

SIMULAÇÕES DE ONDULATÓRIA COM USO DO MAXIMA¹

ELIELZER NUAYED²

29 de julho de 2025

¹This is a footnote.

²<https://github.com/elielzer>

Dedicated to Calvin and Hobbes.

Conteúdo

1 INTRODUÇÃO	3
1.1 Sobre o Maxima	3
1.2 Trabalhando com o wxMaxima	4
1.3 Listas e Matrizes	10
1.4 Construção de Gráficos	11
2 ESTUDO DAS FUNÇÕES TRIGONOMÉTRICAS	17
2.1 As Funções Trigonométricas no Maxima	17
2.2 Transformações Trigonométricas	21
3 DESCRIÇÕES DE MOVIMENTOS ONDULATÓRIOS	27
3.1 O Movimento Ondulatório	29
3.2 Abordagem Matemática	31
3.3 Onda Harmônica como um Modelo Físico	37
4 APLICAÇÕES ONDULATÓRIAS	51
4.1 Corda Estendida	51
4.2 Efeito Doppler	54
4.3 Radiação de um Dipolo Elétrico Oscilante	59
A DICAS SOBRE O MAXIMA	61
A.1 Como Instalar o wxMaxima®	61
A.2 Conhecendo o <i>Interface</i> Gráfico do wxMaxima	66
A.3 Idioma do <i>Interface</i>	69
A.4 Seleção do Compilador	71
A.5 Dicas Diversas	74
A.6 Dicas de Maxima	75

Prefácio

Este livro foi concebido a partir de um *insight* pessoal após eu ler um anúncio sobre uso do *Maxima* no ensino da Física. Então veio-me a ideia de escrever algo nesse sentido. Escolhi sobre ondulatória. O Maxima é introduzido no contexto como ambiente de apoio no momento que seja necessário simular as proposições. Dentre estas abordaremos situações clássicas, pois o objetivo é justamente apresentar o cartel de ferramentas desse software adequadas para transcrever a linguagem da física para representar as situações-problemas.

É provável que o leitor se defronte aqui com uma versão do wxMaxima diferente daquele que esteja instalada em seu computador, porém isso deve não impactar no entendimento do que é exposto no livro, pois o núcleo principal do Maxima mantém-se inalterado quanto a sua finalidade.

INTRODUÇÃO

1.1 Sobre o Maxima

De acordo com descrição no site fabricante, o Maxima é um programa de computador do tipo multiplataforma, ou seja, ele está preparado para trabalhar (compilar) sob diversos sistemas operacionais ou plataformas computacionais. Ainda de acordo com esse site o Maxima é uma versão evoluída de software especialista Macsyma. Os sistemas especialistas foram criados com a finalidade de reproduzir o raciocínio ou expertise de um profissional de alguma área específica do conhecimento.

Assim, o Maxima é um sistema especialista para a manipulação de expressões matemáticas, tanto na forma simbólica como na forma numérica, incluindo:

- diferenciação,
- integração,
- séries de Taylor,
- transformadas de Laplace,
- equações diferenciais ordinárias,
- sistemas de equações lineares,
- polinômios,
- conjuntos,
- listas,
- vetores,
- matrizes e
- tensores.

O que caracteriza a matemática simbólica é de não estar atrelada a um determinado idioma, por exemplo a expressão algébrica: $x + 2 = 7$, tem o mesmo significado em qualquer idioma. Portanto o Maxima é um sistema algébrico computacional para auxílio no cálculo e manipulação da matemática simbólica visado simplificar o esforço de trabalho do estudante ou profissional (professor ou pesquisador).

O Maxima produz também resultados numéricos de alta precisão usando frações exatas, inteiros de precisão arbitrária e números de ponto flutuante de precisão variável. O Maxima pode plotar funções e dados em duas e três dimensões. Utilizaremos em nossos desenvolvimentos a plataforma gráfica do wxMaxima®.

1.2 Trabalhando com o wxMaxima

O Maxima é constituído de *packages* (pacotes). Um pacote vem corresponder a uma biblioteca de comandos programados para uma finalidade específica, e pode ser carregado para o ambiente por meio do comando *load*, alguns pacotes são carregados automaticamente ao abrir o WMaxima.

Cada pacote possui uma chave específica que é usada como argumento do comando *load* Exemplos:

- `load(physical_constants)$`
- `load(worldmap)$`
- `load(unit)$`

Experimente testar o comando *load*, como no exemplo a seguir:

Figura 1.1: Exemplo de como invocar um pacote para o ambiente do Maxima

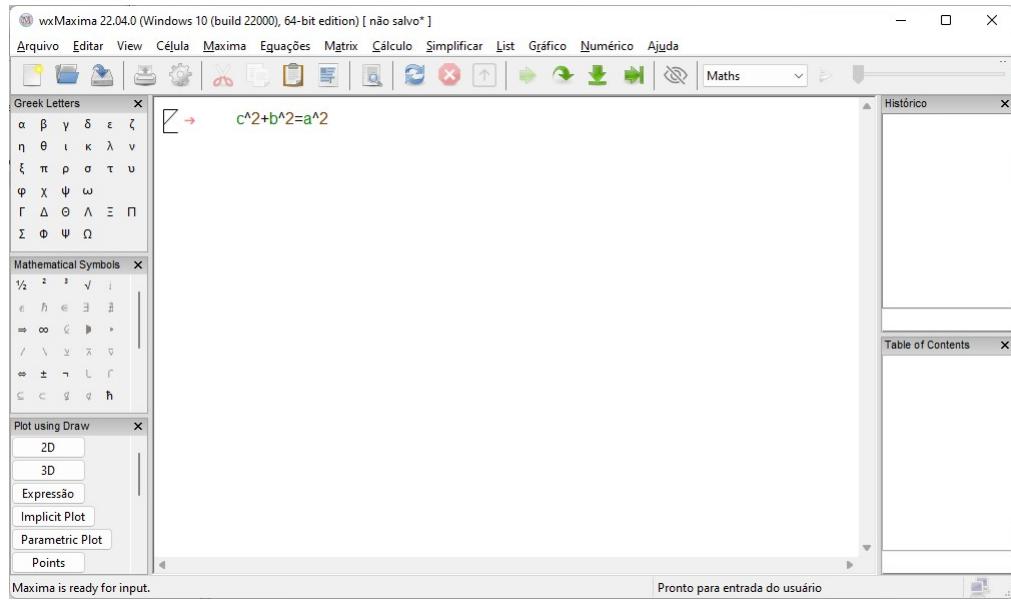
```
(%i5) load(mnewton);
(%o5) C:/maxima-5.47.0/share/maxima/5.47.0/share/mnewton/mnewton.mac
```

A base de conhecimento do Maxima, de acordo com sua documentação, é organizada por meio de um conjunto de categorias, por exemplo: Expressões, Operadores, Avaliação de Expressões, Simplificação, Plotagem, etc.

Há dois tipos de comandos específicos: as *tags* (ou variáveis de controle) e as funções. Algumas funções possuem o mesmo nome das tags. Geralmente as tags servem para comutar dois estados, o *true* ou *false*, um desses será o valor padrão (*default*).

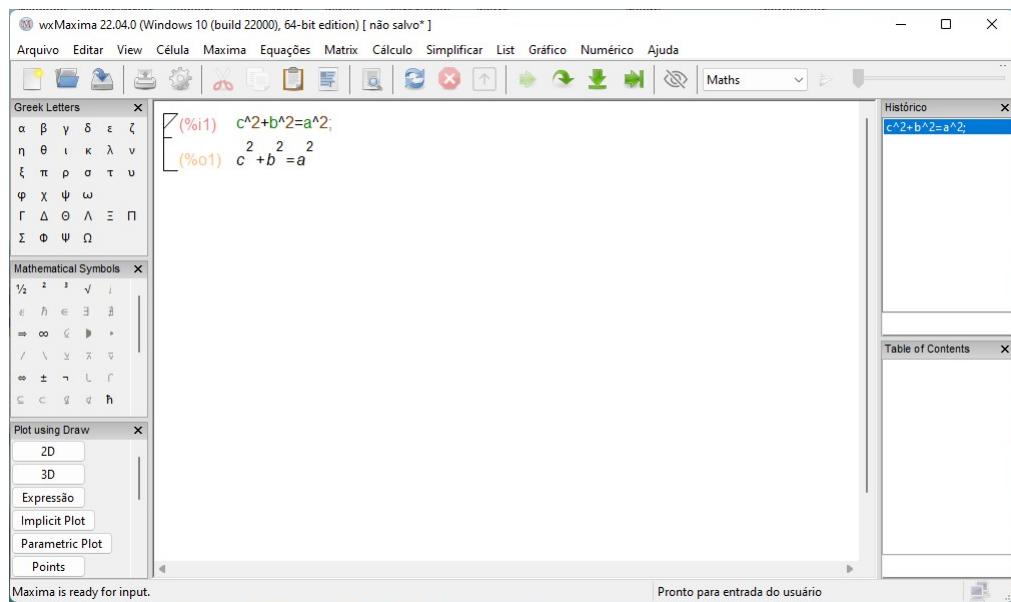
Vamos tomar como primeiro exemplo a fórmula do teorema de Pitágoras. Vamos escrever no *prompt* do wxMaxima a seguinte expressão: $c^2+b^2=a^2$

Figura 1.2: Entrando uma expressão algébrica.



