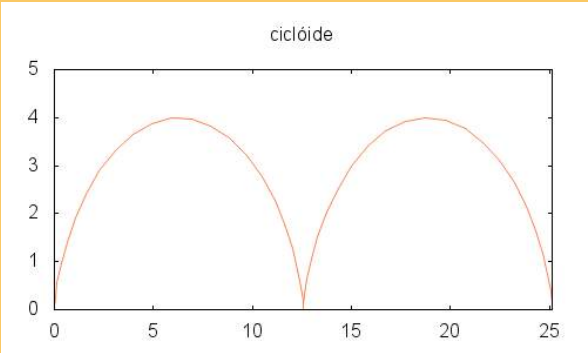
Solução:

Maxima

```
(%i371)load(draw)$
(%i372)cicloide:parametric(2*(t-sin(t)),2*(1-cos(t)),t,0,8*%pi);
```

(%o304) parametric(2 (t − sen(t)), 2 (1 − cos(t)), t, 0, 8 π)

```
(%i373)draw2d(nticks=100,xrange=[0,8*%pi],yrange=[0,5],
       title="ciclóide",color=coral,cicloide);
```



(%o305)

Exemplo 2.49 A epiciclóide é a curva descrita por um ponto P sobre um círculo de raio "b" que rola do lado de fora de um círculo de raio "a" com $a > b$. As equações paramétricas da epiciclóide são

$$\begin{aligned} x &= (a+b)\cos(t) - b\cos\left(\frac{a+b}{b}\right); \\ y &= (a+b)\text{sen}(t) - b\,\text{sen}\left(\frac{a+b}{b}\right). \end{aligned}$$

Desenhe a epiciclóide

$$\begin{aligned} x &= 11\cos(t) - 3\cos\left(\frac{11}{3}\right); \\ y &= 11\,\text{sen}(t) - 3\,\text{sen}\left(\frac{11}{3}\right). \end{aligned}$$

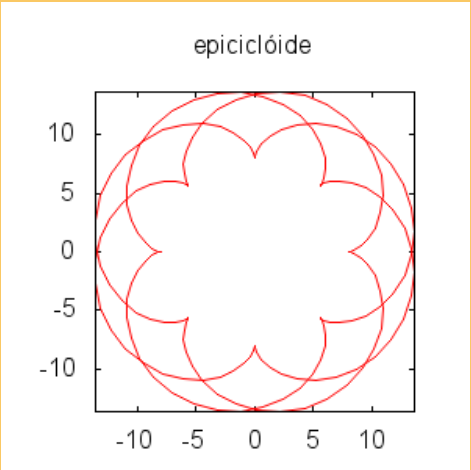
Solução:

Maxima

```
(%i374)load(draw)$
(%i375)set_draw_defaults(user_preamble="set size ratio -1",nticks=200,
       dimensions=[300,300],title="epiciclóide",color=red)$
(%i376)epicicloide:parametric(11*cos(t)-3*cos(11/3*t),
       11*sin(t)-3*sin(11/3*t),t,0,6*%pi);
```

(%o306) $\text{parametric}(11\cos(t) - 3\cos\left(\frac{11\,t}{3}\right), 11\,\text{sen}(t) - 3\,\text{sen}\left(\frac{11\,t}{3}\right), t, 0, 6\,\pi)$

```
(%i377)wxdraw2d(epicicloide);
```



(%o307)

Exemplo 2.50 A hipociclóide é a curva descrita por um ponto P sobre um círculo de raio "b" que rola no interior de um círculo de raio "a" com $a > b$. As equações paramétricas da epiciclóide são

$$\begin{aligned} x &= (a-b)\cos(t) + b\cos\left(\frac{a-b}{b}\right); \\ y &= (a-b)\mathrm{sen}(t) - b\,\mathrm{sen}\left(\frac{a-b}{b}\right). \end{aligned}$$

Desenhe a hipociclóide

$$\begin{aligned} x &= 5\cos(t) + 3\cos\left(\frac{5}{3}\right); \\ y &= 5\mathrm{sen}(t) - 3\,\mathrm{sen}\left(\frac{5}{3}\right). \end{aligned}$$

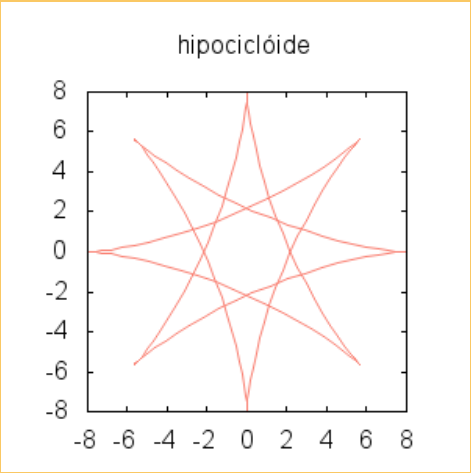
Solução:

Maxima

```
(%i378)load(draw)$
(%i379)set_draw_defaults(user_preamble="set size ratio -1",nticks=200,
       dimensions=[300,300],title="hipociclóide",color=salmon)$
(%i380)hipocicloide:parametric(5*cos(t)+3*cos(5/3*t),
       5*sin(t)-3*sin(5/3*t),t,0,6*%pi);
```

(%o308) $\mathrm{parametric}(5\cos(t) + 3\cos\left(\frac{5t}{3}\right), 5\,\mathrm{sen}(t) - 3\,\mathrm{sen}\left(\frac{5t}{3}\right), t, 0, 6\pi)$

```
(%i381)wxdraw2d(hipocicloide);
```



(%o309)

Exemplo 2.51 Desenhe as seguintes curvas:

(a) rosácea: $r = \cos\left(\frac{2\theta}{3}\right)$ com $\theta \in [0, 6\pi]$;

(b) Espiral hiperbólica: $r = 2/\theta$;

(c) Estrofóide: $r = 2\cos(2\theta)\sec(\theta)$.

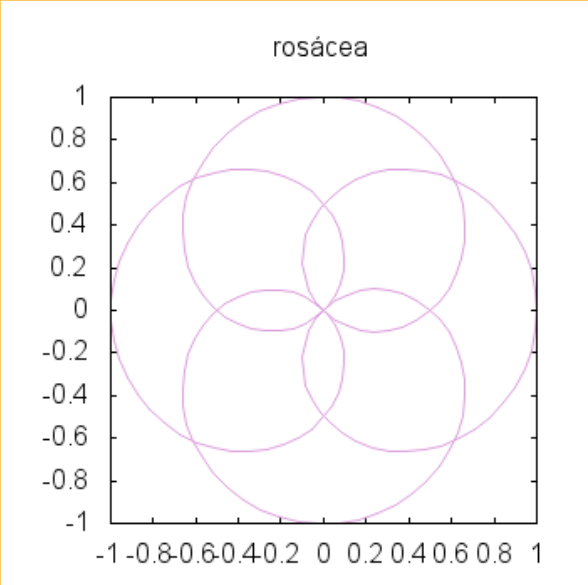
Solução: (a)

Maxima

```
(%i382)load(draw)$
(%i383)set_draw_defaults(user_preamble="set size ratio -1",nticks=200,
       dimensions=[350,350],title="rosácea",color=plum)$
(%i384)rosacea: polar(cos(2*th/3),th,0,6*%pi);
```

(%o310) $\text{polar}\left(\cos\left(\frac{2\,\text{th}}{3}\right), \text{th}, 0, 6\pi\right)$

```
(%i385)wxdraw2d(rosacea);
```



(%o311)

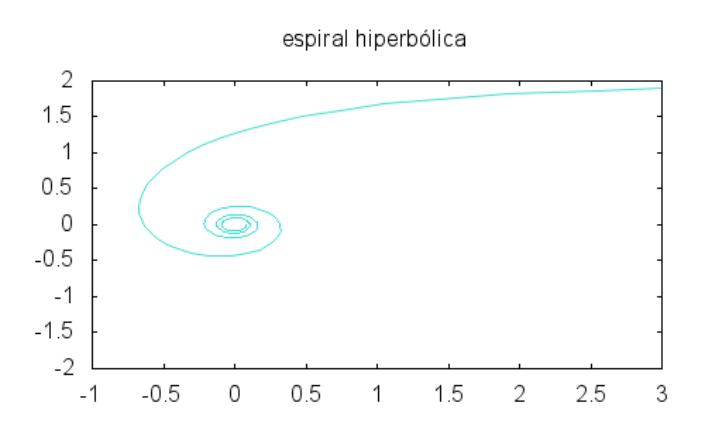
Solução: (b)

Maxima

```
(%i386)espiral: polar(2/th,th,0,8*%pi);
```

(%o312) $\text{polar}\left(\frac{2}{\text{th}}, \text{th}, 0, 8\pi\right)$

```
(%i387)wxdraw2d(nticks=200,title="espiral hiperbólica",color=turquoise,
       xrange=[-1,3],yrange=[-2,2],espiral);
```



(%o313)

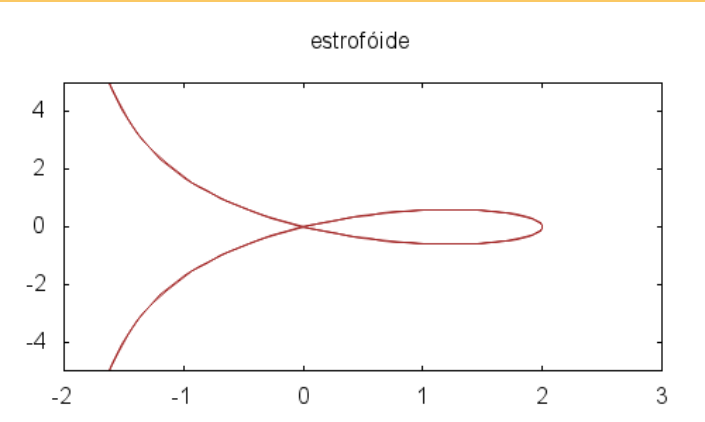
Solução: (c)

Maxima

```
(%i388)estrofoide:polar(2*cos(2*th)*sec(th),th,0,2*%pi);
```

(%o314) $\text{polar}(2\cos(2\,\text{th})\sec(\text{th}), \text{th}, 0, 2\pi)$

```
(%i389)wxdraw2d(nticks=200,title="estrofóide",color=brown,
       xrange=[-2,3],yrange=[-5,5],estrofoide);
```



(%o315)

2.4 Desenhando superfícies

Exemplo 2.52 Desenhe o plano $x - 5y + 2z = 0$.

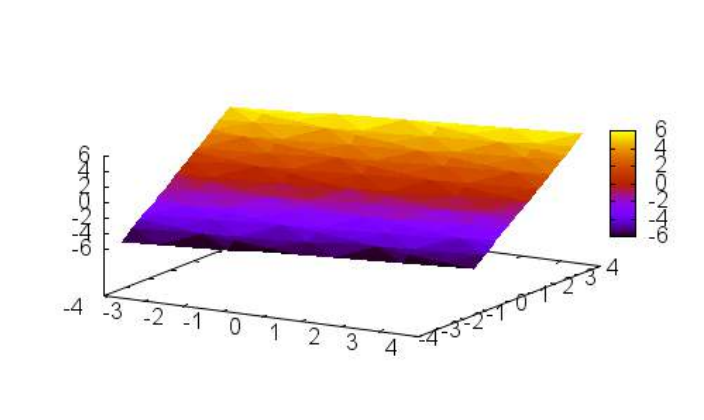
Solução:

Maxima

```
(%i390)plano1:x-5*y+2*z =0;
```

(%o316) $2z - 5y + x = 0$

```
(%i391)draw3d(enhanced3d = true,implicit(plano1,x,-4,4,y,-4,4, z,-6,6));
```



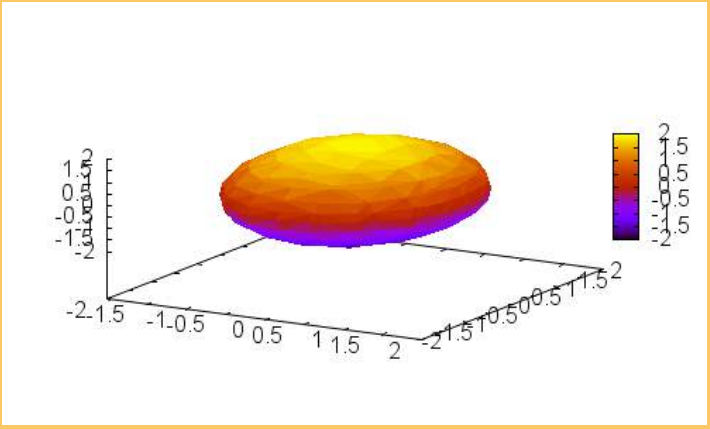
(%o317)

Exemplo 2.53 Desenhe o elipsóide $\frac{x^2}{2} + \frac{y^2}{3} + \frac{z^2}{4} = 1$.

Solução:

Maxima

```
(%i392)elipsoide1:x^2/2+y^2/3+z^2/4=1;
(%i393)draw3d(enhanced3d =true,
       implicit(elipsoide1,x,-2,2,y,-2,2,z,-2,2));
```



(%o318)

Exemplo 2.54 Desenhe a parte do sólido no 1^o octante, limitado pelas superfícies $x = y^2 + 1$, $x = -y^2 + 9$ e $z = 1 - y^2$.

Solução: Vamos parametrizar as superfícies, considerando $0 \leq y \leq 1$ e $0 \leq z \leq 1$. Assim,

$S_1 : \ x = y^2 + 1 \Rightarrow y = t, x = t^2 + 1; \ 0 \leq t \leq 1; \ 0 \leq z \leq 1 - t^2;$

$S_2 : \ x = -y^2 + 9 \Rightarrow y = t, x = -t^2 + 9; \ 0 \leq t \leq 1; \ 0 \leq z \leq 1 - t^2;$

$S_3 : \ z = 1 - y^2 \Rightarrow y = t, z = 1 - t^2.$

Para obtermos a variação da variável "x" , vamos calcular as interseções entre as superfícies S_1 e S_3 e as superfícies S_2 e S_3. Assim,

$S_1 \cap S_3 : \ x + z = 2 \Rightarrow x = 1 + t^2;$

$S_2 \cap S_3 : \ x - z = 8 \Rightarrow x = 9 - t^2.$

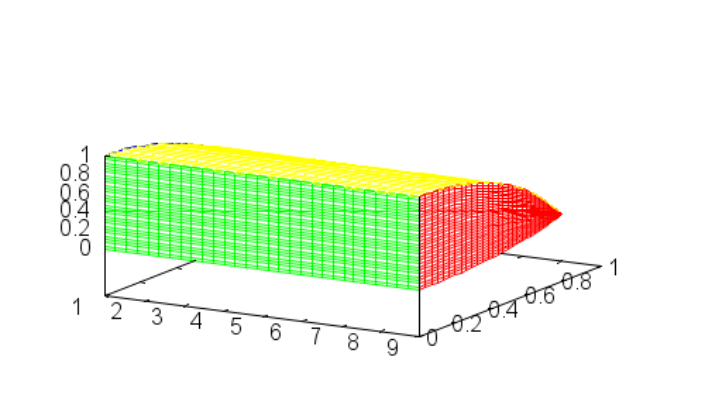
Logo, $0 \leq t \leq 1; \ 1 + t^2 \leq x \leq 9 - t^2$.

Para desenharmos este sólido usaremos o comando **draw3d**. Como os extremos das superfícies são funções do parâmetro, faremos uma reparematrização para termos extremos fixos.

Primeiro, desenharemos as superfícies do tipo *grade* e depois do tipo *cheia*. Para melhor vizualização, desenharemos o plano xz e no tipo *grade* desenharemos cada superfície de uma cor diferente.

Maxima

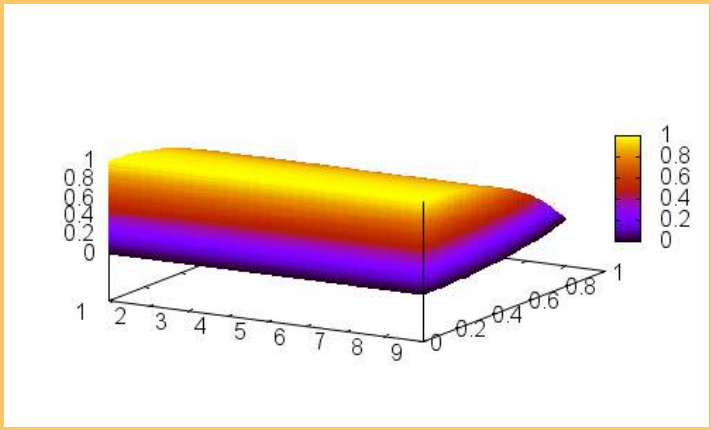
```
(%i394)reparametrize(f1,f2,f3,iv,iv0,iv1,dv,dv0,dv1) :=
       apply(parametric_surface, append(subst([ iv = u ,
       dv = (1-v)*subst([iv=u],dv0) + v * subst([iv=u],dv1)],
       [f1,f2,f3]),[u, iv0, iv1, v, 0, 1]));
(%o319)  out
(%i395)wxdraw3d(surface_hide = true,
       reparametrize(t^2+1,t,z, t, 0, 1, z,0,1-t^2),
       color = red,reparametrize(-t^2+9,t,z, t, 0, 1, z,0,1-t^2),
       color = yellow,reparametrize(x,t,1-t^2, t, 0,1, x, 1+t^2,
       9-t^2),
       color = green,reparametrize(x,0,z, x, 1,9, z, 0, 1))
```



(%o320)

Maxima

```
(%i396)wxdraw3d(enhanced3d = true,
       reparametrize(t^2+1,t,z, t, 0, 1, z,0,1-t^2),
       color = red,reparametrize(-t^2+9,t,z, t, 0, 1, z,0,1-t^2),
       color = yellow,reparametrize(x,t,1-t^2, t, 0,1, x, 1+t^2,
       9-t^2),
       color = green,reparametrize(x,0,z, x, 1,9, z, 0, 1))
```




```
(%o321)
```

Exemplo 2.55

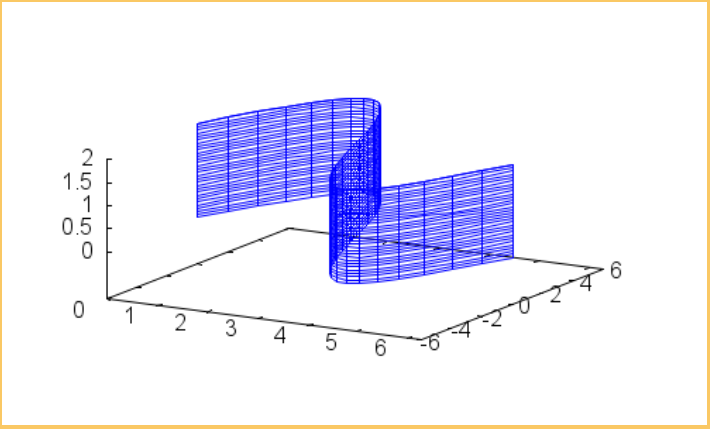
(a) Desenhe o cilindro senoidal $y = 2\,sen(x)$;

(b) Desenhe o Hiperbolóide $z = x^2 - y^2$, $(x, y) \in [-2, 2] \times [-2, 2]$ e suas curvas de níveis.