Essa expressão é o input do processo. O Maxima processará essa entrada após acionada a combinação de tecla Shift+Enter do que, em seguida, apresentará uma saída na área do *prompt*, figura 1.3.

Figura 1.3: Saída apresentada após processamento da expressão algébrica entrada.

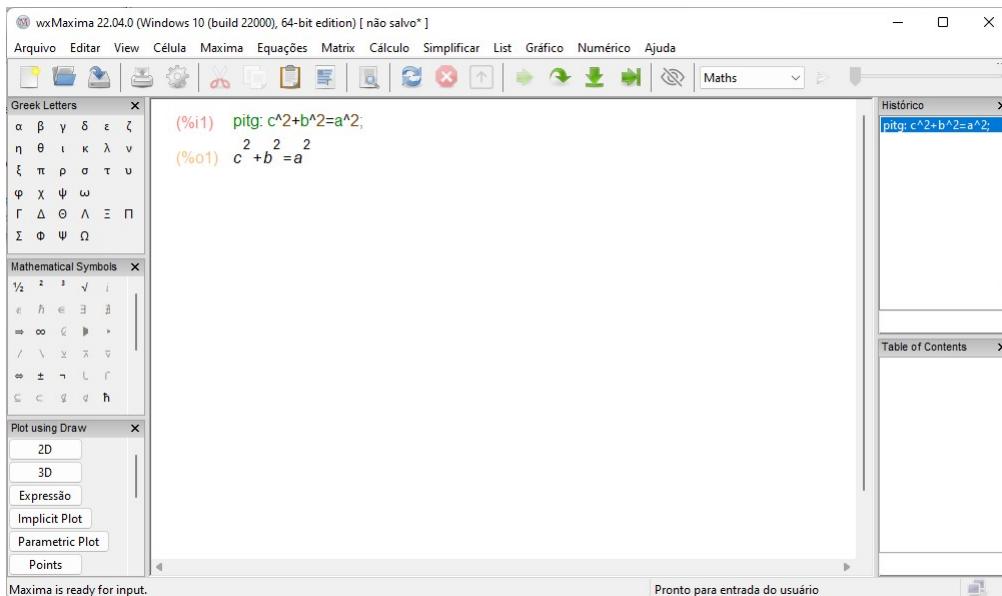


Observe que o resultado é disponibilizado com identificadores chamados **label**, tais que:

- (% i1) é o *label* indica a entrada ou *input*, e
- (% o1) é o *label* que indica a saída ou *output*.

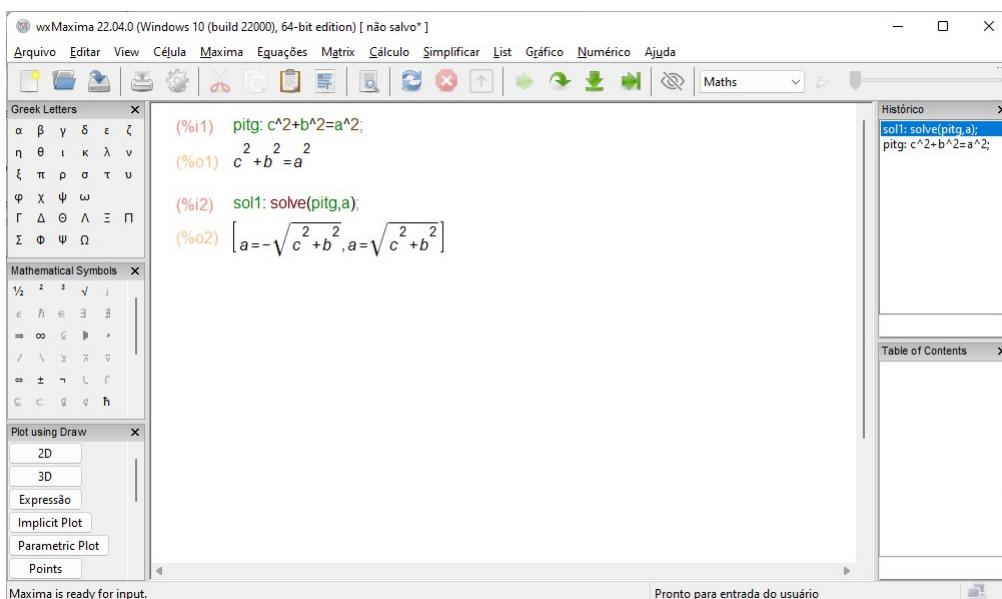
No Maxima é possível *nomear uma expressão* de forma que possa ser reutilizada. Por exemplo, escrevendo no prompt a seguinte expressão: pitg:c^ 2+b^ 2=a^ 2. Observe o resultado disso na figura 1.4.

Figura 1.4: Nomeando expressões.



Essa ação de nomear as expressões algébricas é importante pois facilita a reutilização recorrente da fórmula. Por exemplo, com a seguinte expressão nomeada sol1:solve(pitg,a). O termo *solve* corresponde a uma fórmula (função) interna do Maxima. Os resultados aparecem na figura 1.5.

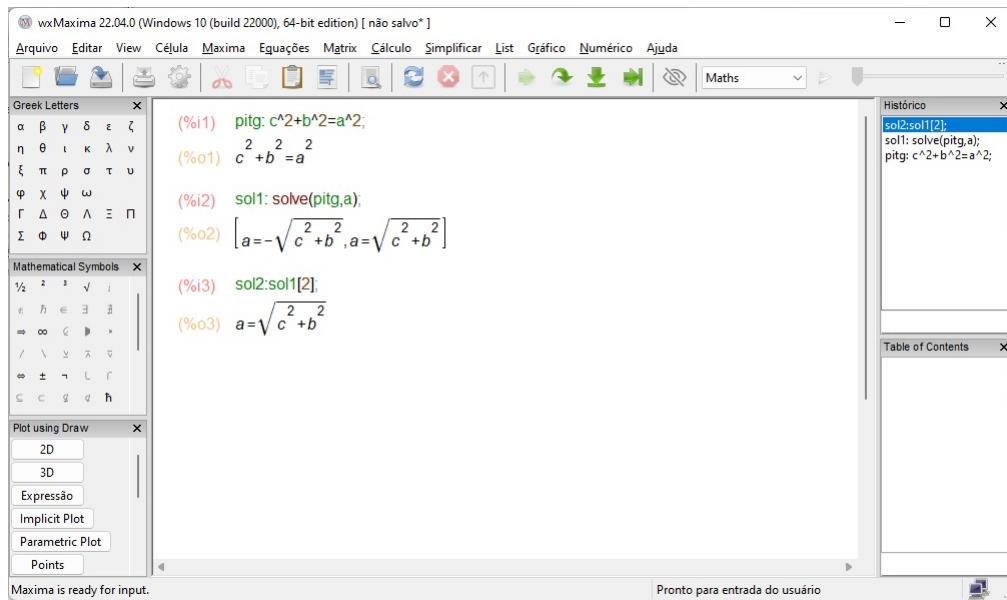
Figura 1.5: Usando expressões.



Ao aplicar um termo *solve*, o Maxima soluciona a expressão. No caso, *pitg* é resolvida para a literal *a* tratando-a como a variável incógnita. Observe que o Maxima retorna dois valores como saída, aquilo que seria obtido, caso fossemos desenvolver a solução manualmente, e tais valores são apresentados no formato de uma lista ou vetor linha de dois elementos.

Uma funcionalidade do Maxima é a possibilidade de recuperar um elemento de um vetor ou lista. Por exemplo, vamos recuperar uma das soluções obtidas do passo anterior. Vamos usar a seguinte sintaxe como *input*: *sol2:sol1[2]*. O resultado é mostrado na figura 1.6

Figura 1.6: Obtendo o valor de um elemento de uma lista.



Observe que o Maxima atribuiu à expressão *sol2* o valor do segundo elemento do vetor *sol1*, ou seja, o valor do elemento de índice [2]

O Maxima também suporta um ambiente de texto, misturado entre os comandos algébricos, recurso muito útil no caso de o usuário precisar de documentar seu trabalho com texto explicativo e formatado. Por exemplo vemos colocar um título introdutório em nossa tela e uma texto explicativo intermediário. O título será um texto a ser inserido antes da primeira célula. Uma célula é o espaço delimitado pelo seletor lateral (parentese “exótico” que aparece à esquerda das expressões) idêntico ao desenho abaixo



O cursor horizontal deve ser transportado com auxílio das teclas de navegação do teclado, ou diretamente com o uso de dispositivo apontador (“mouse”), ver figura ??.

Em seguida digita-se o texto pretendido para título:

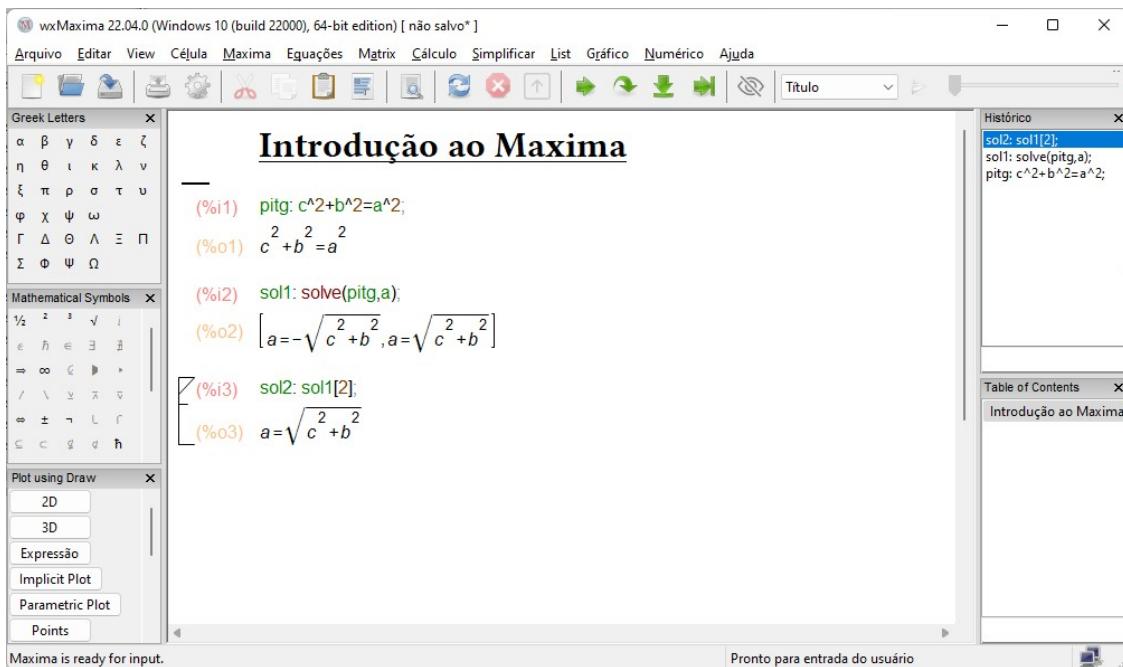
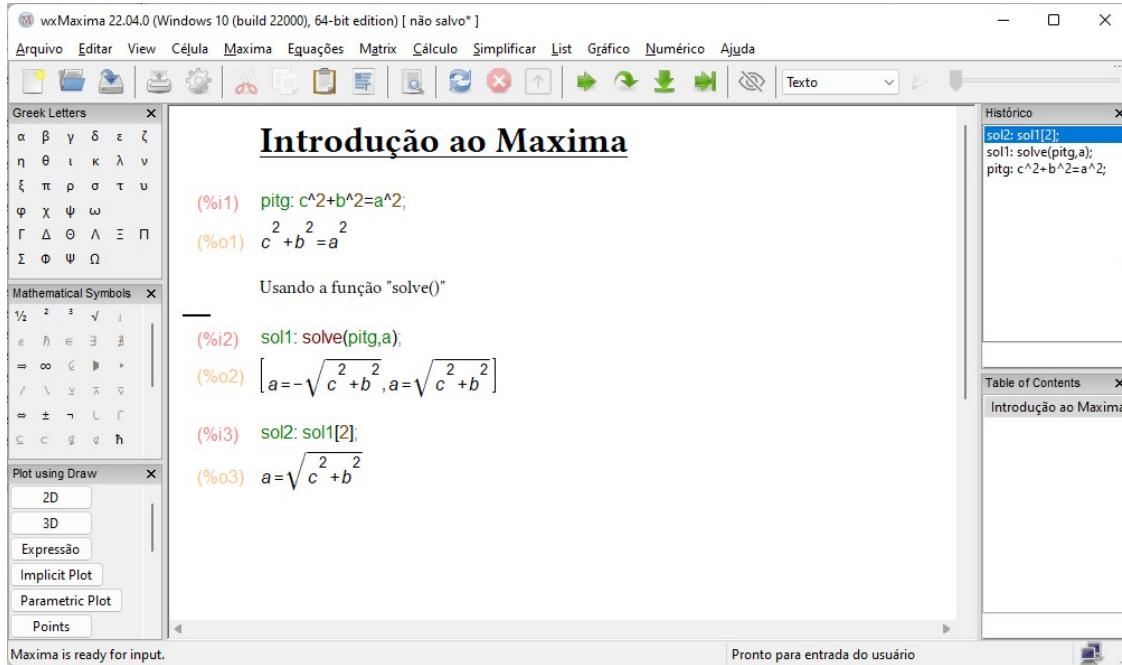


Figura 1.8: Resultado da inserção de título

E, digita-se um texto explicativo pretendido:

Figura 1.9: Resultado da inserção de texto comum



Alguns elementos da estrutura do Maxima são estratégicos, como os *flags*. Os *flags* são os valores que determinadas variáveis internas do Maxima assumem de forma padrão ou conforme a definição do usuário. Exemplos:

- **showtime** Tem valor *false* como padrão. Quando definida para o valor *true*, o tempo de processamento é impresso conjuntamente com cada saída de resultado.

Figura 1.10: Resultado obtido após aplicar configuração *showtime*.

```
(%i4) showtime: false;
      cos(pi/4);

      /*
      serão impressos os tempos para
      os dois comando seguintes
      */
      showtime: true;
      cos(pi/4);

(%o1) false
(%o2) 
$$\frac{1}{\sqrt{2}}$$

Evaluation took 0.0000 seconds (0.0000 elapsed) using 0 bytes.
(%o3) true
Evaluation took 0.0000 seconds (0.0001 elapsed) using 0 bytes.

(%o4) 
$$\frac{1}{\sqrt{2}}$$

```

- `display2d` Tem valor `true` como padrão. Quando definido como `true` isso faz com que o Maxima apresente as expressões matemáticas na sua forma como as vemos nos livros.

Figura 1.11: Uso do variável `display2D`

```
(%i17) display2d: true$  

      ( 2·x^2 + x/2 ) / ( x+ 1/3 );  

      /* será impresso com formato destacado */  

      display2d: false$  

      ( 2·x^2 + x/2 ) / ( x+ 1/3 );  

      /* será impresso com formato linear */  

(%o15) 
$$\frac{2x^2 + \frac{x}{2}}{x + \frac{1}{3}}$$
  

(%o17) (2*x^2+x/2) / (x+1/3)
```

- `powerdisp` Quando definido como `true` expõe os termos de uma soma em ordem crescente de suas potências.

Figura 1.12: Demonstração do flag `powerdisp`.

```
(%i14) powerdisp: true;  

      x^2+x^3+x^4;  

      powerdisp: false;  

      x^2+x^3+x^4;  

(%o1) true  

(%o2)  $x^2 + x^3 + x^4$   

(%o3) false  

(%o4)  $x^4 + x^3 + x^2$ 
```

1.3 Listas e Matrizes

Uma lista é usualmente uma coleção ou conjunto de dados com um significado específico. No Maxima uma lista deve ser entrada iniciando com o operador [e finalizada com o]. Os elementos da lista devem estar separados por sinal de vírgula. A posição ordinal de cada elemento é relevante para efeitos de manipulação da lista, ou seja, para localizar um elemento basta fazer referência à sua posição ordinal na lista.

```
(%i3) l1: [Brasil, Argentina, "2024-01-01 00:00:00", 1000, x, y];
l1[2];
l1[1];
l1 [Brasil,Argentina,2024-01-01 00:00:00,1000,x,y]
(%o2) Argentina
(%o3) Brasil
```

Figura 1.13: Uma lista de dados entrada e apresentada no wxMaxima.

O funcionamento do Maxime é fortemente baseado no processamento em listas e saber trabalhar com lista é fundamental, eis alguns comandos que serão importantes já conhecê-los.

```
(%i6) t: cons("Rad", []);
t: endcons(2, t);
s: endcons(5, t);
s: endcons(15, s);
l: makelist(u^3, u, 5);
sl: append(s,l);
t [Rad]
t [Rad,2]
s [Rad,2,5]
s [Rad,2,5,15]
l [1,8,27,64,125]
sl [Rad,2,5,15,1,8,27,64,125]
```

Figura 1.14: Comandos para manipulação de listas no Maxima.

1.4 Construção de Gráficos

A construção de gráficos por intermédio do wxMaxima consiste na execução das funções de gráfico, que já estão embutidas como comandos da barra de menus.

O gráfico pode ser gerado em referência a um label de saída (%o) ou mesmo uma entrada direta no campo da janela secundária da função de gráfico acionada pelo comando de menu.