Solução: (a)

Maxima

```
(%i397)draw3d(enhanced3d=false,surface_hide=false,
       parametric_surface(t,6*sin(t),u,t,0,2*%pi,u,0,2));
```

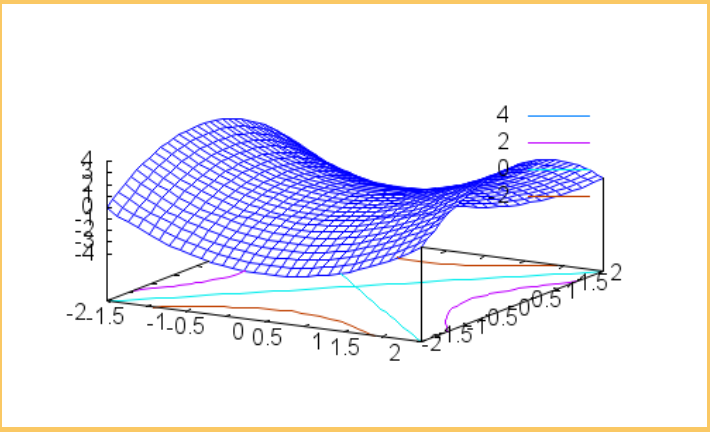


(%o322)

Solução: (b)

Maxima

```
(%i398)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       explicit(x^2-y^2,x,-2,2,y,-2,2))
```



(%o323)

Exemplo 2.56 Parabolóide hiperbólico é uma superfície de revolução obtida girando a hiperbóle $(t, 1/t)$ em torno do eixo x. Uma parametrização desta superfície é

$$\left(t, \frac{cos(v)}{t}, \frac{sen(v)}{t})\right)$$

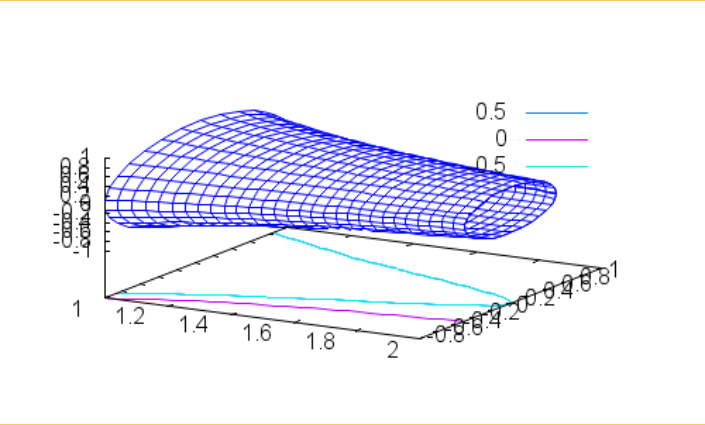
com $1 \leq t \leq 2$ e $0 \leq v \leq 2\pi$.

Desenhe o parabolóide usando esta parametrização em torno dos eixos y e z. Desenhe as curvas de níveis.

Solução:

Maxima

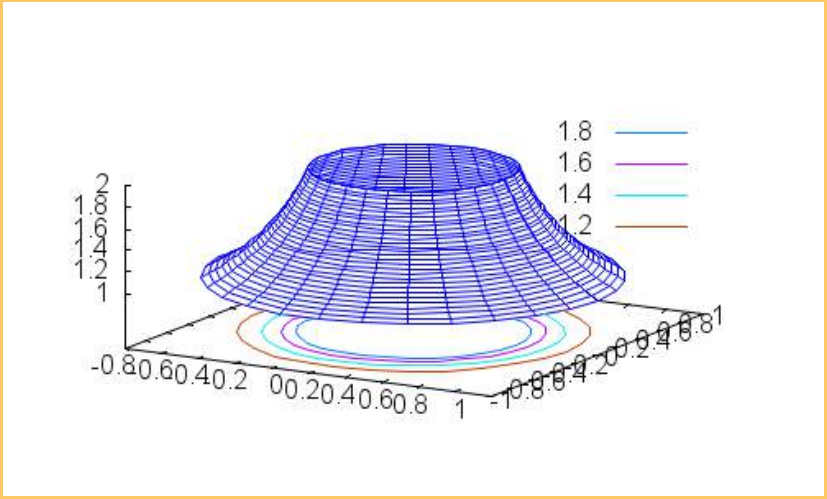
```
(%i399)wxdraw3d(enhanced3d=false,colorbox=false, surface_hide=true,
       contour=base,parametric_surface
       (u,cos(v)/u,sin(v)/u,u,1,2,v,0,2*%pi))
```




```
(%o324)
```

Maxima

```
(%i400)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(cos(v)/u,sin(v)/u,u,u,1,2,
       v,0,2*%pi))
```



(%o325)

Exemplo 2.57 Uma parametrização do Elipsóide é

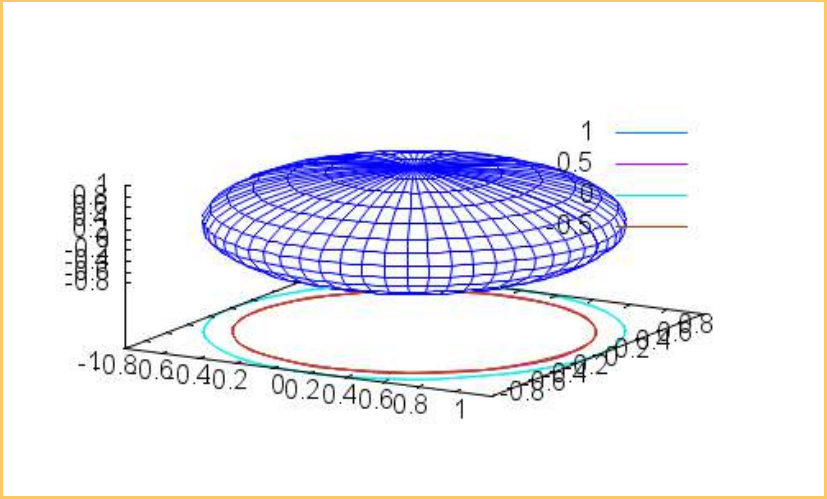
$$(a\,sen(u)\cos(v), b\,sen(u)\,sen(v), c\cos(u))$$

com $(u,v) \in [0,2\pi] \times [0,2\pi]$ e $a,b,c \in \mathbb{R}$. Desenhe o elipsóide e as curvas de níveis.

Solução:

Maxima

```
(%i401)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(sin(u)*cos(v),sin(u)*sin(v),cos(u),
       u,0,2*%pi,v,0,2*%pi))
```



(%o326)

Exemplo 2.58 Uma parametrização do Cone é

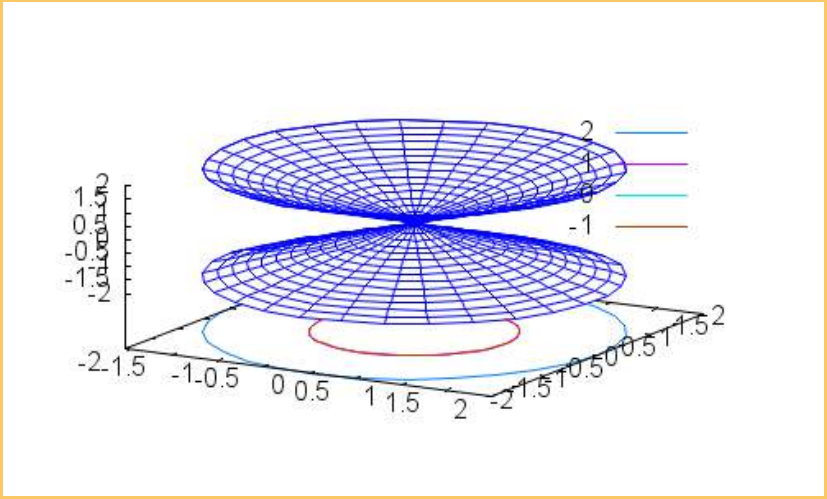
$$(u\,cos(v), u\,sen(v), u)$$

com $(u,v) \in [a,b] \times [0, 2\%pi]$ e $a,b \in \mathbb{R}$. Desenhe o cone no formato grade de no formado cheio. Desenhe curvas de níveis.

Solução:

Maxima

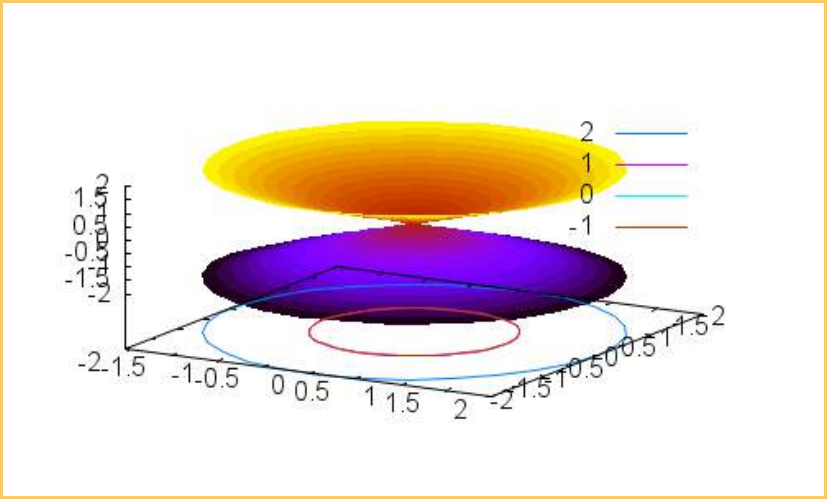
```
(%i402)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(u*cos(v),u*sin(v),u,u,-2,2,
       v,0,2*%pi))
```



(%o327)

Maxima

```
(%i403)wxdraw3d(enhanced3d=true,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(u*cos(v),u*sin(v),u,u,-2,2,
       v,0,2*%pi))
```



(%o328)

Exemplo 2.59 O Catenóide é uma superficie de revolução obtida girando-se a catenária $(t, a\,cosh(t/a))$ em torno do eixo x, com $-b \leq t \leq b$ e $a, b > 0$. Uma parametrização do catenóide é

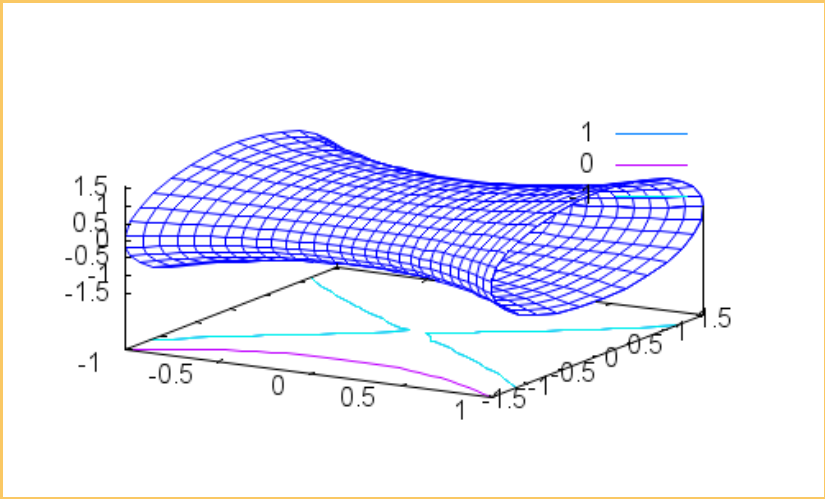
$$(t, (a\,cosh(t/a))\,cos(w), (a\,cosh(t/a))\,sen(w))$$

com $-b \leq t \leq b$ e $0 \leq w \leq 2\,\pi$. Desenhe o catenóide em torno dos eixos x e z. Desenhe curvas de níveis.

Solução:

Maxima

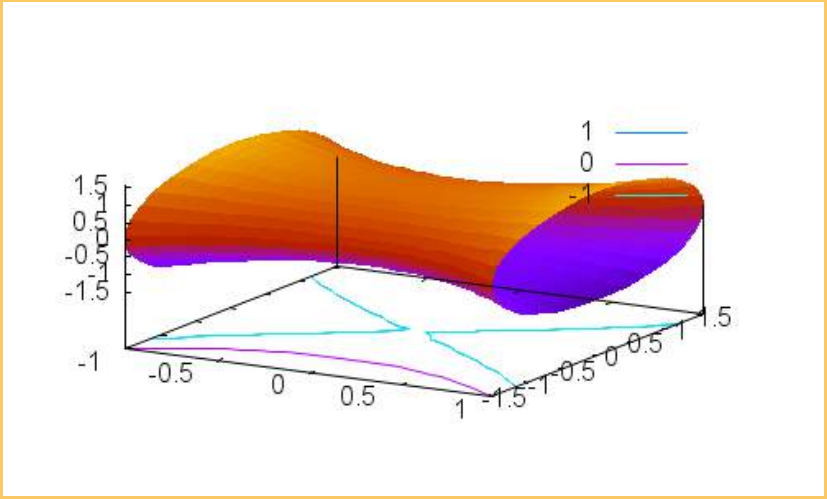
```
(%i404)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(t,cosh(t)*cos(w),cosh(t)*sin(w),
       t,-1,1,w,0,2*%pi))
```




```
(%o329)
```

Maxima

```
(%i405)wxdraw3d(enhanced3d=true,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(cosh(t)*cos(w),cosh(t)*sin(w),
       t,t,-1,1,w,0,2*%pi))
```



(%o330)

Exemplo 2.60 O Toro é uma superfície de revolução obtida girando-se um círculo de raio r e centro $(0,a,0)$ em torno do eixo z, com $a > r > 0$. Uma parametrização do Toro é

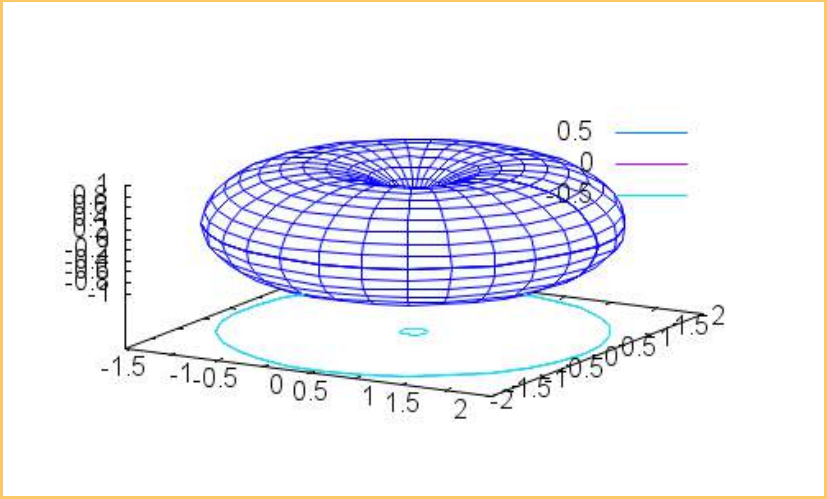
$$((a + r\,cos(v))\,cos(u), (a + r\,cos(v))\,sen(u), r\,sen(v).$$

Desenhe o Toro nos formatos grade e cheio. No formato grade, desenhe curvas de níveis.

Solução:

Maxima

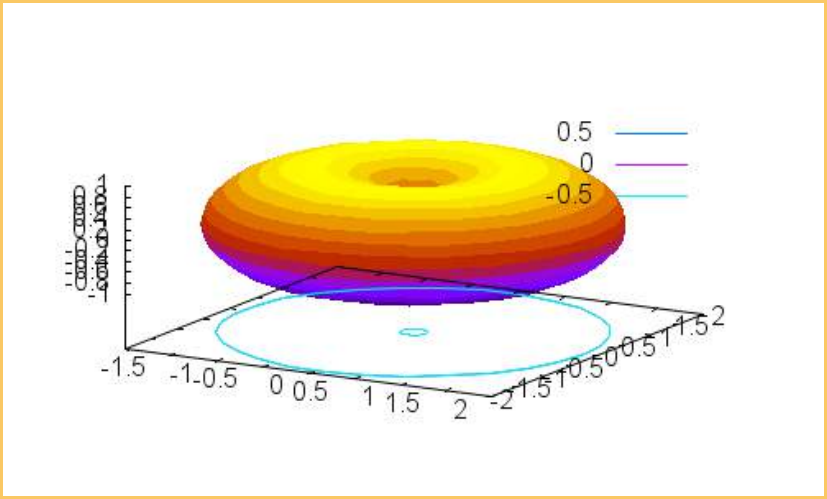
```
(%i406)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface((1+cos(v))*cos(u),
       (1+cos(v))*sin(u),sin(v),u,0,2*%pi,v,0,2*%pi))
```



(%o331)

Maxima

```
(%i407)wxdraw3d(enhanced3d=true,
       colorbox=false, surface_hide=true,
       parametric_surface((1+cos(v))*cos(u),
       (1+cos(v))*sin(u),sin(v),u,0,2*%pi,v,0,2*%pi))
```



(%o332)

Exemplo 2.61 Uma parametrização do Helicóide é

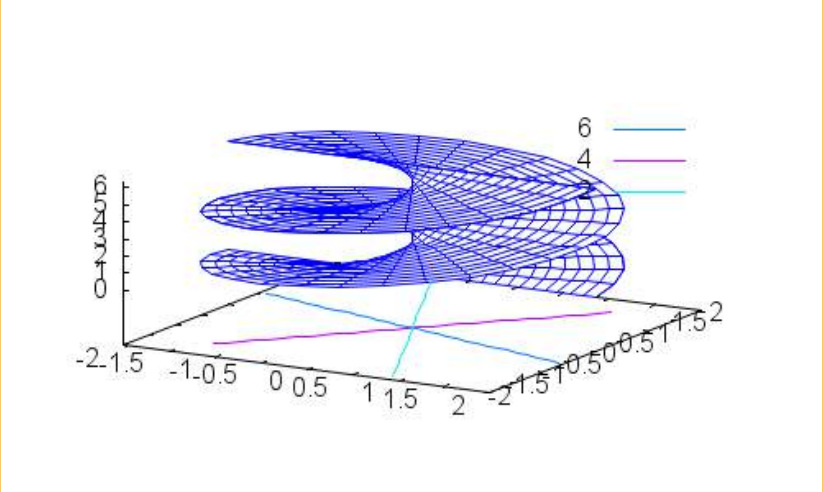
$$(u\,cos(v), u\,sen(v), v)$$

com $a \leq u \leq b$ e $0 \leq v \leq 2\pi$. Desenhe o Helicóide nos formatos grade de cheio. Desenhe curvas de níveis.

Solução:

Maxima

```
(%i408)wxdraw3d(enhanced3d=false,
       colorbox=false, surface_hide=true, contour=base,
       parametric_surface(u*cos(v),u*sin(v),v,u,-2,2,
       v,0,2*%pi))
```




```
(%o333)
```

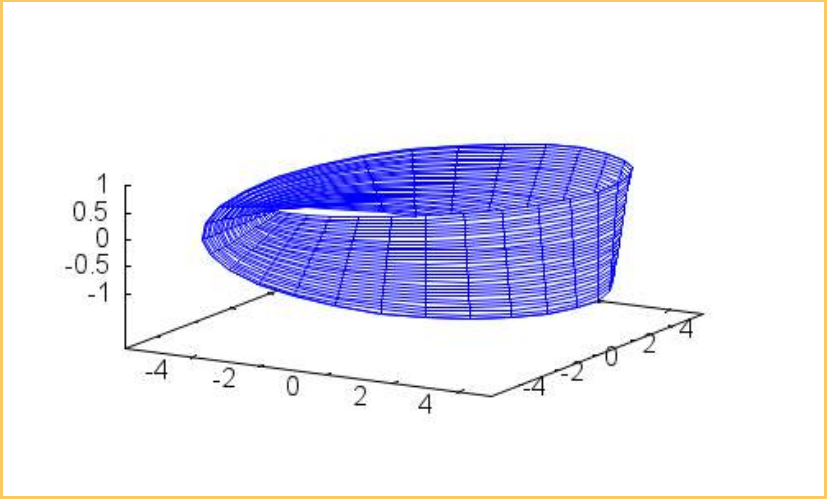
Exemplo 2.62 Uma parametrização da Faixa de Möbius é

$$((a + u\,sen(v/2))\,cos(v), (a + u\,sen(v/2))\,sen(v), u\,cos(v/2))$$

com $-a \leq u \leq a$ e $0 \leq v \leq 2\pi$. Desenhe a Faixa de Möbius.

Maxima

```
(%i409)wxdraw3d(enhanced3d=false,surface_hide=true,
       parametric_surface((5+u*sin(v/2))*cos(v),
       (5+u*sin(v/2))*sin(v),u*cos(v/2),u,-1,1,v,0,2*%pi))
```



(%o334)

2.5 Equações diferenciais ordinárias

Para resolvermos equações diferenciais ordinárias com o `Maxima` usamos o seguinte comando:

ode2(edo, vardep, varindep)

edo - equação diferencial
vardep - variável dependente
varindep - variável independente.

Quando o aplicativo resolve a EDO, retorna uma solução (explícita ou implícita) para a variável dependente e representa as constantes por “%c1", “%c2", etc. Se não pode obter a solução, retorna a palavra **false** seguida, às vezes, de uma mensagem de erro. Para obtermos o valor das constantes usamos o seguinte comando:

icj(sol, varindep=x_0, vardep=y_0)

j=1 - EDO de primeira ordem.
j=2 - EDO de segunda ordem.
sol - solução geral da EDO.
x_0 - valor inicial da variável independente.
y_0 - valor inicial da variável dependente.

Para o caso de problemas de fronteira com EDO de segunda ordem, obtemos o valor das constantes usando o seguinte comando:

bc2(sol, frontx=x_1, fronty=y_1, frontx=x_2, fronty=y_2)

sol - solução geral da EDO
x_1, x_2 - valores de fronteira da variável independente
y_1, y_2 - valores fronteira da variável dependente.

Exemplo 2.63 Encontre a solução geral das seguintes equações diferenciais lineares:

(a) $y' + a\,y = 0$;

(b) $y' + a\,y = f(x)$;

(c) $y' + a(x)\,y = f(x)$.

com a uma constante não nula e $f(x)$ uma função dada.

Solução: (a)

Maxima

```
(%i410)eq1:'diff(y,x)+a*y=0;
```

$$(\%o335)\quad \frac{d}{dx}y + a\,y = 0$$

```
(%i411)sol1:ode2(eq1,y,x);
```

$$(\%o336)\quad y = \%c\,\%e^{-a\,x}$$

Solução: (b)

Maxima

```
(%i412)depends(f,x);
```

$$(\%o337)\quad [f(x)]$$

```
(%i413)eq2:'diff(y,x)+a*y=f(x);
```

$$(\%o338)\quad \frac{d}{dx}y + a\,y = f(x)$$

```
(%i414)sol2:ode2(eq2,y,x);
```

$$(\%o339)\quad y = \%e^{-a\,x}\left(\int \%e^{-a\,x} f(x)\,dx + \%c\right)$$

Solução: (c)

Maxima

```
(%i415)depends(f,x);
```

$$(\%o340)\quad [f(x)]$$

```
(%i416)depends(a,x);
```

$$(\%o341)\quad [a(x)]$$

```
(%i417)eq03:'diff(y,x)+a(x)*y=f(x);
```

$$(\%o342)\quad \frac{d}{dx}y + a(x)\,y = f(x)$$

```
(%i418)sol03:ode2(eq03,y,x);
```

$$(\%o343)\quad y = \%e^{\int a(x)\,dx}\left(\int \%e^{-\int a(x)\,dx} f(x)\,dx + \%c\right)$$

Exemplo 2.64 Considere a equação

$$\frac{dy}{dx} - 3y = -e^{-x}.$$

(a) Ache a solução geral;

(b) Ache a solução $y(x)$ que satisfaz a condição inicial $y(0) = 0$. Esboce o gráfico.

Solução: (a)

Maxima

```
(%i419)eq3:'diff(y,x)-3*y=exp(-x);
```

(%o344) $\frac{d}{dx}y - 3y = \%e^{-x}$

```
(%i420)sol3:ode2(eq3,y,x);
```

(%o345) $y = \left(\%c - \frac{\%e^{-4x}}{4}\right)\%e^{3x}$

Solução: (b)

Maxima

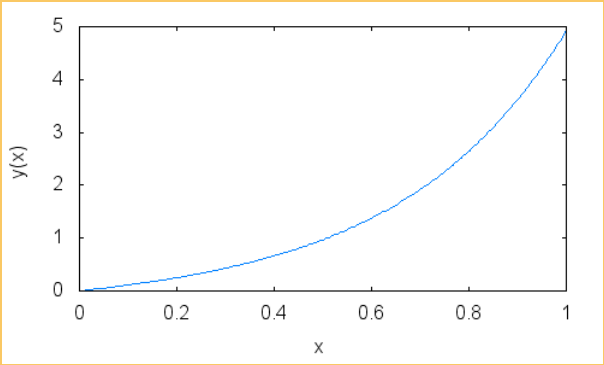
```
(%i421)solP:ic1(sol3,x=0,y=0);
```

(%o346) $y = \frac{\%e^{-x}(\%e^{-4x} - 1)}{4}$

```
(%i422)y(x):=''rhs(solP);
```

(%o347) $y(x) := rhs(solP)$

```
(%i423)wxplot2d(y(x),[x,0,1],[y,0,5],[ylabel, "y(x)"]);
```



(%o348)

Exemplo 2.65 Resolva a equação

$$\frac{dy}{dx} + 2xy = x.$$

Solução:

Maxima

```
(%i424)eq04:'diff(y,x)2*x*y=x;
```

(%o349) $\frac{d}{dx}y + 2y = x$

```
(%i425)sol04:ode2(eq04,y,x);
```

(%o350) $y = \%e^{-x^2}\left(\frac{\%e^{x^2}}{2} + \%c\right)$

Exemplo 2.66 Considere a seguinte equação

$$\frac{dy}{dx} + 2xy = xy^2$$

chamada **equação de Bernoulli**.

(a) Ache a solução geral;

(b) Ache a solução que satisfaz $y(0) = 1$.

Solução: (a)

Maxima

```
(%i426)eq02:'diff(y,x)+2*x*y=x*y^2;
```

$$(\%o351)\quad \frac{d}{dx}y + 2y = xy^2$$

```
(%i427)sol05:ode2(eq05,y,x);
```

$$(\%o352)\quad -\frac{\log(y+2)-\log(y)}{2} = \frac{x^2}{2} + \%c$$

Solução: (b)

Maxima

```
(%i428)ic1(sol05,x=0,y=1);
```

$$(\%o353)\quad -\frac{\log(y+2)-\log(y)}{2} = \frac{x^2-\log(3)}{2}$$

Exemplo 2.67 Encontre a solução $y(x)$ do seguinte problema de valor inicial

$$\begin{cases} (x-1)y^3 + (y-1)x^3y' = 0, \\ y(3) = 2 \end{cases}$$

Solução:

Maxima

```
(%i429)eq:(x-1)*y^3+(y-1)*x^3*'diff(y,x)=0;
```

$$(\%o354)\quad x^3(y-1)\frac{d}{dx}y + (x-1)y^3 = 0$$

```
(%i430)ode2(eq,y,x);
```

$$(\%o355)\quad \frac{2y-1}{2y^2} = \%c - \frac{2x-1}{2x^2}$$

```
(%i431)ic1(%o54,x=2,y=-3);
```

$$(\%o356)\quad \frac{2y-1}{2y^2} = -\frac{x^2+72x-36}{72x^2}$$

Exemplo 2.68 Encontre a solução geral da equação diferencial de 2a ordem com coeficientes constantes dada por

$$y'' + b\,y' + c\,y = 0.$$

Solução:

Maxima

```
(%i432)eq4:'diff(y,x,2)+b*'diff(y,x)+c*y=0;
```

$$(\%o357)\quad \frac{d^2}{dx^2}y + b\left(\frac{d}{dx}y\right) + c\,y = 0$$

```
(%i433)sol4:ode2(eq4,y,x);
  Is 4 c- b^2 positive, negative, or zero?positive
```

$$(\%o358)\quad y = \%e^{-\frac{bx}{2}}\left(\%k1\sin\left(\frac{\sqrt{4c-b^2}\,x}{2}\right) + \%k2\cos\left(\frac{\sqrt{4c-b^2}\,x}{2}\right)\right)$$

```
(%i434)sol4:ode2(eq4,y,x);
  Is 4 c- b^2 positive, negative, or zero?negative
```

$$(\%o359)\quad y = \%k1\%e^{\frac{(\sqrt{b^2-4}-b)x}{2}} + \%k2\%e^{\frac{(-\sqrt{b^2-4}c-b)x}{2}}$$

```
(%i435)sol4:ode2(eq4,y,x);
  Is 4 c- b^2 positive, negative, or zero?zero
```

$$(\%o360)\quad y = (\%k2\,x + \%k1)\%e^{-\frac{bx}{2}}$$

Exemplo 2.69 Considere a equação

$$y'' + 2\,y' + 2\,y = 0.$$

(a) Ache a solução geral;

(b) Esboce o gráfico da solução que satisfaz as condições iniciais $y(0) = 0$ e $y'(0) = 1$.

Solução:

Maxima

```
(%i436)eq5:'diff(y,x,2)+2*'diff(y,x)+2*y =0;
```

$$(\%o361)\quad \frac{d^2}{dx^2}y + 2\left(\frac{d}{dx}y\right) + 2\,y = 0$$

```
(%i437)sol5:ode2(eq5,y,x);
```

(%o362) $y = \%e^{-x}(\%k1\,sin(x) + \%k2\,cos(x))$

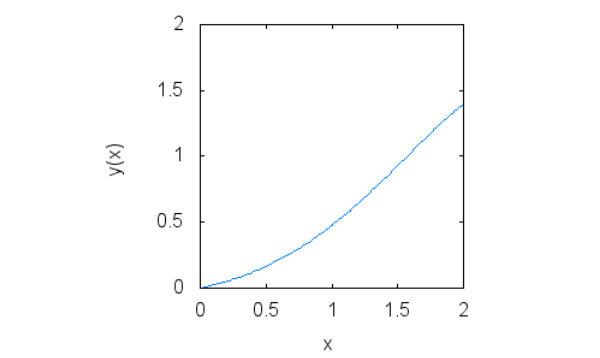
```
(%i438)solP:ic2(sol5,x=0,y=0,'diff(y,x)=1);
```

(%o363) $y = \%e^{-x}\,sin(x)$

```
(%i439)y(x):=''rhs(solP);
```

(%o364) $y(x) := rhs(solP)$

```
(%i440)wxplot2d(y(x),[x,-3,3],[y,-8,2],[ylabel, "y(x)"],
       [gnuplot_preamble,"set size ratio 1; set zeroaxis;"]);
```



(%o365)

Exemplo 2.70 Ache a solução geral da equação

$$y'' + 3y' + 2y = x.$$

Maxima

```
(%i441)eq6:'diff(y,x,2)+3*'diff(y,x)+2*y =x;
```

(%o366) $\frac{d^2}{dx^2}y + 3\left(\frac{d}{dx}y\right) + 2y = x$

```
(%i442)sol6:ode2(eq6,y,x);
```

(%o367) $y = \%k1\,\%e^{-x} + \%k2\,\%e^{-2x} + \frac{2x-3}{4}$

Exemplo 2.71 Ache a solução da equação

$$y'' - 2y' + 2y = 0$$

que satisfaz as condições de fronteira $y(0) = 0$ e $y(\pi/2) = 1$. Esboce o gráfico da solução.

Solução:

Maxima

```
(%i443)eq7:'diff(y,x,2)-2*'diff(y,x)+2*y =0;
```

(%o368) $\frac{d^2}{dx^2}y - 2\left(\frac{d}{dx}y\right) + 2y = 0$

```
(%i444)sol7:ode2(eq7,y,x);
```

(%o369) $y = \%e^x\,(\%k1\,\sin(x) + \%k2\,\cos(x))$

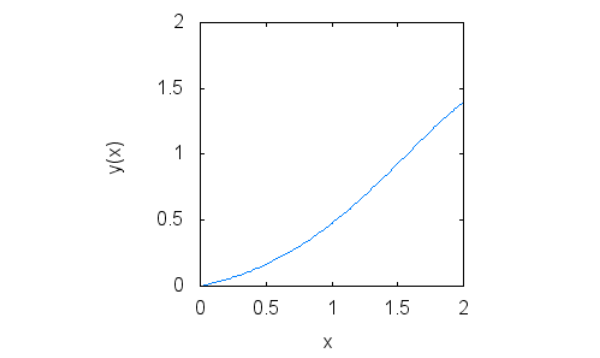
```
(%i445)solP02:bc2(sol7,x=0,y=0,x=\%pi/2,y=1);
```

(%o370) $y = \%e^{x-\frac{\pi}{2}}\,\sin(x)$

```
(%i446)y(x):=''rhs(solP02);
```

(%o371) $y(x) := rhs(solP02)$

```
(%i447)wxplot2d(y(x),[x,0,2],[y,0,2],[ylabel, "y(x)"],
       [gnuplot_preamble,"set size ratio 1; set zeroaxis;"]);
```



(%o372)

Exemplo 2.72 Encontre a solução $y(x)$ do seguinte problema de valor inicial

$$\begin{cases} (x-1)y^3 + (y-1)x^3 y' = 0; \\ y(3) = 2. \end{cases}$$

Solução;

Maxima

```
(%i448)eq8:(x-1)*y^3+(y-1)*x^3*'diff(y,x)=0;
```

(%o373) $x^3(y-1)\frac{d}{dx}y + (x-1)y^3 = 0$

```
(%i449)sol8:ode2(eq8,y,x);
```

(%o374) $\frac{2y-1}{2y^2} = \%c - \frac{2x-1}{2x^2}$

```
(%i450)ic1(sol8,x=2,y=-3);
```

(%o375) $\frac{2y-1}{2y^2} = -\frac{x^2+72x-36}{72x^2}$

Exemplo 2.73 Encontre a solução $y(x)$ do seguinte problema de fronteira

$$\begin{cases} y''(x) + y(y')^3 = 0; \\ y(0) = 1; \\ y(1) = 3. \end{cases}$$

Solução:

Maxima

```
(%i451)eq:'diff(y,x,2) + y*'diff(y,x)^3 = 0;
```

(%o376) $\frac{d^2}{dx^2}y + y\left(\frac{d}{dx}y\right)^3 = 0$

```
(%i452)sol:ode2(eq,y,x);
```

(%o377) $\frac{y^3 + 6\%k1\,y}{6} = x + \%k2$

```
(%i453)bc2(sol,x=0,y=1,x=1,y=3);
```

(%o378) $\frac{y^3 - 10y}{6} = x - \frac{3}{2}$

Exemplo 2.74 Calcule $y'(x)$ para $y(x)$ dada implicitamente por $y^4 + x^2y^2 + x^4 = 3$.

Solução: Como a solução $y(x)$ é dada implicitamente, derivaremos a equação usando o comando **diff** e depois resolveremos a equação resultante usando o comando **solve**.

Maxima

```
(%i454)F:y^4+x^2*y^2+x^4-3=0;
```

(%o379) $y^4 + x^2 * y^2 + x^4 - 3 = 0$

```
(%i455)depends(y,x);
```

(%o380) $[y(x)]$

```
(%i456)diff(F,x);
```

(%o381) $4y^3\frac{d}{dx}y + 2x^2y\frac{d}{dx}y + 2xy^2 + 4x^3 = 0$

```
(%i457)solve(%o60,diff(y,x));
```

(%o382) $[\frac{d}{dx}y = -\frac{xy^2 + 2x^3}{2 * y^3 + x^2y}]$

2.5.1 Campo de direções

Muita informação importante sobre a equação diferencial de primeira ordem pode ser obtida por uma simples análise geométrica da função $f(x,y)$.

Observe que, em cada ponto do plano (x,y), a inclinação da tangente da solução $y(x)$ é dada por $f(x,y)$. O **campo de direções** é o desenho no plano, de vetores tangentes com a inclinação descrita por $f(x,y)$.

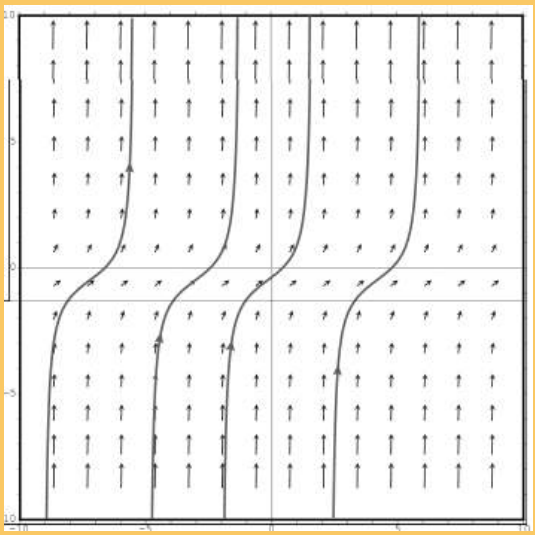
Para desenharmos campos de direção com o `Maxima` usa-se o pacote **plotdf**(f(x,y)).

Exemplo 2.75 Desenhe o campo de direções da equação $y' = 1 + y + y^2$.

Solução:

Maxima

```
(%i458)load(plotdf)$
(%i459)plotdf(1 + y + y^2);
```



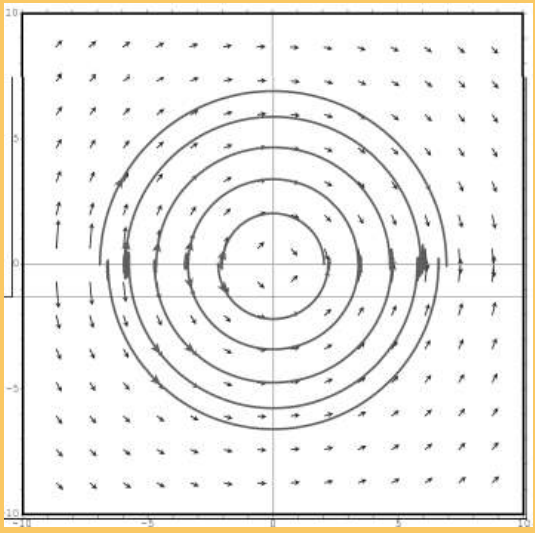
(%o383)

Exemplo 2.76 Desenhe o campo de direções da equação $y' = \frac{x}{y}$.