Gráfico 2D

Figura 1.15: Gráfico 2D a partir do comando do menu

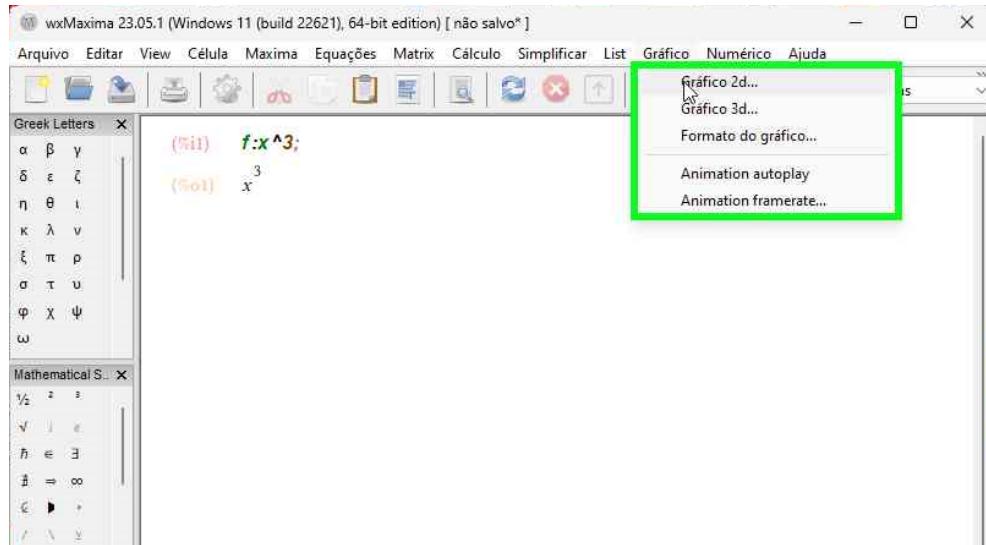


Figura 1.16: Entrando uma expressão rotulada

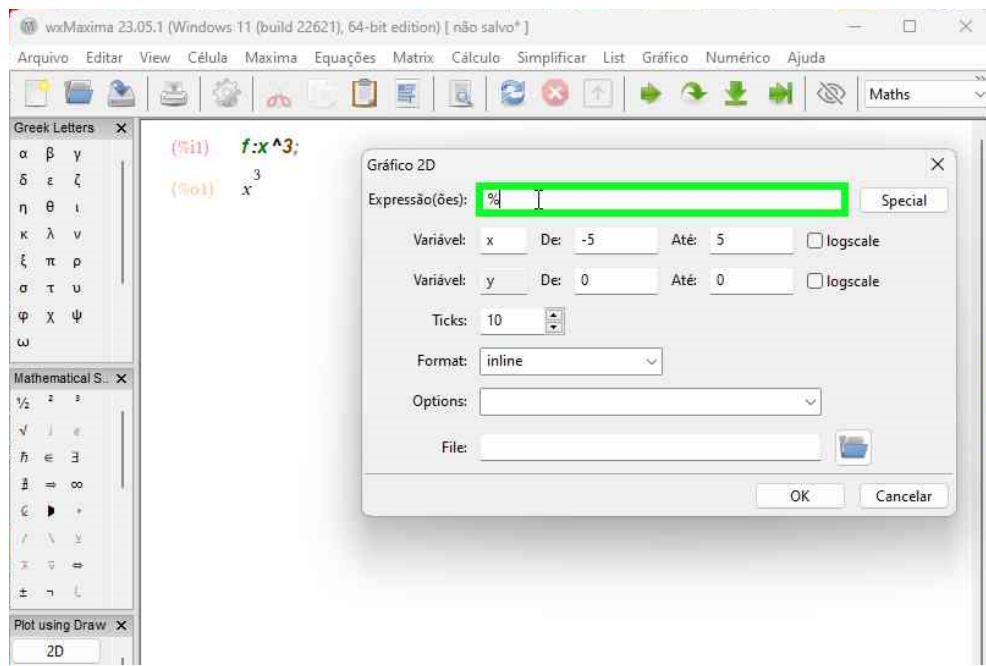


Figura 1.17: Instruir no campo *Format* para o wxMaxima para desenhar o gráfico na mesma janela da planilha