Solução:

Maxima

```
(%i460)load(plotdf)$
(%i461)plotdf(-x/y);
```



(%o384)

2.5.2 Sistema de equações diferenciais

Para resolvermos um sistema de equações diferenciais de primeira ordem do tipo

$$\begin{cases} y_1' = f_1(x,y); \\ y_2' = f_2(x,y); \\ \cdots \\ y_m' = f_m(x,y). \end{cases}$$

usamos o seguinte comando:

desolve([eqn1,eqn2,..., eqnm],[y_1(x), y_2(x),..., y_m(x)])

A forma de declaração das funções das equações diferenciais para o comando **desolve** devem ser explícita.

Exemplo 2.77 Encontre a solução do seguinte sistema de equações

$$\begin{cases} u' = v - z; v' = z - u; z' = u - v \\ u(0) = 1; v(0) = -1; z(0) = 1. \end{cases}$$

Maxima

```
(%i462)eqn1: 'diff(u(x),x)=v(x)-z(x);
```

(%o385) $\frac{d}{dx}u(x) = v(x) - z(x)$

```
(%i463)atvalue(u(x),x=0,1);
```

(%o386) 1

```
(%i464)eqn2:'diff(v(x),x)=z(x)-u(x);
```

(%o387) $\frac{d}{dx}v(x) = z(x) - u(x)$

```
(%i465)atvalue(v(x),x=0,-1);
```

(%o388) -1

```
(%i466)eqn3:'diff(z(x),x)=u(x)-v(x);
```

```
(%o389)  d/dx z(x) = u(x) - v(x)
(%i467)atvalue(z(x),x=0,1);

(%o390)  1
(%i468)sist:[eqn1,eqn2,eqn3];

(%o391)  [d/dx u(x) = v(x) - z(x), d/dx v(x) = z(x) - u(x), d/dx z(x) = u(x) - v(x)]
(%i469)solS1:desolve(sist,[u(x),v(x),z(x)]);

(%o392)  [u(x) = 2 sin(sqrt(3) x)/sqrt(3) + 2 cos(sqrt(3) x)/3 + 1/3, v(x) = 1/3 - 4 cos(sqrt(3) x)/3,
          z(x) = 2 sin(sqrt(3) x)/sqrt(3) + 2 cos(sqrt(3) x)/3 + 1/3]
```

Exemplo 2.78 Encontre a solução do seguinte sistema de equações

$$\begin{cases} u'' = v; v'' = u; \\ u(0) = 1; u'(0) = 2; \\ v(0) = 2; v'(0) = 2. \end{cases}$$

Maxima

```
(%i470)eqn1: 'diff(u(x),x,2)=v(x);

(%o393)  d^2/dx^2 u(x) = v(x)
(%i471)atvalue(u(x),x=0,1);

(%o394)  1
(%i472)atvalue('diff(u(x),x),x=0,2);

(%o395)  2
(%i473)eqn2:'diff(v(x),x,2)=u(x);

(%o396)  d^2/dx^2 v(x) = u(x)
(%i474)atvalue(v(x),x=0,2);

(%o397)  2
(%i475)atvalue('diff(v(x),x),x=0,2);

(%o398)  2
(%i476)sist:[eqn1,eqn2];
```

```
(%o399)  [d^2/dx^2 u(x) = v(x), d^2/dx^2 v(x) = u(x)]
(%i477)solS2:desolve(sist,[u(x),v(x)]);
(%o400)  [u(x) = -cos(x)/2 + 7 %e^x/4 - %e^x/4, v(x) = cos(x)/2 + 7 %e^x/4 - %e^x/4]
```

2.6 Integração Numérica

A Integração numérica é um conjunto de técnicas numéricas para o cálculo aproximado da integral definida da função f.

Estas técnicas são importantes nos casos, por exemplo, que a função f tem uma expressão analítica muito complicada ou é dada por uma tabela. Em geral, uma regra de integração numérica consiste em aproximar a integral definida por uma soma finita, supondo-se que f é uma função "bem comportada".

Para isto, considere uma partição $\mathcal{P} : a = x_0 < x_1 < \ldots < x_i < \ldots < x_N = b$ do intervalo de integração $[a, b]$ e a seguinte aproximação:

$$\int_a^b f(x)dx \cong \sum_{i=0}^{n} w_i f(x_i). \tag{2.2}$$

Os pontos x_i são chamados *pontos de integração* e w_i são os *pesos* da fórmula de integração. Cada escolha de x_i e w_i define uma regra de integração.

Nesta seção, descreveremos algumas regras clássicas de integração numérica e usaremos o `Maxima` nas implementações. Para maiores detalhes consulte [3].

Trapézio repetida

Considere uma partição $\mathcal{P}$ do intervalo $[a, b]$. Aplicando a regra dos trapézios em cada subintervalo $[x_{i-1}, x_i]$ obtemos

$$\int_a^b f(x)\,dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx \cong \frac{h}{2}\Big(f(x_0) + 2f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-1}) + f(x_n)\Big).$$

E o erro é dado por

$$E_{TR} = -\frac{h^3}{12}\Big(f''(\beta_1) + \cdots + f''(\beta_n)\Big)$$

com $\beta_i \in (x_{i-1}, x_i)$. Usando o teorema do valor médio para integrais tem-se

$$E_{TR} = -(b-a)\frac{h^2}{12} f''(\gamma), \qquad \gamma \in (a,b).$$

Exemplo 2.79 Usando a regra dos Trapézios repetida calcule um valor aproximado da integral $\int_0^1 e^{x^2}\, dx$.

Solução:

Maxima

```
(%i478)n:10, a:0, b:1, h:(b-a)/n$
(%i479)xx:makelist(a+i*h,i,0,n);
```

(%o401) $[0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1]$

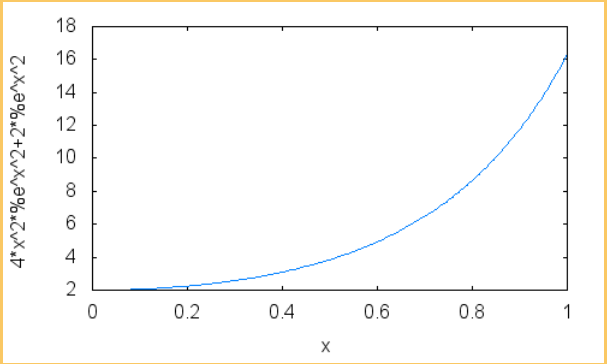
```
(%i480)f(x):=exp(x^2)$
(%i481)TR:h/2*(sum(2*f(xx[i]),i,2,n)+f(xx[1])+f(xx[n+1])),numer;
```

(%o402) 1.467174692738799

Estimativa do erro:

Maxima

```
(%i482)define(dTRf(x),abs(diff(f(x),x,2)))$
(%i483)plot2d(dTRf(x),[x,0,1]);
```



(%o403)

```
(%i484)ErroTR:(abs(b-a)*h^2*17)/12, numer;
```

(%o404) 0.014166666666667 $\Rightarrow$ |ErroTR| $\leq$ 0.014166666666667

Simpson repetida

Considere uma partição $\mathcal{P}$ do intervalo $[a,b]$. Para aplicarmos a regra de Simpson devemos subdivir o intervalo em um número *par* de subintervalos, pois cada parábola requer três pontos de interpolação. Assim,

$$\begin{aligned}\int_a^b f(x)\,dx &= \sum_{i=1}^{n/2}\int_{x_{2(i-1)}}^{x_{2i}} f(x)\,dx \\ &\cong \frac{h}{3}\Big(f(x_0)+4f(x_1)+2f(x_2)+\ldots+2f(x_{n-2})+4f(x_{n-1})+f(x_n)\Big) \\ &\cong \frac{h}{3}\left[f(x_0)+4\sum_{i=1}^{n/2}f(x_{2i-1})+2\sum_{i=1}^{(n-2)/2}f(x_{2i})+f(x_n)\right]\end{aligned}$$

E o erro é dado por

$$E_{SR} = -(b-a)\frac{h^4}{180}f^{(4)}(\gamma), \qquad \gamma\in(a,b).$$

Exemplo 2.80 Usando a regra de Simpson repetida calcule um valor aproximado da integral $\int_0^1 e^{x^2}\,dx$.

Solução:

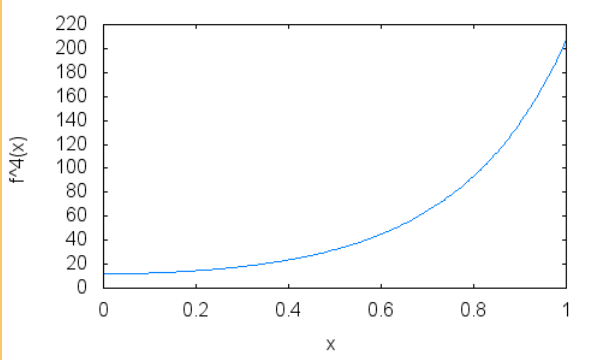
Maxima

```
(%i485)n:10; a:0; b:1; h:(b-a)/n;$
(%i486)xx:makelist(a+i*h,i,0,n);
(%o405)  [0, 1/10, 1/5, 3/10, 2/5, 1/2, 3/5, 7/10, 4/5, 9/10, 1]
(%i487)f(x):=exp(x^2)$
(%i488)SR:h/3*(f(xx[1])+4*sum(f(xx[2*i]),i,1,n/2)+
       2*sum(f(xx[2*i+1]),i,1,(n-2)/2)+f(xx[n+1])), numer;
(%o406)  1.462681400099797
```

Estimativa do erro:

Maxima

```
(%i489)define(dSRf(x),abs(diff(f(x),x,2)))$
(%i490)plot2d(dSRf(x),[x,0,1]);
```



(%o407)

```
(%i491)ErroSR:(abs(b-a)*h^4*220)/180, numer;
```

(%o408) $1.222222222222222\,10^{-4} \Rightarrow |\text{ErroSR}| \leq 0.0001222222$

Fórmulas gaussianas

Fórmulas gaussiana são regras de integração cujos pontos não são igualmente $\{x_0, x_1, \ldots, x_n\}$ não são necessariamente igualmente espaçados. Serão consideradas aqueles exatas para polinômios de grau menor ou igual a $2n+1$.

As fórmulas gaussinas mais conhecidas são as chamadas *Gauss-Legendre* por estarem associadas aos polinômios de Legendre

$$\psi_0(x) = 1, \quad \psi_1(x) = x, \quad \psi_2(x) = x^2 - \frac{1}{3}, \quad \ldots$$

Para obtermos a fórmulas gaussiana de dois pontos vamos considerar as raízes de $\psi_2(x) = x^2 - \frac{1}{3}$, ou seja, $x_0 = -\sqrt{\frac{1}{3}}$ e $x_1 = \sqrt{\frac{1}{3}}$. Para obtermos os pesos w_0 e w_1 devemos garantir que a fórmula

$$\int_{-1}^{1} f(x)dx \cong w_0 f(x_0) + w_1 f(x_1)$$

seja exata para polinômios de grau ≤ 1, ou seja, $f(x) = 1$ e $f(x) = x$. Assim,

$$\int_{-1}^{1} 1\,dx = w_0 + w_1 \quad \Rightarrow \quad w_0 + w_1 = 2$$

$$\int_{-1}^{1} x\,dx = -\frac{w_0}{\sqrt{3}} + \frac{w_1}{\sqrt{3}} \quad \Rightarrow \quad w_0 - w_1 = 0.$$

Logo, $w_0 = w_1 = 1$ e a *fórmula Gauss-Legendre para dois pontos* é dada por

$$\int_{-1}^{1} f(x)dx = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

que é exata para polinômios de grau ≤ 3.

Analogamente, *fórmula Gauss-Legendre para três pontos* é dada por

$$\int_{-1}^{1} f(x)dx = \frac{1}{9}\Big(5f(-\sqrt{0.6}) + 8f(0) + 5f(\sqrt{0.6})\Big).$$

que é exata para polinômios de grau ≤ 5.

Para usarmos as fórmula gaussinas no intervalo $[a,b]$ fazemos a seguinte mudança de variável:

$$t = \frac{1}{2}\Big((b-a)x + (b+a)\Big).$$

Assim,

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{b+a}{2}\right)dx \cong \frac{b-a}{2}\sum_{i=0}^{n} w_i f\left(\frac{b-a}{2}x + \frac{b+a}{2}\right).$$

Exemplo 2.81 Usando a regra gaussiana calcule um valor aproximado da integral $\int_0^1 e^{x^2}\,dx$.

Solução:

Maxima

```
(%i492)Gauss:0.5*(exp(0.25*(-1/sqrt(3)+1)^2) +
       exp(0.25*(1/sqrt(3)+1)^2)),numer;
(%o409)  1.45416788923913
(%i493)quad_qag(f(x),x,0,1,1);
(%o410)  [1.462651745907182,1.619055973138259 10^-13,15,0]
```

Programas do Maxima para integração numérica

Podemos também usar os programas do Maxima para calcularmos numericamente a integral de uma função. Por exemplo, podemos usar o seguinte comando:

quad_qags(f(x),x,a,b, opções)	-	calcula numericamente a integral de f(x), usando regras do tipo de quadratura Gaussiana

O comando **quad_qags** retorna uma lista de quatro elementos:

- uma aproximação para o valor da integral;
- o erro absoluto estimado da aproximação;
- o número de avaliações do integrando;
- um código de erro.

O código de erro são os seguintes:

- 0 - nenhum problema foi encontrado;
- 1 - foram utilizados muitos subintervalos;
- 2 - foi detectato um erro de arredondamento excessivo;
- 3 - o integrando se comportar muito mal;
- 6 - a entrada não é válida.

As "opções" do comando **quad_qags** são as seguintes:

ordem: é um inteiro m que define a ordem da regra de integração; varia em $\{1, 2, 3, 4, 5, 6\}$.

erro: erro relativo desejado; padrão entre -1 à -8;

particao: número máximo de subintervalos;padrão 200.

Para ilustrar o comando **quad_qags**, tentaremos calcular a integral

$$\int_0^1 \text{arctg}(x)^2\,dx$$

com o comando **integrate**:

Maxima

```
(%i494)integrate(atan(x)^2,x,0,1);
```

$$(\%o411)\quad \int_0^1 atan(x)^2\,dx$$

A resposta do aplicativo (retorno da integral definida) indica que o comando **integrate** não é capaz de calcular uma primitiva da função $f(x) = \text{arctg}^2(x)$. Vamos obter um valor aproximado usando o comando **quad_qags**.

Maxima

```
(%i495)quad_qags(atan(x)^2,x,0,1);
```

$$(\%o412)\quad [0.24528120346677, 2.723168395965719910^{-15}, 21, 0]$$

Observação 2.2 *Os comandos* **quad_qapi** *e* **romberg** *também são programas para calcular numericamente a integral de uma função.*

3 Fractais

3.1 Elementos da Geometria Fractal

Benoit Mandelbrot, matemático polonês, criou este termo em 1975 para designar conjuntos com características muito especiais. O termo vem do latim, do adjetivo *fractus*, derivado do verbo *frangere* que significa *quebrar, fracionar*.

Muitos matemáticos referem-se aos fractais como *conjuntos de dimensão não inteira*, *conjuntos de medida Hausdorff*, *conjuntos com estrutura fina* ou simplesmente *conjuntos irregulares*.

Algumas propriedades caracterizam as figuras que chamaremos de **fractais**. Sabemos que uma figura é um conjunto de pontos. O círculo é um conjunto de pontos do plano, a esfera é um conjunto de pontos no espaço tridimensional. Estas figuras são estudadas na geometria clássica. Porém, os fractais são figuras com propriedades e características bem peculiares, que os diferenciam das figuras geométricas clássicas. Tais figuras são construídas, em geral, por processos iterativos e, seguindo Kennet Falconer [5], classificaremos uma figura como um fractal quando apresentar todas ou a maiorias das seguintes propriedades:

Lei de formação simples e complexidade

O processo de construção de um fractal é geralmente iterativo ou recorrente, porém relativamente simples. Porém, a simplicidade do processo não é suficiente para descre- ver o fractal como um lugar geométrico de pontos que satisfazem uma propriedade simples ou que são soluções de equações simples.

Os fractais podem ser construídos por equações simples, mas não podem ser descritos analiticamente por uma equação, como, por exemplo, a reta e a circunferência.

Estrutura fina em qualquer escala

A estrutura fina é a propriedade da "complexiadade em escalas", ou seja, se sucessivas ampliações da figura forem feitas, mais detalhes sobre a mesma serão des-

cobertos.

As figuras geométricas clássicas não possuem esta estrutura. Por exemplo, ao examinarmos o gráfico da função $f(x) = sen\, x$ no intervalo $[0, 2\pi]$, observaremos uma série de detalhes tais como pontos de máximo e mínimo, raízes, pontos de inflexão, curvatura, etc. Agora, se restringirmos o estudo ao intervalo $[0, 1]$, perceberemos somente uma ligeira curvatuta e nada mais. Se reduzirmos ainda mais o diâmetro do intervalo, digamos a apenas alguns milésimos de radiano, observaremos que o gráfico (da senóide) se assemelha ao gráfico de uma reta, e a reta é uma "figura sem detalhes". Portanto, à medida em que a região de investigação é reduzida, o detalhamento tende a desaparecer e então podemos concluir que a senóide não tem a propriedade de estrutura fina.

Dimensão topológica e dimensão fractal

As figuras geométricas clássicas têm dimensão fractal igual à dimensão topológica, mas os fractais têm, em geral, dimensão fractal estritamente menor que a sua dimensão topológica.

Autossimilaridade ou auto-afinidade

A autossemelhança ou autossimilaridade é uma ideia antiga e uma propriedade geométrica simples. É a "simetria através das escalas", ou seja, um objeto possui autossemelhança se apresenta a mesma forma em qualquer escala em que seja observado. Naturalmente, nem todos os objetos geométricos têm esta propriedade. Por exemplo, um círculo numa escala muito grande não é nada mais do que uma reta. Por outro lado, um quadrado é um conjunto autossimilar do plano, pois pode ser formado por quatro cópias deles mesmo reduzidas por um fator r.

Matematicamente, a propriedade de autossimilaridade é descrita por transformações cujas imagens dos objetos sejam "cópias reduzidas" do mesmo e "reconstituam" integralmente o mesmo objeto. Tais transformações são chamadas **transformações similares**.

A auto-afinidade é um tipo de autossimilaridade mais geral. Uma figura é auto-afim se pode ser decomposta em partes menores que são não exatamente iguais, mas apenas modificadas por transformações afins tais como: contrações, dilatações, translações, rotações, reflexões ou combinações dessas transformações. Por exemplo, se uma figura é reduzida uniformemente em todas as direções formará uma réplica geometricamente semelhante à figura original. Um círculo que é reduzido

deste modo formará um círculo menor. Porém, se a contração se efetuar uniformemente em uma só direção, resultará em uma elipse. Um retângulo só será *similar* a outro retângulo que possua mesma razão entre o comprimento da base e sua altura, enquanto será *afim* a qualquer paralelogramo.

3.2 Sistema de funções iteradas

Nesta seção descreveremos as principais propriedades matemáticas dos fractais. Para mais detalhes consulte [4, 5].

Definição 3.1 *Seja $r > 0$ um número real, dizemos que a transformação $f : X \to X$ é uma "transformação similar" se*

$$\|f(\vec{x}) - f(\vec{y})\| = r\|\vec{x} - \vec{y}\|, \quad \forall \vec{x}, \vec{y} \in \mathbb{R}^2.$$

Se $r < 1$, f é chamada uma "contração".

Exemplo 3.1 Considere um triângulo equilátero Δ de lado igual a 1 e a transformação que reduz Δ de tal modo que cada lado seja a metade do anterior. Então, $f : \Delta \to \mathbb{R}^2$ é dada por

$$f(x, y) = \frac{1}{2}(x, y).$$

Observe que, se aplicarmos a transformação f infinitamente temos que os triângulos serão reduzidos de modo que seus lados serão cada vez menores, ou seja, a sequência de triângulos tenderá a um único ponto.

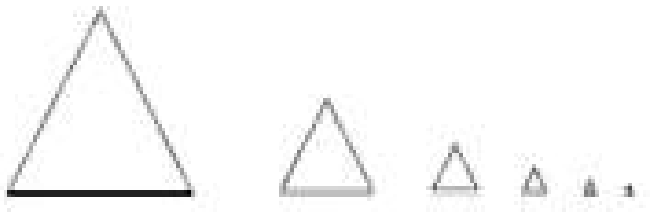


Figura 3.1: Redução

Definição 3.2 *Dizemos que dois conjuntos são autossimliares se um é a imagem do outro por transformações similares. O conjunto E é* **autosimilar** *se é formado pela união de* **m** *imagens similares de si mesmo, ou seja,*

$$E = \bigcup_{k=1}^{m} f_k(E). \tag{3.1}$$

Um conjunto autosimilar definido por (3.1) *é chamado* **atrator** *ou* **conjunto invariante** *do* **sistema de funções iteradas-IFS** $\{f_1, f_2, \ldots, f_m\}$ *com* f_k *funções similares com raios* $r_1, r_2, \ldots, r_m$.

A teoria das IFS (veja, [5]) garante que se o espaço $(X, \|\cdot\|)$ é completo e $\{f_1, f_2, \ldots, f_m\}$ são contrações então existe um único compacto não vazio dado por (3.1). Deste modo, temos um processo iterativo de construção do atrator E: dado qualquer compacto não vazio inicial K_0, para $\forall n$, considere

$$K_{n+1} = \bigcup_{k=1}^{m} f_k(K_n).$$

Então, a sequência $\{K_n\}$ converge para E (num certo sentido). Os seguintes resultados garantem estas afirmações:

Teorema 3.1 *Considere X um espaço métrico completo e* $\mathbb{K}(X)$ *a família de subconjuntos compactos de X. Se* $f_k : X \to X$ *são contrações com fatores de contração* $r_i \in (0,1)$, $i = 1, 2, \ldots, m$. *Então a aplicação* $F : \mathbb{K}(X) \to \mathbb{K}(X)$ *definida por*

$$F(K) = \bigcup_{k=1}^{m} f_k(K)$$

tem um único atrator $K = F(K)$.

O atrator dado no Teorema 3.1 pode ser obtido pelo método das sucessivas aproximações. De fato.

Corolário 3.1 *Dado qualquer* $K_0 \in \mathbb{K}(X)$. *Se*

$$K_{n+1} = F(K_n) = \bigcup_{i=1}^{m} f_i(K_n) \tag{3.2}$$

para $n \geq 0$, *então a sequência* (K_n) *converge, na métrica Hausdorff, para o atrator do* IFS $\{f_1, f_2, \ldots, f_m\}$.

O Corolário 3.1 justifica o fato que chamarmos o ponto fixo K de F de atrator.

Agora, também podemos justificar o uso do termo "sistema de funções iteradas". A iteração tratada aqui é, por exemplo, aquela que dado qualquer $x_0 \in \mathbb{R}^n$ aplicamos repetidamente as funções f_k em qualquer ordem. Mais especificamente, sejam $x_0 \in \mathbb{R}^n$ e (x_n) uma sequência definida por

$$x_n = f_{k_n}(x_{n-1})$$

para $n \geq 1$. Então,

(i) todo ponto de acumulação da sequência (x_n) pertence ao atrator K de F;

(ii) todo ponto do atrator K é um ponto de acumulação de cada sequência (x_n) para alguma escolha $k \in \{1, 2, \ldots, m\}$;

(iii) existe um ponto x_0 e uma escolha de $k \in \{1, 2, \ldots, m\}$ tais que K é igual ao conjunto de todos os pontos de acumulação de (x_n).

Uma versão mais sofisticada do resultado (iii) diz que uma escolha "aleatória" de $k \in \{1, 2, \ldots, m\}$ (com probabilidade 1) terá como fecho o conjunto K (para mais detalhes consulte "jogo do caos" em [4, 5]).

Podemos construir um objeto autossimilar reduzindo (ou ampliando), rodando e/ou transladando este objeto. Por exemplo, considere um quadrado $\mathcal{Q}$ unitário, se queremos reduzir este quadrado a metade (fator de redução $1/2$) devemos "transformar" todos os pontos deste quadrado em novos pontos cujas coordenadas serão a metade das coordenadas dos pontos originais (veja figura 3.2). Se queremos transladar $\mathcal{Q}$ duas unidades a direita da sua posição inicial devemos "transformar" todos os pontos de $\mathcal{Q}$ em novos pontos cujas coordenadas serão as coordenadas originais mais duas unidades. No que segue, descreveremos estas transformações.

Por exemplo, se $P = (x, y)$ é um ponto de $\mathcal{Q}$, para reduzirmos $\mathcal{Q}$ a metade transformaremos $P = (x, y)$ no ponto $P_1 = (x_1, y_1)$ tal que $x_1 = \dfrac{x}{2}$ e $y_1 = \dfrac{y}{2}$. Podemos representar esta transformação do seguinte modo: $f_1(x, y) = \left(\dfrac{x}{2}, \dfrac{y}{2}\right)$.

O mesmo vale para a translação. Transformaremos o ponto $P = (x, y)$ no ponto $P_1 = (x_1, y_1)$ tal que $x_1 = x + 2$ e $y_1 = y$. Esta transformação tem a seguinte representação: $f_2(x, y) = (x + 2, y)$.

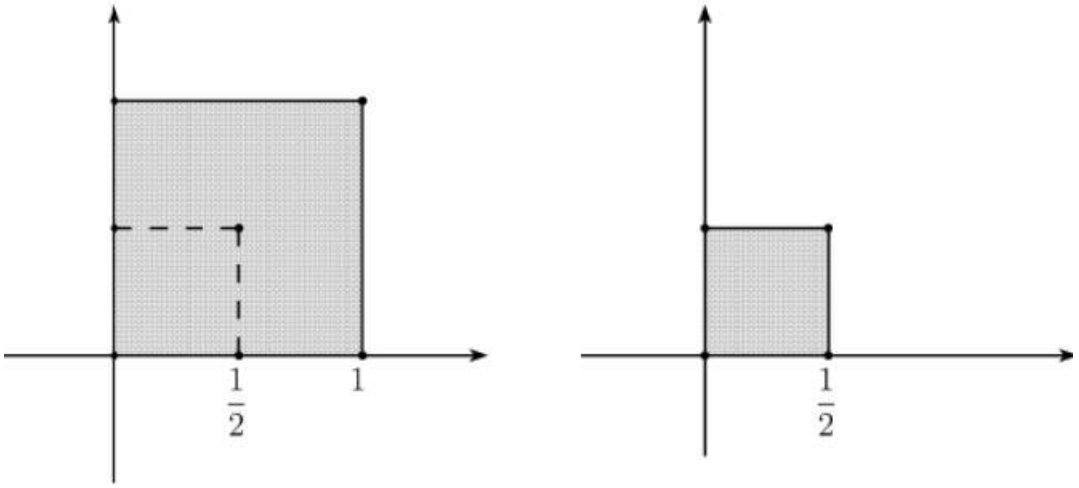


Figura 3.2:

Agora, se quisermos rodar $\mathcal{Q}$ um ângulo $\theta°$ no sentido anti-horário, cada ponto $P = (x, y)$ de $\mathcal{Q}$ deve ser transformado no ponto $P_1 = (x_1, y_1)$ tal que $x_1 = x\cos\theta - y\,\mathrm{sen}\theta$ e $y_1 = x\,\mathrm{sen}\theta + y\cos\theta$ (veja figura 3.3). Esta transformação tem a seguinte representação: $f_3(x, y) = (x\cos\theta - y\,\mathrm{sen}\theta, x\,\mathrm{sen}\theta + y\cos\theta)$.

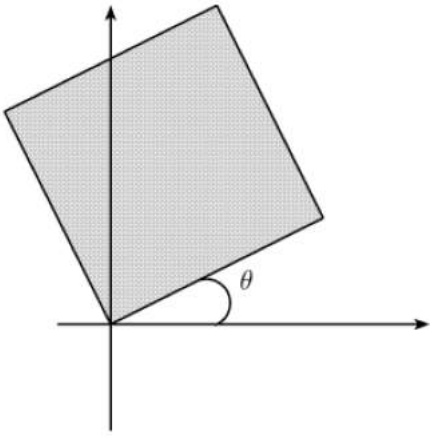


Figura 3.3:

Observe que as transformações acima podem ser representadas na seguinte forma matricial:

$$f(x, y) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

De fato,

$$f_1(x,y) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

$$f_2(x,y) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix},$$

$$f_3(x,y) = \begin{bmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

3.3 Fractais com Maxima

Nesta seção descreveremos alguns fractais bem conhecidos, os quais implementaremos com o aplicativo Maxima. Estes fractais podem ser implementados usando-se os seguintes processos: processo geométrico ou de remoção, processo aleatório ou jogo do caos e processo de autossimilares.

Jogo do caos Escolhe-se os vértices do conjunto da etapa inicial da construção do fracatl, re- presentado por $\{(p_1,q_1),\ldots,(p_m,q_m)\}$. Desenha-se o ponto inicial (x_0,y_0) e escolhe-se aleatoriamente um dos pontos de $\{(p_1,q_1),\ldots,(p_m,q_m)\}$. Então, desenha-se um novo ponto que estará no segmento entre o último ponto desenhado e o ponto que foi seleccionar aleatoriamente, a uma distância do ponto selecionado. Esta distância será β vezes o comprimento do segmento, com β o fator de redução do fractal. Repete-se novamente o processo, um número grande de iterações.

O pacote **dynamics** do aplicativo Maxima tem o comando **chaosgame** que implementa o **jogo do caos**.

Fractais por autossimilaridade Podemos construir fractais usando as transformações autossimilares do fractal. Estas transformações são do tipo

$$f_k(x,y) = \begin{bmatrix} a_{11}^k & a_{12}^k \\ a_{21}^k & a_{22}^k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1^k \\ b_2^k \end{bmatrix}$$

para $k = 1,2,3,\ldots,m$.

Assim, cada fractal é definido por m matrizes $\mathbf{A}_k$=$[a_{ij}^k]$ e m vectores $\mathbf{b}_k$=$[b_i^k]$. E

neste caso, o jogo do caos é como segue: escolhido um ponto inicial (x_0, y_0) aleatoriamente, aplica-se, também aleatoriamente, uma transformações autossimilares em (x_0, y_0) para gerar o ponto (x_1, y_1) e repete-se o processo. Deste modo, na n-ésima iteração teremos gerado (x_n, y_n) pontos do fractal ou muito próximos dos fractal. Como o fractal é o conjunto atrator destas transformações, independente do ponto inicial e para um número muito grande de iterações, o resultado final será uma figura muito próxima ao fractal.

Para controlar melhor a densidade de pontos em diferentes partes do fractal, convém que algumas das transformações autossimilares sejam escolhidas com maior freqüência. Para isso usa-se diferentes probabilidades para a escolha de cada transformação.

O comando **ifs**, no pacote **dynamics** do aplicativo `Maxima`, implementa a construação de fractais por autossimilaridade. Os parâmetros de entrada do programa são uma lista de probabi- lidades, uma lista de matrizes $\mathbf{A}_k$, uma lista de vetores $\mathbf{b}_k$, um ponto inicial (x_0, y_0) e o número de iterações que deverão ser realizadas.

O pacote **fractals** também gera alguns fractais por autossimilaridade como exemplificaremos nesta seção.

Programando fractais

Podemos usar os recursos de programação do aplicativo `Maxima` e implementar fractais, sem necessidade de usarmos pacotes. Apresentaremos alguns exemplos na Seção 3.4.

3.3.1 O conjunto de Cantor ternário

Construção por remoção

O conjunto triádico de Cantor é o conjunto formado pela remoção de uma seqüência de intervalos abertos do intervalo [0,1]. Primeiro, dividimos intervalo [0,1] em três partes iguais e removemos a parte do meio, isto é, o intervalo $(1/3, 2/3)$ para obter o seguinte conjunto:

$$\mathcal{C}_1 = \left[0, \frac{1}{3}\right] \cup \left[\frac{2}{3}, 1\right].$$

Novamente removemos o terço médio de cada intervalo fechado de $\mathcal{C}_1$ para obter o seguinte conjunto:

$$\mathcal{C}_2 = \left[0, \frac{1}{9}\right] \cup \left[\frac{2}{9}, \frac{1}{3}\right] \cup \left[\frac{2}{3}, \frac{7}{9}\right] \cup \left[\frac{8}{9}, 1\right].$$

Note que $\mathcal{C}_2$ é a união de $2^2 = 4$ intervalos fechados, cada um do tipo $[j/3^2, (j+1)/3^2]$ e que $\mathcal{C}_1 \supset \mathcal{C}_2$.

Novamente removemos o terço médio de cada um destes intervalos para obter o conjunto $\mathcal{C}_3$, que é a união de $2^3 = 8$ intervalos fechados tal que $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \mathcal{C}_3$. Continuando este processo n-estágios obteremos o conjunto $\mathcal{C}_n$, que é a união de 2^k intervalos do tipo $[j/3^n, (j+1)/3^n]$ tal que $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \ldots \supset \mathcal{C}_n \supset \ldots$

O conjunto triádico de Cantor é a interseção dos conjuntos $\mathcal{C}_n$, $n \in \mathbb{N}$, ou seja,

$$\mathcal{C} = \bigcap_{n\in\mathbb{N}} \mathcal{C}_n.$$

A figura abaixo ilustra o processo de construção do conjunto de Cantor.

Construção por autossimilaridade

Vamos aplicar o Teorema 3.1. Para isto, considere $X = \mathbb{R}$ com a métrica usual e $F : \mathbb{K}(\mathbb{M}) \to \mathbb{K}(\mathbb{M})$ dada por $F(K) = f_1(K) \cup f_2(K)$ com $f_1(x) = \dfrac{x}{3}$ e $f_2(x) = \dfrac{x}{3} + \dfrac{2}{3}$ e $K_0 = [0,1]$.

Observe que, $f_1(x) = \dfrac{x}{3}$ efetua uma redução de $[0,1]$ com fator $\dfrac{1}{3}$ e $f_2(x) = \dfrac{x}{3} + \dfrac{2}{3}$ efetua uma redução de $[0,1]$ com fator $\dfrac{1}{3}$ e depois uma translaçao de $\dfrac{2}{3}$ para a direita. Fazendo $\mathcal{C}_{n+1} = f_1(\mathcal{C}_n) \cup f_2(\mathcal{C}_n)$ temos que

$$\mathcal{C}_1 = \left[0, \frac{1}{3}\right] \cup \left[\frac{2}{3}, 1\right],\ \mathcal{C}_2 = \left[0, \frac{1}{9}\right] \cup \left[\frac{2}{9}, \frac{1}{3}\right] \cup \left[\frac{2}{3}, \frac{7}{9}\right] \cup \left[\frac{8}{9}, 1\right], \ldots$$

Observe que f_1 e f_2 são transformações similaridades e pelo Teorema 3.1, F tem um único ponto fixo. Mostraremos que este ponto fixo é o conjunto ternário de Cantor.

Para isto, observe que, por indução, obtemos $\mathcal{C}_{n+1} = f_1(\mathcal{C}_n) \cup f_2(\mathcal{C}_k)$ para $n \geq 0$. Primeiro, mostraremos que $\mathcal{C} \subseteq f_1(\mathcal{C}) \cup f_2(\mathcal{C})$.

Seja $x \in \mathcal{C}$. Então, $x \in \bigcap_{n\in\mathbb{N}} \mathcal{C}_n$, e logo, $x \in \mathcal{C}_1$, ou seja, $x \in \left[0, \dfrac{1}{3}\right]$ ou $x \in \left[\dfrac{2}{3}, 1\right]$. Vamos considerar $x \in \left[\dfrac{2}{3}, 1\right]$ (o outro caso é análogo). Assim, para todo n sabemos que

$x \in \mathcal{C}_{n+1} = f_1(\mathcal{C}_n) \cup f_2(\mathcal{C}_n)$. Mas

$$f_1(\mathcal{C}_n) \subseteq f_1([0,1]) = \left[0, \frac{1}{3}\right].$$

Logo, $x \in f_2(\mathcal{C}_n)$, ou seja, $3x-2 \in \mathcal{C}_n$ para todo n, então $3x-2 \in \bigcap_{n\in\mathbb{N}} \mathcal{C}_n = \mathcal{C}$, ou seja, $x \in f_1(\mathcal{C})$. No outro caso, temos que $x \in f_2(\mathcal{C})$. Portanto, $x \in f_1(\mathcal{C}) \cup f_2(\mathcal{C})$.