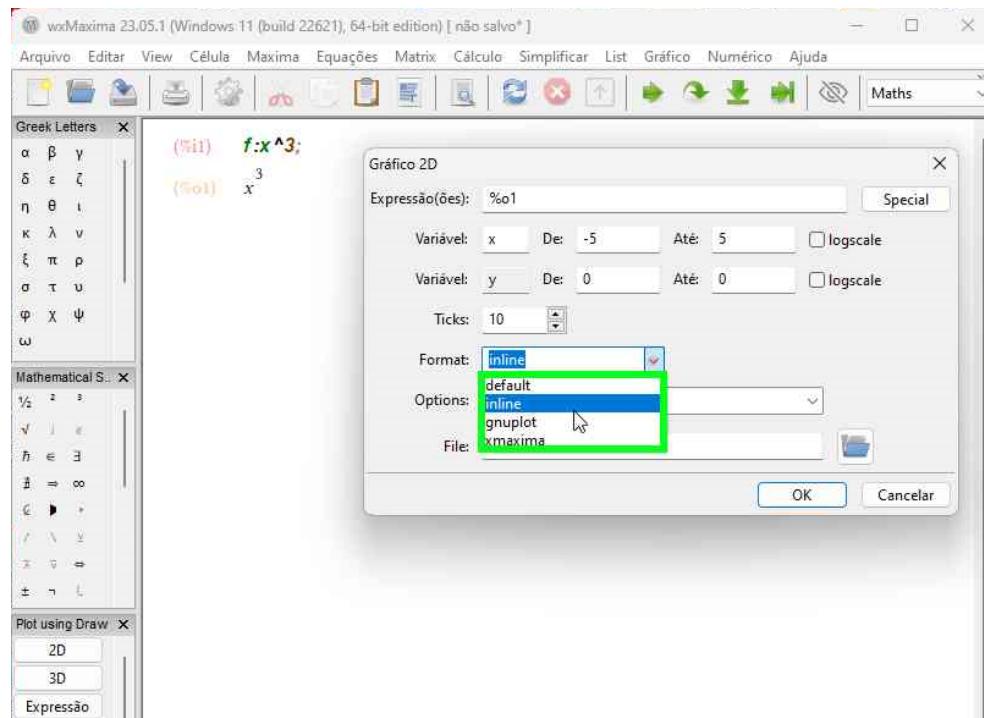


Figura 1.18: Configurando para incluir quadrículas

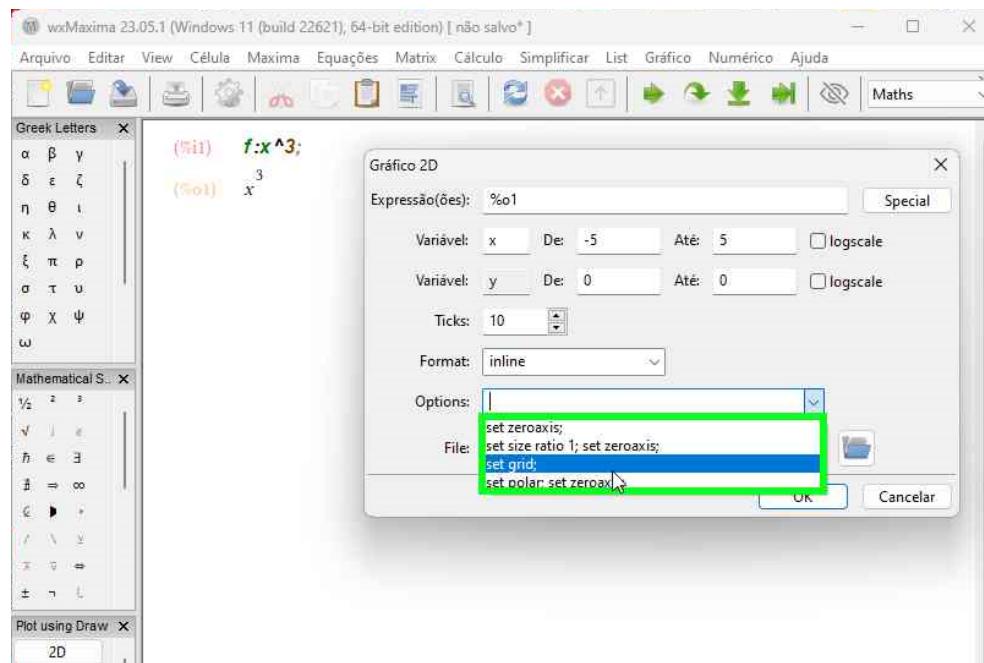


Figura 1.19: Finalizando essa entrada de dados para gerar o gráfico

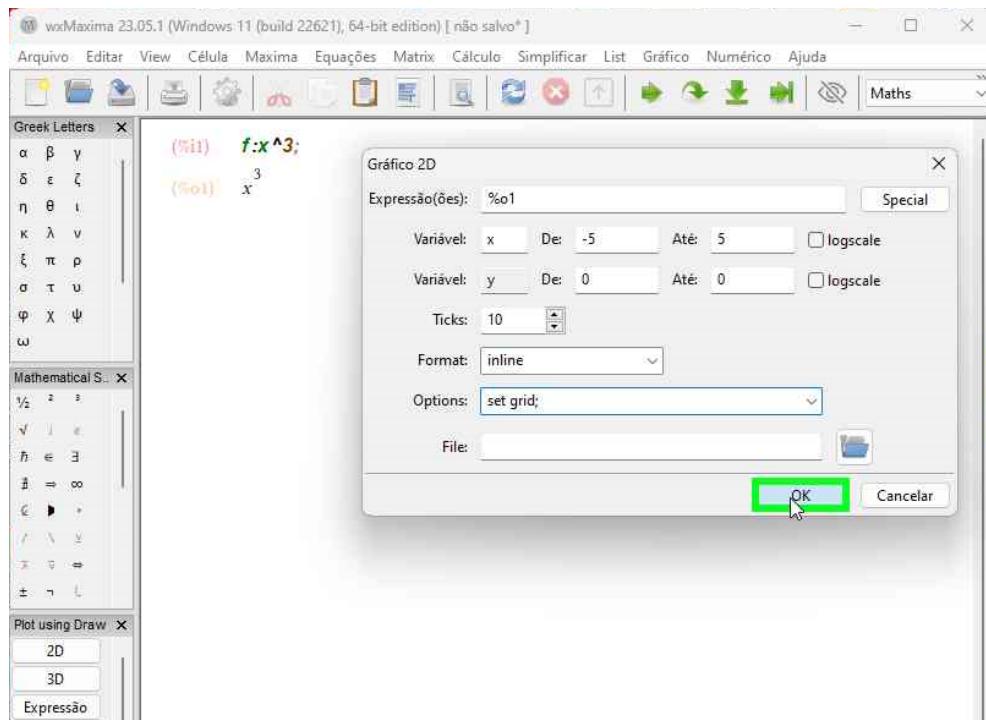
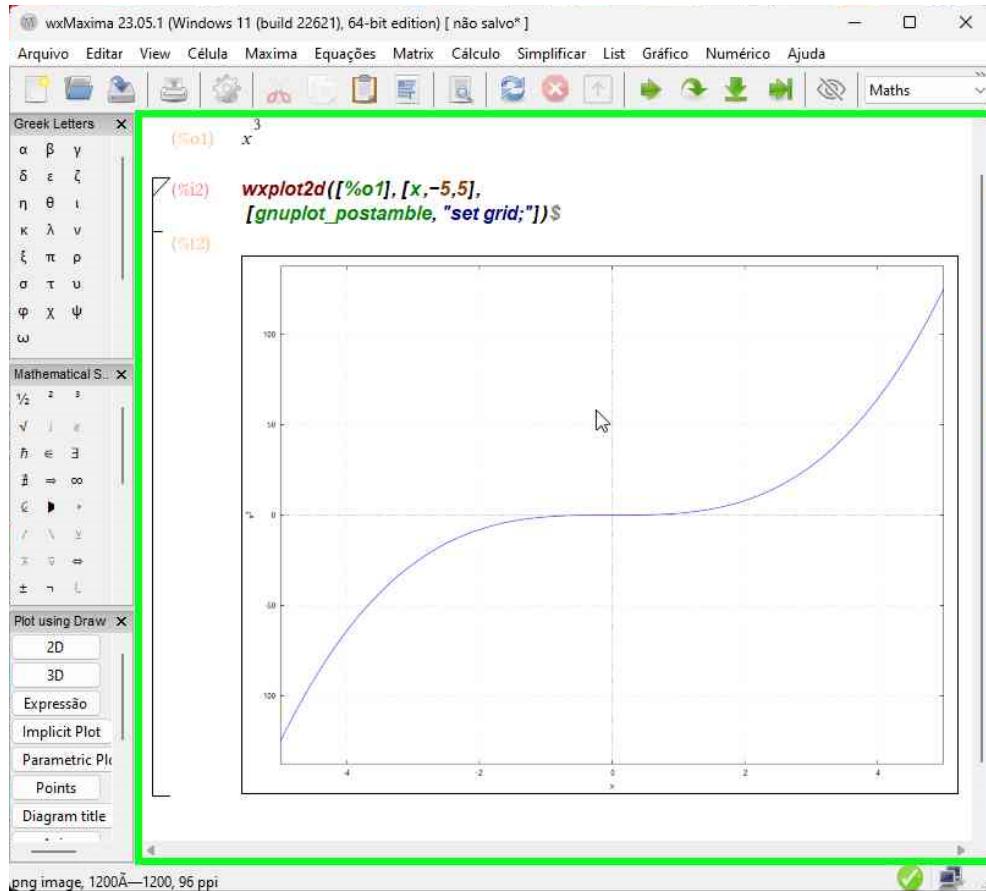


Figura 1.20: O gráfico está gerado e apresentado na planilha



Se fosse selecionada a opção *gnuplot*, seria gerado o mesmo gráfico, porém em uma janela a parte, como mostrado na figura a seguir.