Agora, mostraremos que $\mathcal{C} \supseteq f_1(\mathcal{C}) \cup f_2(\mathcal{C})$. Seja $x \in f_1(\mathcal{C}) \cup f_2(\mathcal{C})$. Então, $x \in f_1(\mathcal{C})$ ou $x \in f_2(\mathcal{C})$. Considere $x \in f_2(\mathcal{C})$ (o outro caso é análogo) então $3x-2 \in \mathcal{C}$. Assim, para qualquer n tem-se $3x-2 \in \mathcal{C}_n$, ou seja, $x \in f_2(\mathcal{C}_n) \subset \mathcal{C}_n$. Logo, $x \in \bigcap_{n\in\mathbb{N}} \mathcal{C}_n = \mathcal{C}$. ■

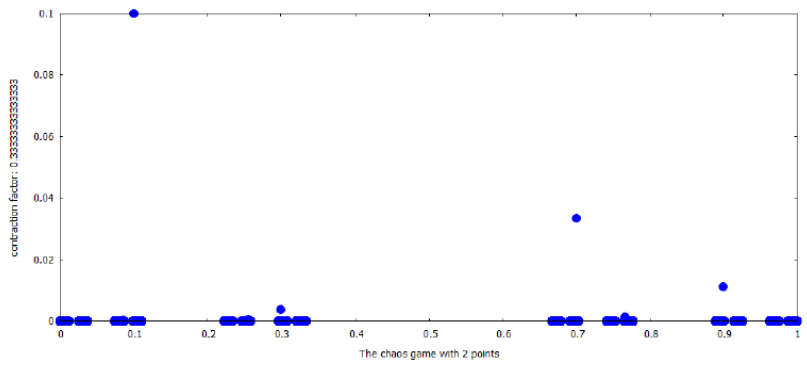
Exemplo 3.2 Desenhe o Conjunto de Cantor usando o jogo do caos

Solução:

Maxima

```
(%i496)load(dynamics)$
(%i497)chaosgame([[0, 0], [1, 0]],
       [0.1, 0.1], 1/3, 20000, [gnuplot_preamble,
       "pointsize=0.5"],[gnuplot_out_file,
       "Cantor.eps"]);
(%o413)   Cantor.eps
```

O aplicativo salvou um arquivo com o nome "Cantor.eps", no formato EPS, na pasta de trabalho.



Observação 3.1 *Note que existem pontos fora do conjunto. Isto acontece por que o jogo do caos é probalilistíco.*

3.3.2 O Triângulo de Sierpiński

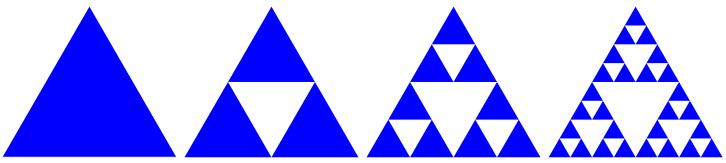
Construção por remoção

O processo de construção do Triângulo de Sierpiński é similar ao da construção do conjunto triádico de Cantor: a partir de um objeto inicial, retiramos *uma parte central* sua, e ainda obtendo cópias similares ao objeto inicial. Procedemos então de forma similar com cada cópia e assim indefinidamente. Mais precisamente, inicia-se com uma região triangular equilátera $\mathcal{S}_0$, de lado 1 e vértices $v_1 = (0,0)$, $v_2 = (1,0)$ e $v_3 = (1/2, \sqrt{3}/2)$, subdividindo-a em quatro regiões triangulares menores, usando para isto os segmentos de reta que unem os pontos médios de cada lado. Remove-se então o interior da região triangular central (a região central, sem sua fronteira). Note que assim é gerado um conjunto $\mathcal{S}_1$ formado por três regiões triangulares menores e congruentes, cujos lados medem a metade da medida dos lados do triângulo original. Repete-se então esse mesmo procedimento em cada uma das três regiões triangulares e assim sucessivamente, gerando-se uma seqüência de conjuntos $\mathcal{S}_0 \supset \mathcal{S}_1 \supset \mathcal{S}_2 \supset \cdots$ com cada $\mathcal{S}_n$ formado por 3^n regiões triangulares, similares à região inicial, cujos lados medem 2^{-n} da medida do lado do triângulo inicial.

O Triângulo de Sierpiński $\mathcal{S}$ é o conjunto limite desta seqüência de pontos, isto é,

$$\mathcal{S} = \bigcap_{n \in \mathbb{N}} \mathcal{S}_n.$$

A figura abaixo ilustra o processo de construção do Triângulo de Sierpiński:



Construção por autossimilaridade

Para construírmos o triângulo de Sierpiński por similaridade, considere $X = \mathbb{R}^2$ com a métrica usual e $F : \mathbb{K}(X) \to \mathbb{K}(X)$ dada por $F(K) = f_1(K) \cup f_2(K) \cup f_3(K)$ com

$$\begin{cases} f_1(x,y) &= \left(\dfrac{x}{2}, \dfrac{y}{2}\right); \\ f_2(x,y) &= \left(\dfrac{x}{2} + \dfrac{1}{2}, \dfrac{y}{2}\right); \\ f_3(x,y) &= \left(\dfrac{x}{2}, \dfrac{y}{2} + \dfrac{1}{2}\right). \end{cases}$$

e K_0 uma região triangular equilátera de lado 1 e vértices $v_1 = (0,0)$, $v_2 = (1,0)$ e $v_3 = (1/2, \sqrt{3}/2)$.

Observe que, $f_1(x,y)$ reduza K_0 a metade, $f_2(x,y)$ reduza K_0 a metade e transla $\frac{1}{2}$ para a direita e $f_3(x,y)$ reduza K_0 a metade, transla $\frac{1}{4}$ para a direita e $\frac{1}{2}$ para cima.

Fazendo $\mathcal{S}_{n+1} = f_i(\mathcal{S}_n) \cup f_2(\mathcal{S}_n) \cup f_3(\mathcal{S}_n)$ temos que o conjunto $\mathcal{S}_1$ é formado por três regiões triangulares menores e congruentes, cujos lados medem a metade da medida dos lados do triângulo inicial. Deste modo, gera-se a seqüência de conjuntos $\mathcal{S}_0 \supset \mathcal{S}_1 \supset \mathcal{S}_2 \supset \cdots$ Do mesmo modo, como procedemos para o Conjunto de Cantor ternário, podemos provar que o ponto fixo de F é o triângulo de Sierpiński.

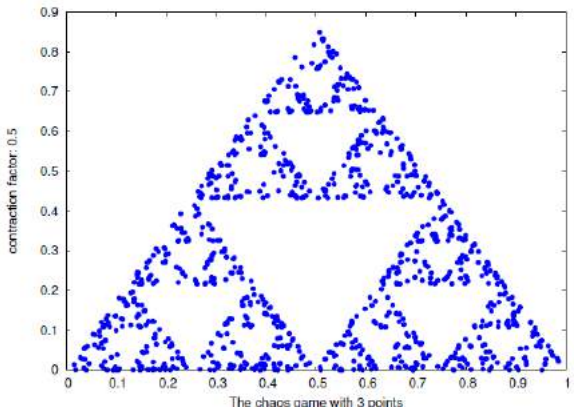
Exemplo 3.3 Desenhe o Triângulo de Sierpisńki usando o jogo do caos.

Solução:

Maxima

```
(%i498)load(dynamics)$
(%i499)chaosgame([[0, 0], [1, 0], [0.5, sqrt(3)/2]],
       [0.1, 0.1], 1/2, 2000, [gnuplot_preamble,
       "pointsize=0.5"],[gnuplot_out_file,
       "triangSierpinski.eps"]);
(%o414)   triangSierpinski.eps
```

O aplicativo salvou um arquivo com o nome “triangSierpinski” , no formato EPS, na pasta de trabalho.



Exemplo 3.4 Desenhe o triângulo de Sienpiński pelo método de IFS, sabendo que as IFS deste fractal são dadas por: $m = m_1 = m_2 = m_3$;

$$m = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}; \quad p_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}; \quad p_2 = \begin{bmatrix} 0.5 \\ 0.0 \end{bmatrix}; \quad p_3 = \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix}.$$

Solução:

Maxima

```
(%i500)load(dynamics)$
(%i501)m: matrix([0.5, 0.0], [0.0, 0.5]);
       p1: matrix([0.0], [0.0]);
```

$$(\%o415)\quad \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}$$

$$(\%o416)\quad \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

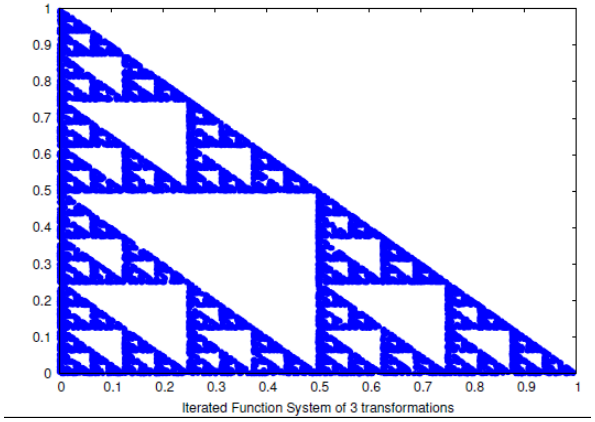
```
(%i502)p2: matrix([0.5], [0.0]);
```

$$(\%o417)\quad \begin{bmatrix} 0.5 \\ 0.0 \end{bmatrix}$$

```
(%i503)p3: matrix([0.0], [0.5]);
```

$$(\%o418)\quad \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix}$$

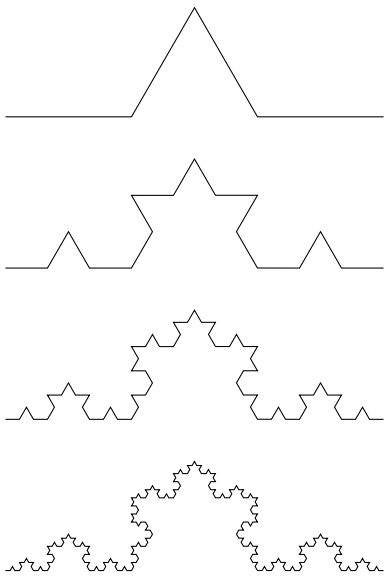
```
(%i504)ifs([1,2,3],[m,m,m],[p1,p2,p3],[0.5,0.5],1000,
       [gnuplot_preamble,"pointsize=0.5"],[gnuplot_term,ps],
       [gnuplot_out_file,"SierpinskiMaxima.eps"]);
(%o419)  SierpinskiMaxima.eps
```



3.3.3 A curva de Koch

Construção por remoção

A construção da curva de Koch inicia-se com um segmento de reta $\mathcal{K}_0$, tomemos por exemplo o segmento $[0,1]$, dividindo-o em três partes iguais e substituindo-se o o terço médio por dois segmentos adjacentes (tais que estes formem um "triângulo" equilátero sem a base), formando então uma linha poligonal $\mathcal{K}_1$ de quatro segmentos consecutivos, em uma espécie de "ponta de estrela". Repete-se então o processo em cada um dos segmentos de $\mathcal{K}_1$, obtendo-se assim $\mathcal{K}_2$ e continua-se indefinidamente gerando uma seqüência de conjuntos $\{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \ldots\}$, cujo limite é a *curva de Koch*.



Construção por autossimilaridade

A curva de Koch é o atrator do sistema de funções iteradas dado pelas similaridades

$$\begin{cases} f_1(x,y) &= \left(\frac{x}{3}, \frac{y}{3}\right) \\ f_2(x,y) &= \left(\frac{\sqrt{2}(x-y)}{6} + \frac{1}{3}, \frac{\sqrt{2}(x+y)}{6}\right) \\ f_3(x,y) &= \left(-\frac{\sqrt{2}(x+y)}{6} + \frac{2}{3}, \frac{\sqrt{2}(-x+y)}{6}\right) \\ f_4(x,y) &= \left(\frac{x}{3} + \frac{2}{3}, \frac{y}{3}\right). \end{cases}$$

Exemplo 3.5 Desenhe a curva de Kock pelo método de IFS, sabendo que as IFS deste fractal são dadas por:

$$m_1 = \begin{bmatrix} 0.3333 & 0 \\ 0 & 0.333 \end{bmatrix}; \quad p_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix};$$

$$m_2 = \begin{bmatrix} 0.2357 & -0.2357 \\ 0.2357 & 0.2357 \end{bmatrix}; \quad p_2 = \begin{bmatrix} 0.3333 \\ 0.0 \end{bmatrix};$$

$$m_3 = \begin{bmatrix} -0.2357 & -0.2357 \\ -0.2357 & 0.2357 \end{bmatrix}; \quad p_3 = \begin{bmatrix} 0.6666 \\ 0.0 \end{bmatrix};$$

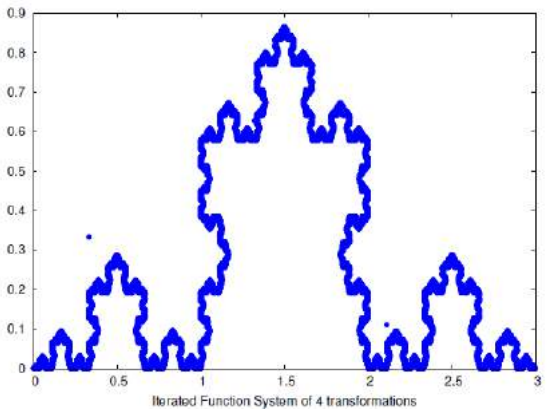
$$m_4 = \begin{bmatrix} 0.3333 & 0 \\ 0 & 0.333 \end{bmatrix}; \quad p_4 = \begin{bmatrix} 0.6666 \\ 0.0 \end{bmatrix}.$$

Solução:

Maxima

```
(%i505)load(dynamics)$
(%i506)m1: matrix([0.3333, 0.0], [0.3333, 0.0]);
       p1: matrix([0.0], [0.0])$
(%i507)m2: matrix([0.2357, -0.2357], [0.2357, 0.2357]);
       p2: matrix([0.3333], [0.0])$
(%i508)m3: matrix([-0.2357, -0.2357], [-0.2357, 0.2357]);
       p3: matrix([0.6666],[0.0])$
```

```
(%i509)m4: matrix([0.3333, 0.0], [0.3333, 0.0]);
       p4: matrix([0.6666], [0.0])$
(%i510)ifs([1,2,3,4],[m1,m2,m3.m4],[p1,p2,p3,p4],[0.3333,0.3333],30000,
       [gnuplot_preamble,"pointsize=0.5"],[gnuplot_term,ps],
       [gnuplot_out_file,"KockMaxima.eps"]);
(%o420)   KockMaxima.eps
```



3.3.4 Curvas de Peano

Em 1878, George Cantor mostrou a existência de uma função bijetora definida no intervalo $[0,1]$ e assumindo valores no quadrado $[0,1]\times[0,1]$ e a Matemática nunca mais foi a mesma. Em outras palavras, Cantor mostrou que existem tantos pontos num quadradro quanto num intervalo. Pela primeira vez postulou-se o problema da "invariância da dimensão". Imediatamente os matemáticos da época se perguntaram se tal função poderia ser contínua. A pergunta é relevante, pois se existisse tal função contínua teríamos que o intervalo $[0,1]$ e o quadrado $[0,1]\times[0,1]$ seriam similares, o que foge totalmente da nossa intuição!

Em 1879, E. Netto mostrou que tal função é necessariamente descontínua e os matemáticos passaram a buscar uma função contínuas e sobrejetora. Como uma função contínua definida em qualquer intervalo assumindo valores no plano é chamada de "curva", a pergunta na mente dos matemáticos da época era:

> *Existe uma curva que "passa" por todos os pontos de um quadrado (e o seu interior) com área positiva?*

Em 1890, G. Peano respondeu esta pergunta construindo a primeira curva que passa por todos os pontos de uma região quadrada. Tais curvas são chamadas *curvas que preenchem o espaço* ou *curvas de Peano*.

Apesar de Peano ter apresentado a primeira curva que preenche o espaço, foi o matemático francês D. Hilber quem, em 1891, construiu, pela primeira vez, tais curvas por um processo geométrico e recursivo.

Nesta seção construíremos algums curvas que preenchem o espaço, e apesar de tais curvas não serem fractais (o quadrado não é um fractal!) o seu processo de construção usa autossimiliaridade.

Curva de Peano Vamos construir a curva de Peano usando transformações similares. Considere um quadrado unitário (e o seu interior) e a sua diagonal, a qual chamaremos de gerador. A curva de Peano é construída fazendo-se nove cópias do gerador com as seguintes transformações similaridade:

$$\begin{cases} f_1(x,y) &= \left(\frac{x}{3},\frac{y}{3}\right) \\ f_2(x,y) &= \left(-\frac{x}{3},\frac{y}{3}\right)+\left(\frac{1}{3},\frac{1}{3}\right)=\left(-\frac{x}{3}+\frac{1}{3},\frac{y}{3}+\frac{1}{3}\right) \\ f_3(x,y) &= \left(\frac{x}{3},\frac{y}{3}\right)+\left(0,\frac{2}{3}\right)=\left(\frac{x}{3},\frac{y}{3}+\frac{2}{3}\right) \\ f_4(x,y) &= \left(\frac{x}{3},-\frac{y}{3}\right)+\left(\frac{1}{3},1\right)=\left(\frac{x}{3}+\frac{1}{3},-\frac{y}{3}+1\right) \\ f_5(x,y) &= \left(-\frac{x}{3},-\frac{y}{3}\right)+\left(\frac{2}{3},\frac{2}{3}\right)=\left(-\frac{x}{3}+\frac{2}{3},-\frac{y}{3}+\frac{2}{3}\right) \\ f_6(x,y) &= \left(\frac{x}{3},-\frac{y}{3}\right)+\left(\frac{1}{3},\frac{1}{3}\right)=\left(\frac{x}{3}+\frac{1}{3},-\frac{y}{3}+\frac{1}{3}\right) \\ f_7(x,y) &= \left(\frac{x}{3},\frac{y}{3}\right)+\left(\frac{2}{3},0\right)=\left(\frac{x}{3}+\frac{2}{3},\frac{y}{3}\right) \\ f_8(x,y) &= \left(-\frac{x}{3},\frac{y}{3}\right)+\left(1,\frac{1}{3}\right)=\left(-\frac{x}{3}+1,\frac{y}{3}+\frac{1}{3}\right) \\ f_9(x,y) &= \left(\frac{x}{3},\frac{y}{3}\right)+\left(\frac{2}{3},\frac{2}{3}\right)=\left(\frac{x}{3}+\frac{2}{3},\frac{y}{3}+\frac{2}{3}\right) \end{cases}$$

Curva de Hilbert

Agora vamos construir a curva de Hilbert. Para isto, vamos tomar como gerador um segmento de reta que une os pontos $(0,0)$ e $(1,0)$. A curva de Hilbert é construída fazendo-se quatro cópias do gerador com as seguintes transformações similares:

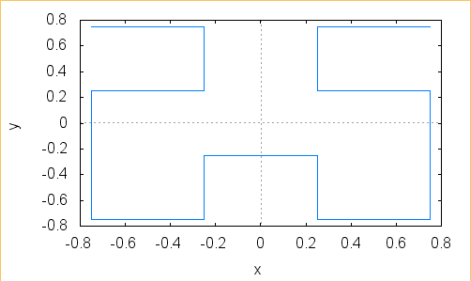
$$\begin{cases} f_1(x,y) &= \left(\frac{y}{2},\frac{x}{2}\right) \\ f_2(x,y) &= \left(-\frac{x}{2},\frac{y}{2}\right)+\left(0,\frac{1}{2}\right)=\left(\frac{x}{2},\frac{y}{2}+\frac{1}{2}\right) \\ f_3(x,y) &= \left(\frac{x}{2},\frac{y}{2}\right)+\left(\frac{1}{2},\frac{1}{2}\right)=\left(\frac{x}{2}+\frac{1}{2},\frac{y}{2}+\frac{1}{2}\right) \\ f_4(x,y) &= \left(-\frac{y}{2},-\frac{x}{2}\right)+\left(1,\frac{1}{2}\right)=\left(-\frac{y}{2}+1,-\frac{x}{2}+\frac{1}{2}\right) \end{cases}$$