Figura 1.21: Saída de gráfico por janela separada

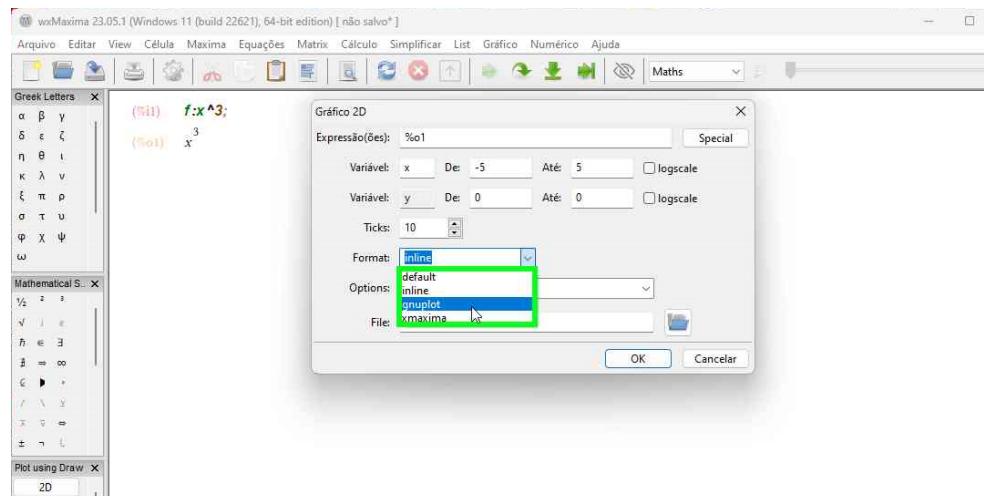


Figura 1.22: Percorrendo agora a mesma sequência de configurações

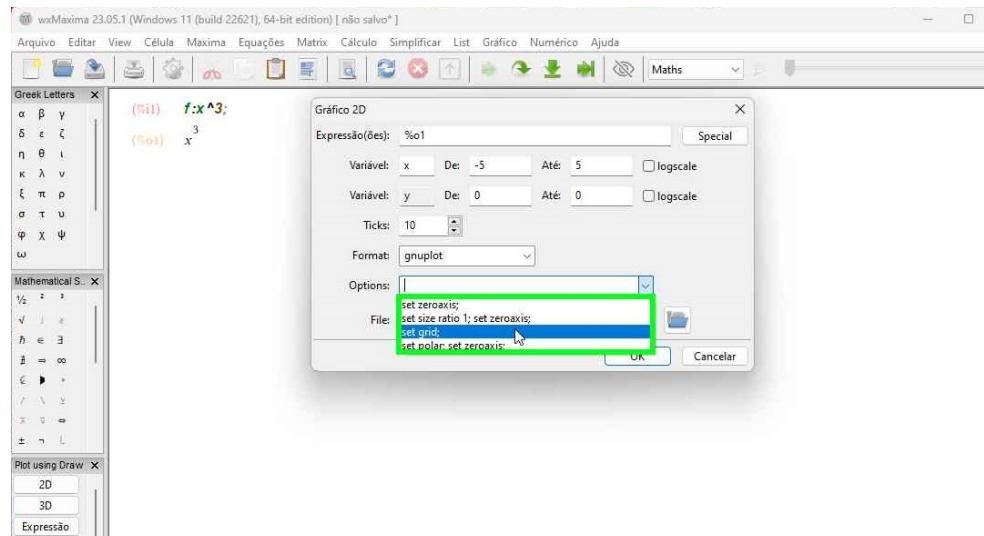


Figura 1.23: Finalizando formato gnuplot

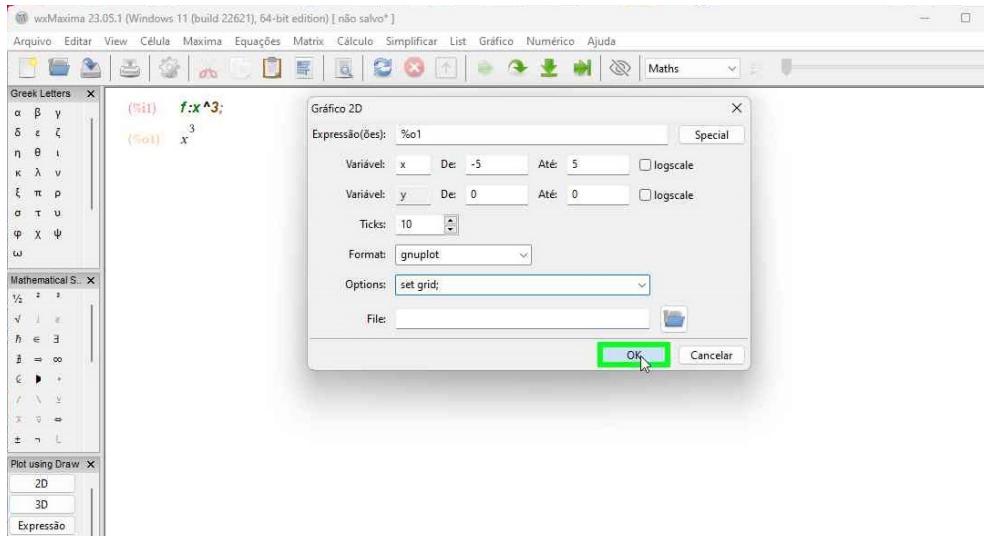
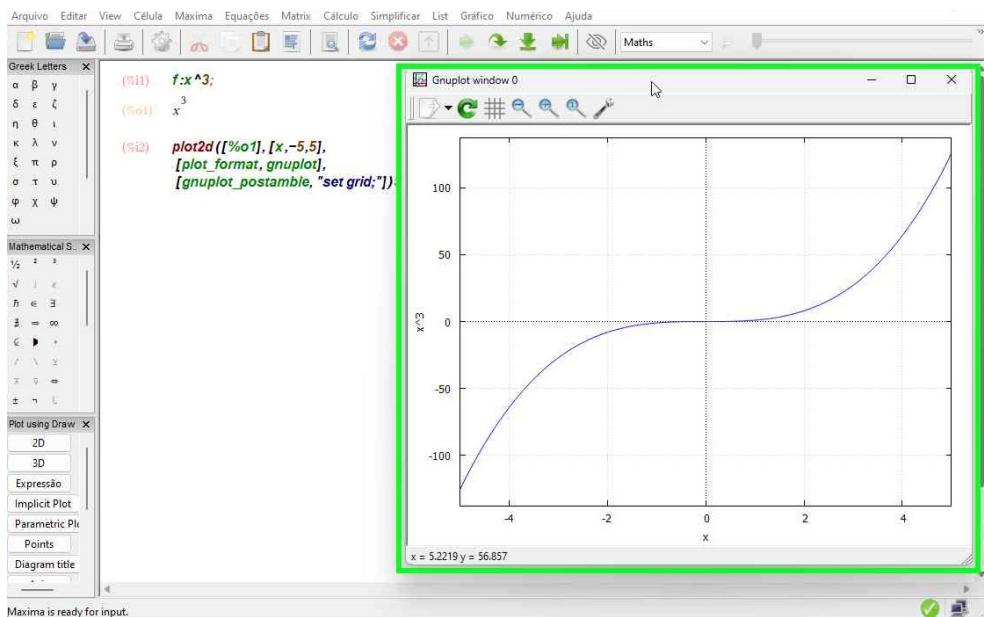


Figura 1.24: A figura mostra o gráfico em detalhe na janela gnuplot



Por hora esses são os conhecimentos sobre o Maxima, indispensáveis para imergir no universo de possibilidades. São conhecimentos básicos e no decorrer dos capítulos estaremos apresentando outros recursos, básicos ou mais complexos, conforme a exigência do contexto.

ESTUDO DAS FUNÇÕES TRIGONOMÉTRICAS

As funções trigonométricas tiveram suas origens no estudo das medidas astronômicas, e elas são de suma importância para as ciências em geral, dado que as suas características permitem de serem utilizadas para representar o comportamento de diversos sistemas reais.

Elas são desenvolvidas por meio do estudo da circunferência de raio unitário com relação às dimensões em um determinado sistema de cordas. A palavra *seno* etimologicamente significa meia corda. As funções trigonométricas também tem relações com a geometria por meio das razões trigonométricas no triângulo retângulo.

As funções trigonométricas são ditas *periódicas*. Uma função $f : \mathbb{R} \rightarrow \mathbb{R}$ é dita periódica com período $T > 0$ se:

1. seu domínio contém $x + T$ sempre que contém x , e se
2. $f(x) = f(x + T)$ para todo x pertencente ao domínio de f .

2.1 As Funções Trigonométricas no Maxima

O Maxima usa := para definir funções. Você irá se referir às trigonométricas usando as seguintes sintaxes.

- seno: $f(u):=\sin(u);$
- cosseno: $f(u):=\cos(u);$
- tangente: $f(u):=\tan(u),$

onde u , ou seja, o argumento deve ser entendido como medida de arco expressa em radianos.

CODE CELL MAXIMA

→ /* Seno */ $f(x):= \sin(x);$
 /* Cosseno */ $g(x):= \cos(x);$
 /* Tangente */ $h(x):= \tan(x);$

```

(%i3) /* Seno */ f(x):= sin(x);
          /* Cosseno */ g(x):= cos(x);
          /* Tangente */ h(x):= tan(x);

(%o1) f(x):=sin(x)
(%o2) g(x):=cos(x)
(%o3) h(x):=tan(x)

```

Outras funções trigonométricas.

[CODE CELL MAXIMA](#)

```

→ /* Secante */ f(t):= sec(t);
   /* Cossecante */ f1(k):= csc(k);
   /* Cotangente */ k(k):= cot(k);
   /* exemplos: */
k: 3; 'k(k) = k(k);
numer: false$ 'f(t:10) = f(t:10);
numer: true$ 'f(t:10) = f(t:10);
'f1(1) = f1(1);

```

```

(%i12) /* Secante */ f(t):= sec(t);
          /* Cossecante */ f1(k):= csc(k);
          /* Cotangente */ k(k):= cot(k);
          /* exemplos: */
k: 3; 'k(k) = k(k);
numer: false$ 'f(t:10) = f(t:10);
numer: true$ 'f(t:10) = f(t:10);
'f1(1) = f1(1);

(%o3) f(t):=sec(t)
(%o4) f1(k):=csc(k)
(%o5) k(k):=cot(k)
k      3
(%o7) k(3)=cot(3)
(%o9) f(10)=sec(10)
(%o11) f(10)=-1.19
(%o12) f1(1)=1.19

```

A constante π , utilizada na trigonometria é representada pelo comando %pi. Como exemplo vamos calcular o seno de $\frac{5\pi}{12}$.

CODE CELL MAXIMA

```
→ kill(all)$
α: 3*%pi/12$
β: 2*%pi/12$
%piargs: true$ /* para usar o valor de π */
t: [ sin(α), cos(α), sin(β), cos(β) ]$
t:~ rat( t[1]*t[4]+ t[2]*t[3] )$
%piargs: false$ /* para mostrar símbolo π */
disp('sin(α+β) = t)$
```

$$\boxed{\begin{aligned} & \text{kill(all)} \\ & \alpha: 3\cdot\pi/12 \\ & \beta: 2\cdot\pi/12 \\ & \text{piargs: true} /* para usar o valor de } \pi \\ & \text{t: [sin}(\alpha\text{), cos}(\alpha\text{), sin}(\beta\text{), cos}(\beta\text{)]\$} \\ & \text{t: rat(t[1]\cdot t[4] + t[2]\cdot t[3])\$} \\ & \text{piargs: false} /* para mostrar símbolo } \pi \\ & \text{disp('sin}(\alpha+\beta\text{) = t)\$} \\ & \sin\left(\frac{5\pi}{12}\right) = \frac{\sqrt{3} + 1}{\sqrt{2}}^3 \end{aligned}}$$

Em muitas aplicações trigonométricas seus argumento são constituídos de funções compostas.

CODE CELL MAXIMA

```
→ kill(all)$
f(x):= sin(a*x + b);
g(x):=f(x)^ 2;
h(x):=f(x)* sin(a_1*x + b_1);
f(1);
'g(1) = g(1) ;
h(%pi/2);
f(x)+g(x);
'f(x)+'g(x) = f(x)+g(x);
```

```
(%i8) kill(all)$
      f(x):= sin(a·x + b);
      g(x):=f(x)^2;
      h(x):=f(x)· sin(a_1·x + b_1);
      f(1);
      'g(1) = g(1) ;
      h(%pi/2);
      f(x)+g(x);
      'f(x)+'g(x) = f(x)+g(x);

(%o1) f(x):=sin(a x+b)
(%o2) g(x):=f(x) ^2
(%o3) h(x):=f(x) sin(a _1 x+b _1)
(%o4) sin(b+a)
(%o5) g(1)=sin(b+a) ^2
(%o6) sin(b+π a /2) sin(b _1 +π a _1 /2)
(%o7) sin(a x+b) ^2 +sin(a x+b)
(%o8) g(x)+f(x)=sin(a x+b) ^2 +sin(a x+b)
```

As funções trigonométricas podem ser manipuladas de forma simbólica e de forma numérica.