Exemplo 3.6 Desenhe cinco estágios da curva de Hilbert com o pacote **fractals**

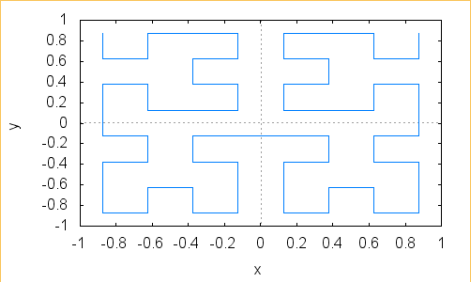
Solução:

Maxima

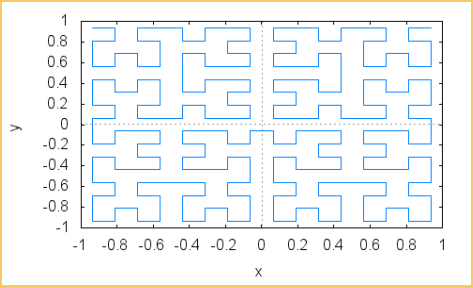
```
(%i511)load(fractals)$
(%i512)n:10000$
(%i513)wxplot2d([discrete,hilbertmap(1)])
```




```
(%o421)
(%i514)wxplot2d([discrete,hilbertmap(2)])
```

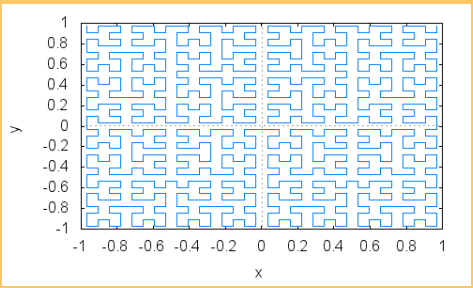



```
(%o422)
(%i515)wxplot2d([discrete,hilbertmap(3)])
```



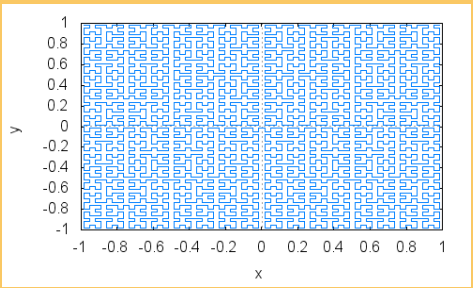
(%o423)

```
(%i516)wxplot2d([discrete,hilbertmap(4)])
```



(%o424)

```
(%i517)wxplot2d([discrete,hilbertmap(5)])
```



(%o425)

3.3.5 O conjunto de Julia e Mandelbrot

O conjunto de Julia é a ilustração muito interessante de como processos aparentemente simples geram conjunto altamente complexos. Funções definidas no plano complexo $\mathbb{C}$ tal como $f(z) = x^2 + c$, com c uma constante, produzem fractais de aparência bastante "éxotica".

Por simplicidade, vamos considerar $f : \mathbb{C} \to \mathbb{C}$ um polinômio de grau $n \geq 2$ com coeficientes complexos, $f(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0$.

Seja f^k a composta de f com ela mesma k vezes, ou seja, $f \circ \ldots \circ f$, e $f^k(z)$ é a k-ésima iterada $f(f(\ldots(f(z)\ldots))$ de z. Os conjuntos de Julia são definidos em termos do comportamento das iteradas $f^k(z)$ para k grande.

Definição 3.3 (Conjunto de Julia) *Definimos o* **conjunto de Julia** *de f como a fronteira do conjunto $K(f) = \{z \in \mathbb{C}; f^k(z) \nrightarrow \infty\}$. Portanto, $J(f) = \partial K(f)$.*

O complementar do conjunto de Julia é chamado *conjunto de Fatou*.

Observação 3.2 *O símbolo $\nrightarrow$ significa que a sequência $f^k(z)$ não tende para ∞ quando $k \to \infty$.*

Aqui estamos interessados nos conjuntos de Julia de polinômios que são fractais. O estudo da seqüência $f^k(z)$ para vários valores iniciais z é conhecido como *dinâmica complexa*. Para mais detalhes consulte [5].

Exemplo 3.7 Seja $f(z) = z^2$ então $f^k(z) = z^{2^k}$. Não é difícil verificar que

$$\lim_{k\to\infty} f^k(z) = 0 \quad \text{para} \quad |z| < 1;$$
$$\lim_{k\to\infty} f^k(z) = \infty \quad \text{para} \quad |z| > 1.$$

Além disso, para $|z| = 1$ temos que $f^k(z)$ pertence ao círculo $|z| = 1$, para todo k.

Portanto, $K(f)$ é o disco unitário $|z| \leq 1$ e o conjunto de Julia é a sua fronteita, ou seja, o círculo unitário $|z| = 1$. Naturalmente, neste caso especial, o conjunto de Julia não é um fractal. ■

Vamos considerar conjuntos de Julia de polinômios do tipo $f_c(z) = z^2 + c$, com c uma constante complexa.

Para definirmos o conjunto de Mandelbrot precisamos da noção de *conjunto conexo*. Dizemos que um conjunto $A \subset \mathbb{R}^n$ é conexo se não existem conjuntos abertos tais que $U \cup V$ contém A e $A \cap U$ e $A \cap V$ são não vazios e disjuntos. Intuitivamente, ser conexo significa ser uma única "peça".

Definição 3.4 (Conjunto de Mandelbrot) *Definimos o conjunto de Mandelbrot* $\mathbb{M}$ *como o conjunto dos parâmetros c para os quais o conjunto de Julia de* f_c *é conexo. Então,*

$$\mathbb{M} = \{c \in \mathbb{C}; J(f_c) \text{ é conexo.}\}.$$

O conjunto de Mandelbrot contém muita informação sobre a estrutura dos conjuntos de Julia. Por exemplo, podemos mostrar que $c \in \mathbb{M}$ se, e somente se, $f_c^k(0) \nrightarrow \infty$ (para mais detalhes consulte [5, p. 223]).

O conjunto de Julia com o `Maxima`

O seguinte comando do `Maxima` gera a imagem de um conjunto de Julia para $c = x + iy$ com x e y reais:

julia(x,y,opções)

Ao final da tarefa, o aplicativo exibe a mensagem *File julia.xpm was created* e cria um arquivo gráfico no diretório de trabalho, no formato gráfico XPM. Esse arquivo gráfico pode ser aberto com qualquer programa gráfico.

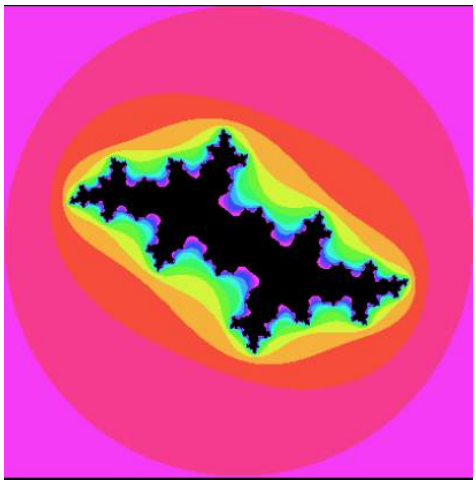
O seguinte exemplo ilustra o uso do comando **julia**:

Exemplo 3.8 Vamos gerar o conjunto de Julia de $f_c(z) = z^2 + c$ com $c = -0.55 + 0.6i$.

Solução:

Maxima

```
(%i518)load(dynamics)$
(%i519)julia(-0.55,0.6);
  File julia.xpm was created
(%o426)   false
```



Os pontos pretos na figura acima são o conjunto de Julia do Exemplo 3.8. A origem encontra-se no centro do quadrado. A região apresentada corresponde a valores reais e imaginários menores que 1,3 em valor absoluto. Os pontos que não pertencem ao conjunto de Julia foram representados com uma cor, que corresponde ao número de iterações antes de a seqüência se afastar da origem mais do que duas unidades (se depois de 40 iterações isso não tivesse acontecido, o ponto continua pintado de negro).

Para aumentarmos o número de iterações, usa-se a opção **levels** no comando **julia**. Por exemplo,

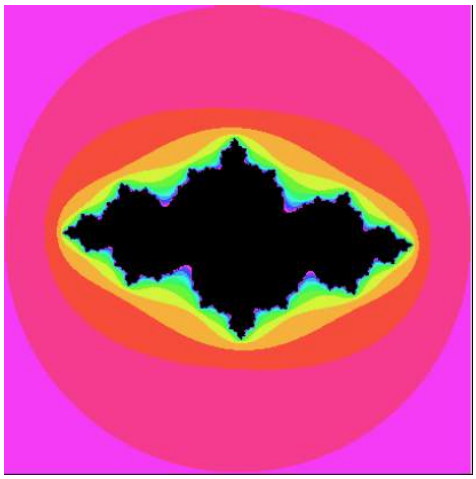
Exemplo 3.9 Vamos gerar o conjunto de Julia de $f_c(z) = z^2 + c$ com $c = -0.75 + 0.1i$ com 36 iterações.

Maxima

```
(%i520)julia(-0.75,0.1, [levels, 36]);
```

File julia.xpm was created

```
(%o427)  false
```



Podemos usar a opcção **radius** para reduzir o tamanho da região mostrada no gráfico:

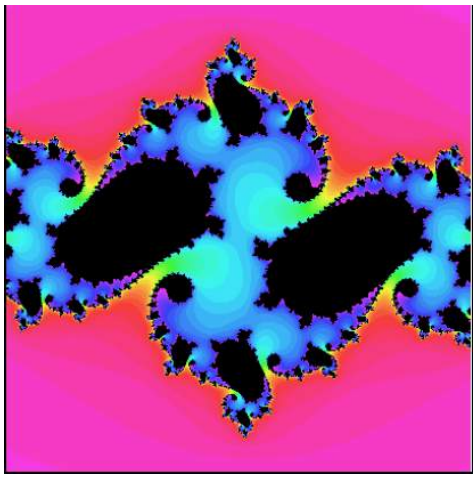
Exemplo 3.10 Vamos gerar o conjunto de Julia de $f_c(z) = z^2 + c$ com $c = -0.75 + 0.1i$ com 48 iterações e raio 1.

Maxima

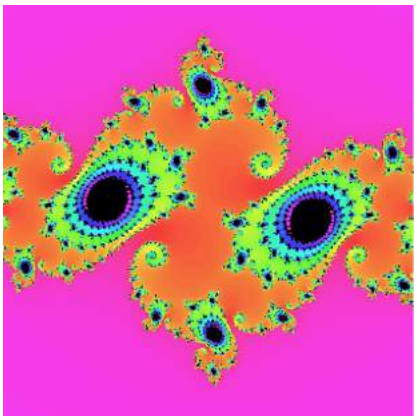
```
(%i521)julia(-0.75,0.1, [levels, 48], [radius, 1]);
```

File julia.xpm was created

```
(%o428)  false
```



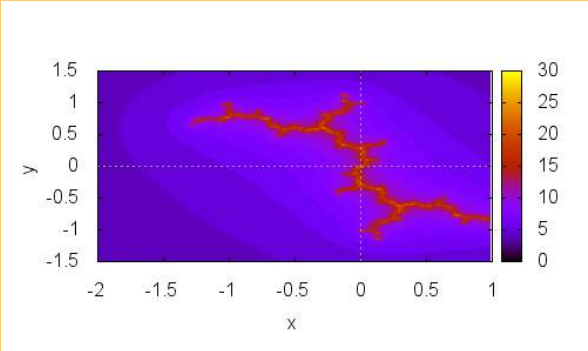
Como 160 iterações (veja figura abaixo), eliminamos ainda mais pontos que não pertencem ao conjunto de Julia:



Podemos gerar conjuntos de Julia com o pacote **fractals** do seguinte modo:

Maxima

```
(%i522)load(fractals)$
(%i523)n:10000$
(%i524)plot3d (julia_set, [x, -2, 1], [y, -1.5, 1.5],
       [gnuplot_preamble, "set view map"],
       [gnuplot_pm3d, true],[grid, 150, 150])
```



(%o429)

O conjunto de Mandelbrot com o Maxima

O seguinte comando do aplicativo Maxima gera a imagem de um conjunto de Mandelbrot:

mandelbrot(opções)

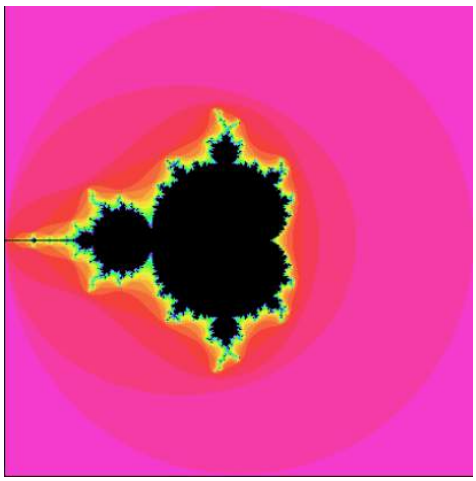
Ao final da tarefa, o aplicativo exibe a mensagem *File mandelbrot.xpm was created* e cria um arquivo gráfico no diretório de trabalho, no formato gráfico XPM. Esse arquivo gráfico pode ser aberto com qualquer programa gráfico.

O seguinte exemplo ilustra o uso do comando **mandelbrot**:

Exemplo 3.11 Vamos gerar um conjunto de Mandelbrot com 30 iterações.

Maxima

```
(%i525)load(dynamics)$
(%i526)mandelbrot([levels, 30]);
  File mandelbrot.xpm was created
(%o430)  false
```

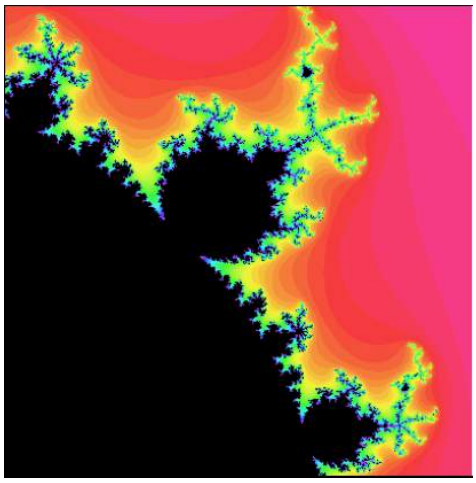


O conjunto de Mandelbrot reune num único gráfico todas as formas encontradas nos conjuntos de Julia. Por exemplo,

Exemplo 3.12 Vamos gerar um conjunto de Mandelbrot com a ampliação de uma pequena região do conjunto, centrada no ponto $0.3 + 0.5i$.

Maxima

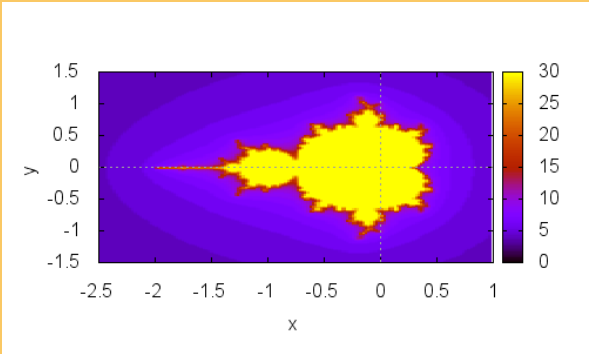
```
(%i527)mandelbrot([levels, 30]);
  File mandelbrot.xpm was created
(%o431)  false
```



Podemos gerar conjuntos de Mandelbrot com o pacote **fractals** do seguinte modo:

Maxima

```
(%i528)load(fractals)$
(%i529)n:10000$
(%i530)plot3d (mandelbrot_set, [x, -2.5, 1], [y, -1.5, 1.5],
       [gnuplot_preamble, "set view map"],
       [gnuplot_pm3d, true],[grid, 150, 150]);
```



(%o432)

3.3.6 Mais exemplos

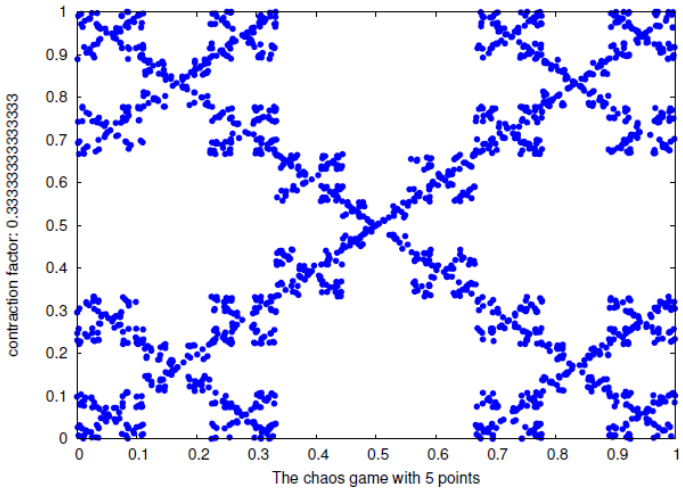
Nesta seção, usaremos o pacote **fractals** para gerar outros fractais.

Exemplo 3.13 Desenhe o fractal gerado por cinco pontos, nos vértices e no centro de um quadrado, com factor de redução 1/3.