CODE CELL MAXIMA

```
→ kill(all)$ numer: false$ %piargs: true$ α: %pi/12$ ARCO: [ α, 2* α, 3*α, 4*α]$ 
block( i: 1, loop1, α: %pi/12, arco: [ α, 2* α, 3*α, 4*α],
seno: append(["Seno"], makelist(sin(α), α, arco)),
cosseno: append(["Cosseno"] , makelist(cos(α), α, arco)),
tangente: append(["Tangente"], makelist(tan(α), α, arco)),
matriz: matrix	append(["ARCO"], ARCO), seno, cosseno, tangente),
disp("Formato de tabela.", matriz),
numer: true, fpprintprec: 3, i: i+1, if i<3 then go(loop1)
)$
```

```
(%i5) kill(all)$ numer: false$ %piargs: true$  $\alpha$ : %pi/12$ ARCO: [  $\alpha$ ,  $2\cdot\alpha$ ,  $3\cdot\alpha$ ,  $4\cdot\alpha$ ]$
block( i: 1, loop1,  $\alpha$ : %pi/12, arco: [  $\alpha$ ,  $2\cdot\alpha$ ,  $3\cdot\alpha$ ,  $4\cdot\alpha$ ],
seno: append(["Seno"], makelist(sin( $\alpha$ ),  $\alpha$ , arco)),
cosseno: append(["Cosseno"] , makelist(cos( $\alpha$ ),  $\alpha$ , arco)),
tangente: append(["Tangente"], makelist(tan( $\alpha$ ),  $\alpha$ , arco)),
matriz: matrix	append([["ARCO"], ARCO], seno, cosseno, tangente),
disp("Formato de tabela.", matriz),
numer: true, fpprintprec: 3, i: i+1, if i<3 then go(loop1)
)$
```

Formato de tabela.

ARCO	$\frac{\pi}{12}$	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$
Seno	$\sin\left(\frac{\pi}{12}\right)$	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}}{2}$
Cosseno	$\cos\left(\frac{\pi}{12}\right)$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$
Tangente	$\tan\left(\frac{\pi}{12}\right)$	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$

Formato de tabela.

ARCO	$\frac{\pi}{12}$	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$
Seno	0.259	0.5	0.707	0.866
Cosseno	0.966	0.866	0.707	0.5
Tangente	0.268	0.577	1.0	1.73

2.2 Transformações Trigonométricas

São relações trigonométricas para as operações com arcos. Temos então que, dados dois arcos α e β temos as seguintes identidades trigonométricas:

$$\begin{aligned} \sin(\alpha + \beta) &= \sin \alpha \cos \beta + \cos \alpha \sin \beta \\ \cos(\alpha + \beta) &= \cos \alpha \cos \beta - \sin \alpha \sin \beta. \end{aligned}$$

Essa forma explícita da soma dos arcos são alcançadas usando a função `trigexpand`.

[CODE CELL MAXIMA](#)

→ `kill(all)$`
`exp1: cos($\alpha + \beta$);`

```

(%i1) kill(all)$
      exp1: cos(α+ β);
(%o1) cos(β+α)

```

→ trigexpand(exp1);

```

(%i1) kill(all)$
      exp1: cos(α+β);
(%o1) cos(β+α)

(%i2) trigexpand(exp1);
(%o2) cos(α)cos(β)-sin(α)sin(β)

```

Usam-se as funções trigonométricas com o argumento arco metade, porém é preciso obedecer certas regras, como mostrado na dica a seguir usando os *flags* halfangles, e %piargs

CODE CELL MAXIMA

→ %piargs: true\$
assume(α>0, α<2*pi)\$
halfangles: false\$
exp1: (sin(α/2))\$
halfangles: true\$
exp1 = sin(α/2) ;

```

(%i6) %piargs: true$
      assume(α>0, α<2·%pi)$
      halfangles: false$
      exp1: (sin(α/2))$
      halfangles: true$
      exp1 = sin(α/2) ;
(%o6) sin(α/2) = √(1 - cos(α)) / √2

```

As funções trigonométricas de arco múltiplo também são possíveis de serem expandidas usando o trigexpand.

CODE CELL MAXIMA

```

→ kill(all) $
/* números inteiros */ m: [2,3,4,5]$

trigexpand: false $
exp1: sin(m*α) $
trigexpand: true $
for i: 1 thru 4 do
(
display (exp1[i] = sin(m[i]*α))
) $

```

(%i5) ***kill(all)*** \$
/ números inteiros */ m: [2,3,4,5]\$*

trigexpand: false \$
exp1: sin(m·α) \$
trigexpand: true \$
for i: 1 thru 4 do
 (
 display (exp1[i] = sin(m[i]·α))
) \$

$\sin(2\alpha) = 2\cos(\alpha)\sin(\alpha)$
 $\sin(3\alpha) = 3\cos(\alpha)^2\sin(\alpha) - \sin(\alpha)^3$
 $\sin(4\alpha) = 4\cos(\alpha)^3\sin(\alpha) - 4\cos(\alpha)\sin(\alpha)^3$
 $\sin(5\alpha) = \sin(\alpha)^5 - 10\cos(\alpha)^2\sin(\alpha)^3 + 5\cos(\alpha)^4\sin(\alpha)$

Algumas funcionalidade permitem obter identidade trigonométricas

CODE CELL MAXIMA

→ eq1: $2 * \sin((\alpha - \beta) / 2) * \cos((\alpha + \beta) / 2)$ \$
 eq2: trigrat(eq1)\$
 display(eq2 = eq1)\$

(%i3) **eq1:** $2 \cdot \sin((\alpha - \beta) / 2) \cdot \cos((\alpha + \beta) / 2)$ \$
eq2: **trigrat** (eq1) \$
display(eq2 = eq1)\$

$$\sin(\alpha) - \sin(\beta) = 2 \sin\left(\frac{\alpha - \beta}{2}\right) \cos\left(\frac{\beta + \alpha}{2}\right)$$

Copie na memória do seu computador as linhas de comando desse exemplo 2.2.1 e cole-as na janela do wxMaxima, avalie com o comando de teclado Shift+Enter e veja o resultado, deverá ser idêntico ao da figura ???. O conteúdo limitado entre '/* */' é interpretado apenas como texto de comentários.

Exercício 2.2.1: No código abaixo acrescente ou modifique comandos de forma que apareçam,

1. uma linha, correspondente aos valores de secante e,
2. uma coluna, ao arco igual a $\frac{5\pi}{12}$.

CODE CELL MAXIMA

```
-> kill(all)$ numer: false$ %piargs: true$  $\alpha$ : %pi/12$ ARCO: [  $\alpha$ , 2*  $\alpha$ , 3* $\alpha$ , 4* $\alpha$ ]$  
block( i: 1, loop1,  $\alpha$ : %pi/12, arco: [  $\alpha$ , 2*  $\alpha$ , 3* $\alpha$ , 4* $\alpha$ ],  
seno: append(["Seno"], makelist(sin( $\alpha$ ),  $\alpha$ , arco)),  
cosseno: append(["Cosseno"] , makelist(cos( $\alpha$ ),  $\alpha$ , arco)),  
tangente: append(["Tangente"], makelist(tan( $\alpha$ ),  $\alpha$ , arco)),  
matriz: matrix	append(["ARCO"], ARCO), seno, cosseno, tangente),  
disp("Formato de tabela.", matriz),  
numer: true, fpprintprec: 3, i: i+1, if i<3 then go(loop1)  
)$
```