Solução:

Maxima

```
(%i531)load(dynamics)$
(%i532)chaosgame([[0,0],[0,1],[1,0],[1,1],[0.5,0.5]],
       [0.5,0.5],1/3, 2000,
       [gnuplot_preamble, "pointsize=0.5"],
       [gnuplot_term,ps],
       [gnuplot_out_file,"fractal1.eps"]);
(%o433)  fractal1.eps
```



Exemplo 3.14 Desenhe o fractal "árvore" dado pelas IFS:

$$m_1 = \begin{bmatrix} -0.550 & -0.179 \\ -0.179 & 0.550 \end{bmatrix}; \quad p_1 = \begin{bmatrix} -0.438 \\ 0.382 \end{bmatrix};$$

$$m_2 = \begin{bmatrix} -0.246 & 0.193 \\ 0.275 & 0.365 \end{bmatrix}; \quad p_2 = \begin{bmatrix} -0.379 \\ 0.538 \end{bmatrix};$$

$$m_3 = \begin{bmatrix} 0.006 & -0.014 \\ -0.147 & -0.459 \end{bmatrix}; \quad p_3 = \begin{bmatrix} -0.283 \\ 0.490 \end{bmatrix}.$$

Solução:

Maxima

```
(%i533)load(dynamics)$
(%i534)m1: matrix([-0.550, -0.179], [-0.179, 0.550]);
       p1: matrix([-0.438], [0.382]);
```

$$(\%o434) \quad \begin{bmatrix} -0.550 & -0.179 \\ -0.179 & 0.550 \end{bmatrix}$$

$$(\%o435) \quad \begin{bmatrix} -0.438 \\ 0.382 \end{bmatrix}$$

```
(%i535)m2: matrix([-0.246, 0.193], [0.275, 0.365]);
       p2: matrix([-0.379], [0.538]);
```

$$(\%o436) \quad \begin{bmatrix} -0.246 & 0.193 \\ 0.275 & 0.365 \end{bmatrix}$$

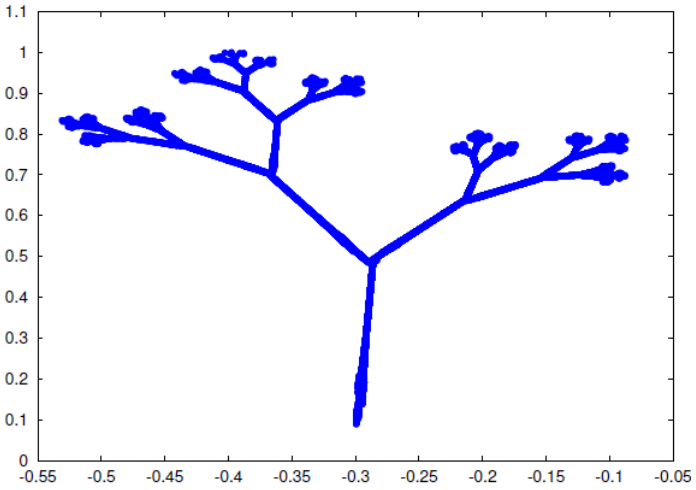
$$(\%o437) \quad \begin{bmatrix} -0.379 \\ 0.538 \end{bmatrix}$$

```
(%i536)m3: matrix([0.006, -0.014], [-0.147, -0.459]);
       p3: matrix([-0.283], [0.490]);
```

$$(\%o438) \quad \begin{bmatrix} 0.006 & -0.014 \\ -0.147 & -0.459 \end{bmatrix}$$

$$(\%o439) \quad \begin{bmatrix} -0.283 \\ 0.490 \end{bmatrix}$$

```
(%i537)ifs([1,2,3],[m1,m2,m3],[p1,p2,p3],[-0.283,0.490],30000,
       [gnuplot_preamble,"pointsize=0.5"],[gnuplot_term,ps],
       [gnuplot_out_file,"arvore.eps"]);
(%o440)   arvore.eps
```



Exemplo 3.15 Desenhe o fractal dado pelas IFS:

$$m_1 = \begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}; \quad p_1 = \begin{bmatrix} 0.0951 \\ 0.5893 \end{bmatrix};$$

$$m_2 = \begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}; \quad p_2 = \begin{bmatrix} 0.4413 \\ 0.7893 \end{bmatrix};$$

$$m_3 = \begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}; \quad p_3 = \begin{bmatrix} 0.0952 \\ 0.9893 \end{bmatrix}.$$

Solução:

Maxima

```
(%i538)load(dynamics)$
(%i539)m1: matrix([0.0, 0.577], [-0.577, 0.0]);
       p1: matrix([0.0951], [0.5893]);
```

(%o441)
$$\begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}$$

(%o442)
$$\begin{bmatrix} 0.0951 \\ 0.5893 \end{bmatrix}$$

```
(%i540)m2:  matrix([0.0, 0.577], [-0.577, 0.0]);
       p2: matrix([0.4413], [0.7893];
```

(%o443)
$$\begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}$$

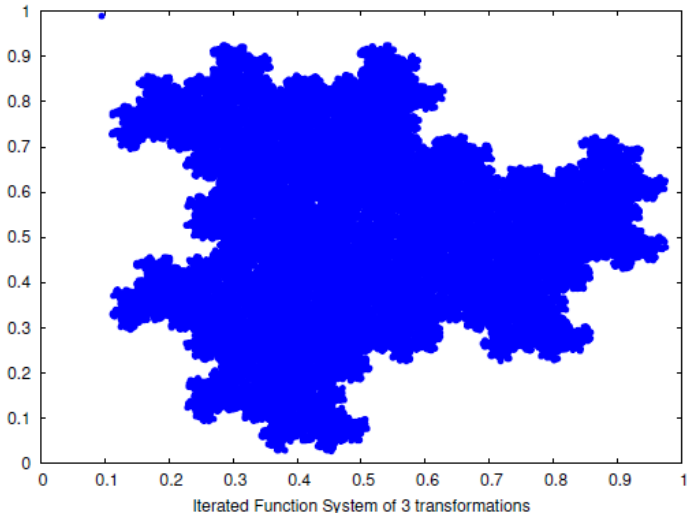
(%o444)
$$\begin{bmatrix} 0.4413 \\ 0.7893 \end{bmatrix}$$

```
(%i541)m3: matrix([0.0, 0.577], [-0.577, 0.0]);
       p3: matrix([0.0952], [0.9893]);
```

(%o445)
$$\begin{bmatrix} 0.0 & 0.577 \\ -0.577 & 0.0 \end{bmatrix}$$

(%o446)
$$\begin{bmatrix} 0.0952 \\ 0.9893 \end{bmatrix}$$

```
(%i542)ifs([1,2,3],[m1,m2,m3],[p1,p2,p3],[0.0952,0.9893],30000,
       [gnuplot_preamble,"pointsize=0.5"],[gnuplot_term,ps],
       [gnuplot_out_file,"dragao.eps"]);
(%o447)  dragao.eps
```



3.4 Programando fractais com o Maxima

Também podemos gerar fractais programando como o aplicativo Maxima. Nesta seção apresentaremos um algoritmo para gerar fractais pelo processo de autossimilaridade. O algoritmo é o seguinte:

1. Inserir as funções iteradas;
2. Definir uma lista de pontos iniciais;
3. Inserir o número de etapas (ou iterações) que o Maxima deverá executar;
4. Gerar a lista de pontos que compõe uma determinada etapa do fractal;
5. Ligar os pontos por segmentos de reta.

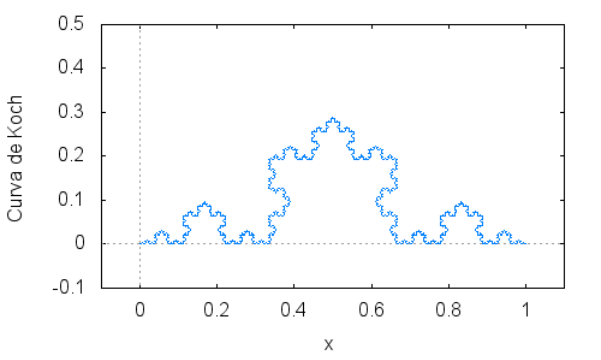
Exemplo 3.16 Faça um programa para gerar a Curva de Koch.

Solução: No apêndice A.1 apresentaremos a programação passo a passo.

Maxima

```
(%i543)f[1](X):=1/3*X$
(%i544)f[2](X):=1/3*[X[1]*0.5-float(sqrt(3))*X[2]*0.5,
       float(sqrt(3))*0.5*X[1]+0.5*X[2]]+[1/3,0]$
(%i545)f[3](X):=1/3*[X[1]*0.5+float(sqrt(3))*X[2]*0.5,
       -float(sqrt(3))*0.5*X[1]+0.5*X[2]]
        +[1/2,float(sqrt(3))/6]$
```

```
(%i546)f[4](X):=1/3*[X[1],X[2]]+[2/3,0]$
(%i547)Koch(n):=(k:n,L[0]:[[1,0]],
       (for m:1 thru k do
           L[m]:create_list(f[i](L[m-1][j]),i,1,4,j,1,4^(m-1))),
       wxplot2d([discrete,append([[0,0]],L[k])],[x,-0.1,1.1],
       [y,-0.1,0.5],[ylabel, "Curva de Koch"]))$
(%i548)Koch(5)
```




```
(%o448)
```

Exemplo 3.17 Faça um programa para gerar cinco iterações da Curva de Peano.

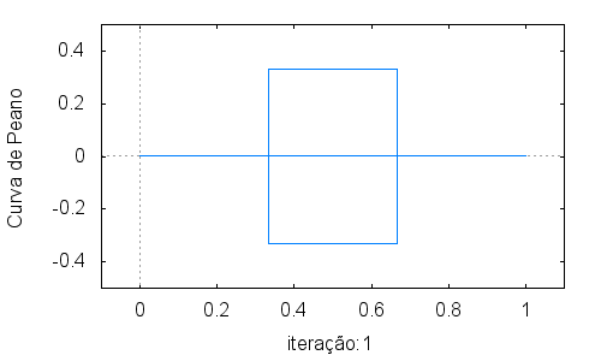
Solução:

Maxima

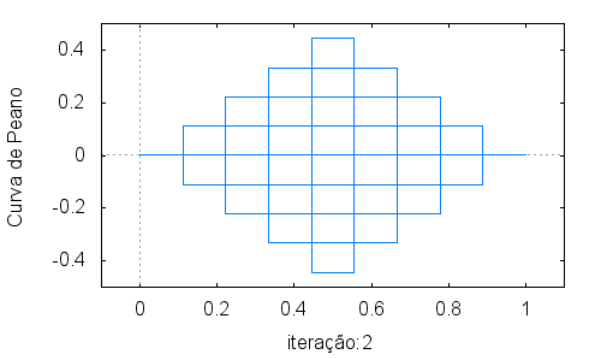
```
(%i549)f[1](X):=1/3*[X[1],X[2]]$
(%i550)f[2](X):=1/3*[-X[2],X[1]]+[1/3,0]$
(%i551)f[3](X):=1/3*[X[1],X[2]]+[1/3,1/3]$
(%i552)f[4](X):=1/3*[X[2],-X[1]]+[2/3,1/3]$
(%i553)f[5](X):=1/3*[-X[1],X[2]]+[2/3,0]$
(%i554)f[6](X):=1/3*[X[2],-X[1]]+[1/3,0]$
(%i555)f[7](X):=1/3*[X[1],X[2]]+[1/3,-1/3]$
(%i556)f[8](X):=1/3*[-X[2],X[1]]+[2/3,-1/3]$
(%i557)f[9](X):=1/3*[X[1],X[2]]+[2/3,0]$
```

Maxima

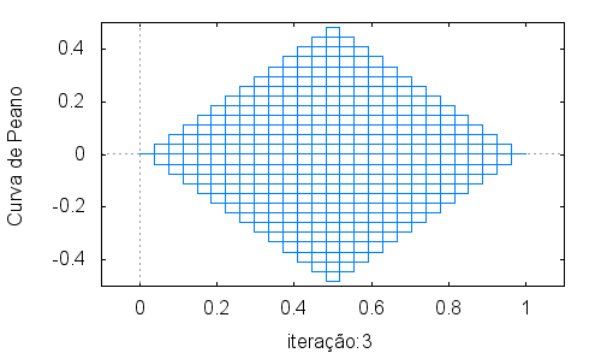
```
(%i558)Peano(n):=(k:n,L[0]:[[1,0]],
       (for m:1 thru k do
           L[m]:create_list(f[i](L[m-1][j]),i,1,9,j,1,9^(m-1))),
       wxplot2d([discrete,append([[0,0]],L[k])],[x,-0.1,1.1],
       [y,-0.5,0.5],[xlabel,"Curva de Peano"],[ylabel,""]))$
(%i559)Peano(1)
```




```
(%o449)
(%i560)Peano(2)
```

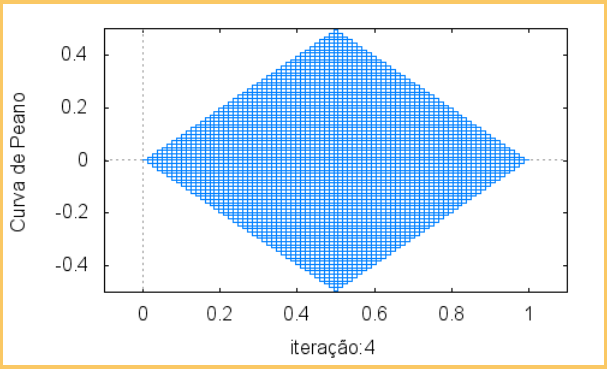



```
(%o450)
(%i561)Peano(3)
```



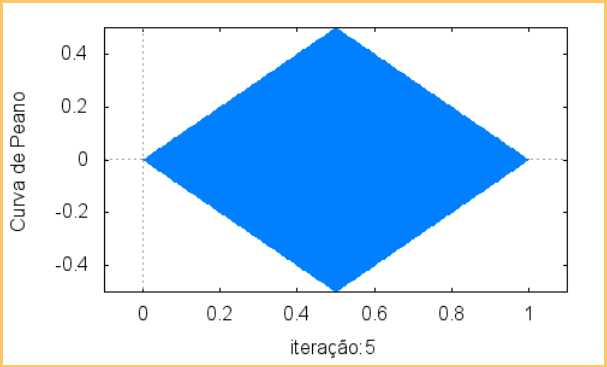
(%o451)

(%i562)Peano(4)



(%o452)

(%i563)Peano(5)



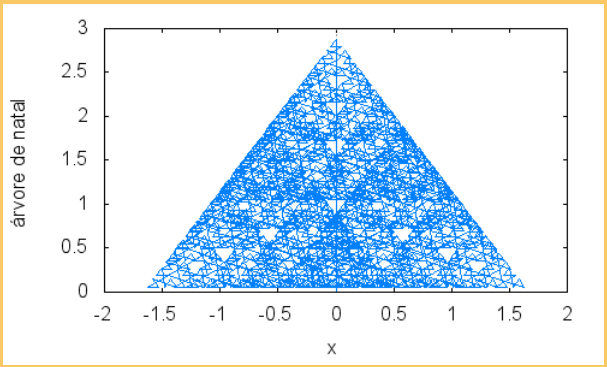
(%o453)

Exemplo 3.18 Neste exemplo programamos um fractal de nossa autoria, o qual chamamos "Árvore de Natal."

Solução: Vamos gerar o fractal "Árvore de Natal".

Maxima

```
(%i564)f[1](X):=2/3*[X[1],X[2]]+[0,1]$
(%i565)f[2](X):=2/3*[float(cos(2*%pi/3))*X[1]-float(sin(2*%pi/3))*X[2],
       float(sin(2*%pi/3))*X[1]+float(cos(2*%pi/3))*X[2]]+[0,1]$
(%i566)f[3](X):=2/3*[float(cos(4*%pi/3))*X[1]-float(sin(4*%pi/3))*X[2],
       float(sin(4*%pi/3))*X[1]+float(cos(4*%pi/3))*X[2]]+[0,1]$
(%i567)fractalnovo(n):=(k:n,L[0]:[[0,1]],
       (for m:1 thru k do
           L[m]:create_list(f[i](L[m-1][j]),i,1,3,j,1,3^(m-1))),
       wxplot2d([discrete,append([[0,0]],L[k])],[x,-2,2],
       [ylabel, "árvore de natal"]))$
(%i568)fractalnovo(7);
```



(%o454)

A.1 Fractais

Curva de Koch

Maxima % Inserindo as IFS%

```
(%i569)f[1](X):= 1/3*X$
(%i570)f[2](X):= 1/3*[X[1]*0.5-float(sqrt(3))*X[2]*0.5,
       float(sqrt(3))*0.5*X[1]+0.5*X[2]]+[1/3,0]$
(%i571)f[3](X):= 1/3*[X[1]*0.5+float(sqrt(3))*X[2]*0.5,
       -float(sqrt(3))*0.5*X[1]+0.5*X[2]]+[1/2,
       float(sqrt(3))/6]$
(%i572)f[4](X):=1/3*[X[1],X[2]]+[2/3,0]$
```

% Pontos iniciais %

```
(%i573)L[0]:[[1,0]]$
```

% Iterações %

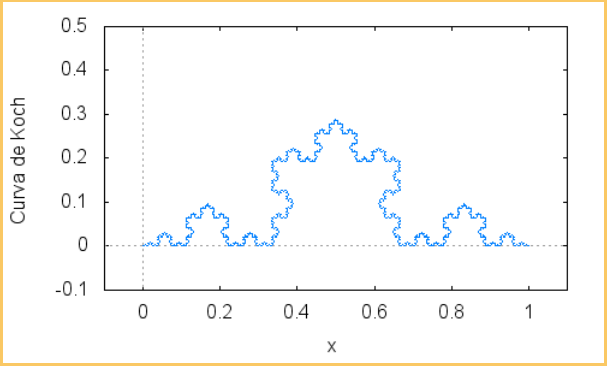
```
(%i574)k = 5$
```

%Lista de pontos%

```
(%i575)for m:1 thru k do L[m]:create_list(f[i](L[m-1][j]),
       i,1,4,j,1,4^(m-1))$
```

%Ligando os pontos por segmentos de reta: construção do fractal%

```
(%i576)plot2d([discrete,append([[0,0]],L[k])],[x,-0.1,1.1],
       [y,-0.1,0.5],[ylabel, "Curva de Koch"]);
```



(%o455)

Referências Bibliográficas

[1] BROWM, D.E., **Graphics in the wxMaxima GUI**, 2012. Disponível em http://emp.byui.edu/BrownD/CAS/wxMaxima/Graphs-Graphics-07-wxMax.pdf. Acesso em março 2016.

[2] CALVO, J.A., **Scientific Programming using Maxima**. Ave Maria University, 2014. Disponível em https://sites.google.com/site/jorgealbertocalvo/maxima. Acesso em março 2016.

[3] CUNHA, M.C., **Métodos Numéricos**. Editora Unicamp, 2000.

[4] EDGAR, G.A., **Topology and Fractal Geometry**. Springer-Verlag, 1992.

[5] FALCONER, K., **Techniques in Fractal Geometry**. Jonh Wiley and Sons, Chichester, 1997.

[6] HAAGER, W., **Graphics with MAXIMA**. ET Elektrotchnik, 2001.

[7] KERNS, G.J., **Multivariables Calculus with Maxima**. 2009. Disponível em http://gkerns.people.ysu.edu/maxima/maximaintro/maximaintro.pdf. Acesso em março 2016.

[8] LARSON, R. e EDWARDS, B.H., **Calculus**. vol 1 and vol 2, Brooks/Cole, 2010.

[9] **Maxima homepage**: http://maxima.sourceforge.net

[10] NERI Junior, E.P., **Elementos da Geometria Fractal**. Universidade Federal do Pará - UFPA, Trabalho de conclusão de curso - TCC, 2009.

[11] RIOTORTO, M.R., **Primeiros passos com o Maxima**. Notas, 2006. Disponível em http://maxima.sourceforge.net/docs/tutorial/pt/max.pdf. Acesso em março 2016.

[12] RIOTORTO, M. R: http://riotorto.users.sourceforge.net, acesso em março 2016.

[13] SOUZA, de P.N., FATEMAN, R.J., MOSES, J. Moses e YAPP, C., **The Maxima Book**. 2002. Disponível em http://maxima.sourceforge.net/docs/maximabook/maximabook-19-Sept-2004.pdf. Acesso em março de 2016.

[14] VAZ, C., **Aprendendo o aplicativo Maxima**. Notas, UFPA, 2009.

[15] WOOLETT, E.L., **Maxima by Exemple**. Notas, 2009. Disponível em http://web.csulb.edu/ woollett/mbe1intro.pdf. Acesso em março de 2016.

Índice Remissivo