DESCRIÇÕES DE MOVIMENTOS ONDULATÓRIOS

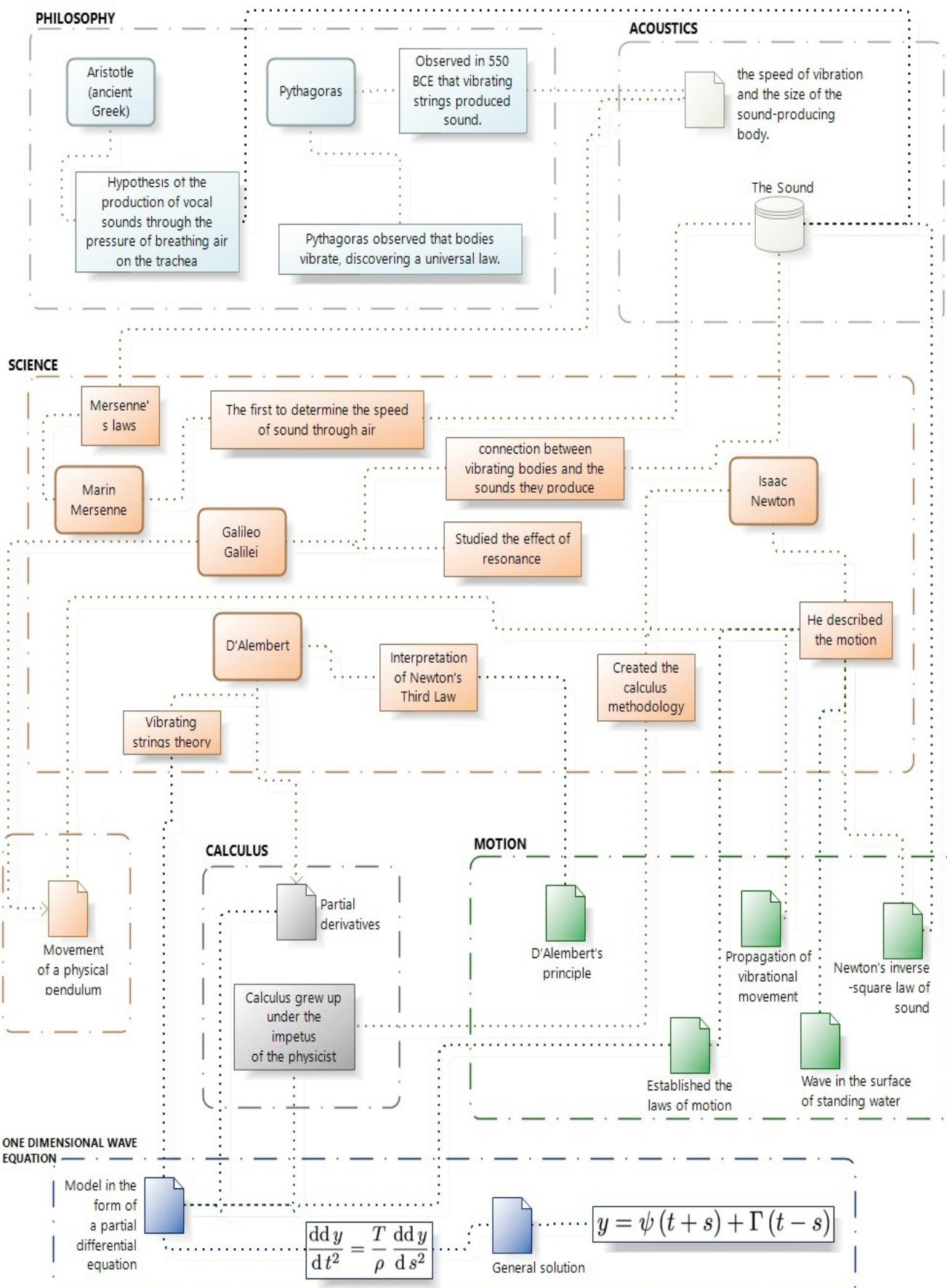
Há várias situações no dia a dia nas quais as coisas acontecem repentinamente, ou inesperadamente, vindo, depois a cessar seus efeitos, ou, em outros casos, repercutem mais além. É comum nos referirmos a isso com a expressão: “isso é apenas uma onda, logo passará”. Isso, claro, devido a que tais situações são transitórias, exemplos: pandemias, endemias, o preço de commodities, tsunamis, terremotos, até notícias ou eventos cotidianos. Algumas vezes tais situações apresentam comportamento oscilante, como na superfície de um líquido em uma piscina, ou mesmo em fontes naturais de água, como em rios, mares e oceanos durante a passagem de uma embarcação.

É importante notar que existem características diferentes em tais situações. Em algumas simplesmente há oscilações ou vibrações e, em outros casos, além de oscilar ou vibrar, essa condição é posteriormente transmitida para outros locais do ambiente longe de sua origem. Neste segundo caso, diz-se que há uma propagação do fenômeno. De tal forma que, *oscilação*, *onda*, *propagação* são termos correlacionados e condicionam a natureza do movimento ondulatório de interesse da física.

Etimologicamente, a palavra *onda* teria a ver com o significado de algo que flutua na água. As descobertas e desenvolvimentos neste tema em seus primórdios derivaram do estudo dos sons musicais. As ondas foram abordadas pela filosofia clássica, onde predominou a Acústica. Diz-se que o estudo moderno de ondas se originou com *Galileu Galilei*, passando então efetivamente para o domínio das ciências.

A figura 3.1 apresenta as contribuições de vários investigadores numa perspectiva histórica do conhecimento científico desta temática.

Figura 3.1: Representação da evolução do conhecimento sobre ondas.



3.1 O Movimento Ondulatório

Vamos supor uma haste fina de metal em que alguém bata com uma ferramente sobre sua extremidade então outra pessoa oposta sentirá as vibrações dos impactos cujo efeito seria, dada sua elasticidade, devido a sucessão de pequenos movimentos MHS sobre extremidade observada. Porém, na medida em que as estocadas forem mais intensas parte da energia fornecida será acumulada localmente na haste originando uma sucessão de movimentos abruptos avançando pela extensão até isso ser percebido na extremidade oposta. Ou seja, nessa condição houve uma transferência de energia inicialmente acumulada na elasticidade de uma seção do material, na forma de um *pacote* e que, por efeito secundário, ela entrou em propagação nas seções subsequentes. Essa transferência da energia mecânica assim descrito é tipicamente um movimento ondulatório.

O movimento ondulatório comprehende tudo que ocorre como consequência de uma perturbação que se propaga além de seu local e com a condição de apresentar nisso inerentes efeitos de difração e interferência. Nesse sentido, e uma vez que se estabeleça um sistema de referência, então, um observador, sob a perspectiva de um experimento ou uma hipótese, descreverá seus resultados por meio de uma terminologia, a qual traduz, de certa forma, as propriedades do movimento ondulatório:

- A *perturbação*, que é a ação inicial, intencional ou não, que leva a efeito a produção do que vem a ser uma onda. O local onde houve a perturbação marca origem ou a *fonte*. Então uma onda diz-se que é *produzida*.
- Uma vez que a ação perturbatória resultou em uma onda sobre o meio ou campo então essa onda é chamada de um *pulso*. O pulso é *enviado* e possui perspectiva de direção, tempo e espaço. Portanto o pulso viaja ou caminha.
- A atuação contínua da perturbação leva a efeito uma sucessão de pulsos, ou *trem de ondas*. A ação periódica da fonte gera um *trem de ondas periódico*.
- Todos os pontos do meio ou campo sujeitos a uma mesma onda possuem o mesmo *estado de movimento*. Tais pontos constituem uma *frente de onda*.
- A curva ou superfície decorrente da frente de onda define o *perfil da frente de onda*. Os perfis de onda podem ser descritos por meio de funções matemáticas.
- A direção é um aspecto importante na descrição do movimento ondulatório e, como a frente de onda possui uma conotação geométrica, reside sobre ela o conceito de *raio*, que é uma linha reta perpendicular à curva ou superfície definida pela frente de onda. O raio indica a *direção do movimento ondulatório* ou de *propagação*.
- As partículas do meio ou linhas de campo afetadas pelo pulso ou onda efetuam apenas deslocamento temporário ou oscilação em torno de sua posição original ou de equilíbrio, ou seja, não há um deslocamento resultante para elas. Portanto, a *amplitude* se configura pela medida do deslocamento de um elemento do meio ou linha de campo submetido ao movimento perturbatório a partir da posição de equilíbrio.
- É de grande interesse para o estudo da Ondulatória o perfil *senoidal* ou harmônico. O que leva a abordagem do *estado da onda*, que pode ser
 - estacionário;
 - em propagação.

- O tipo de onda quanto ao seu princípio físico governante, que devem ser: *ondas mecânicas, eletromagnéticas e materiais*.

Máxwell descobriu que o campo elétrico dependente do tempo propaga-se no vácuo e tal condição pode ser interpretada pela ondulatória. Tem-se portanto a teoria das ondas eletromagnéticas. Um dos mecanismo para a produção de ondas eletromagnética é o dipolo elétrico e um outro é magnético, ou seja são fontes desses tipos de onda.

3.2 Abordagem Matemática

O movimento ondulatório está vinculado necessariamente à uma natureza perturbatória. Isso sugere abordagens matemáticas similares se considerarmos aspectos oscilatórios. Na física, oscilações tem como modelo matemático padrão o movimento harmônico simples (MHS), o qual é retratado como o deslocamento periódico de uma partícula em torno do seu ponto de equilíbrio medido como uma função senoidal do tempo.

Se considerarmos tal situação de oscilação ocorrendo sobre o eixo X do plano cartesiano e em torno da origem então o deslocamento será

$$x(t) = A \sin \omega t.$$

No caso mais geral de o movimento não se iniciar na origem então teremos

$$x(0) = A' \neq 0,$$

e como, nesse caso, $|A'| < |A|$, então podemos fazer

$$x(0) = A \sin \alpha,$$

e assim a expressão mais geral seria

$$x(t) = A \sin (\omega t + \alpha),$$

onde o argumento $(\omega t + \alpha)$ é um valor que se dá em radianos trigonométricos, inclusive, isso permite a representação do MHS pela técnica de *vetor girante*. Isto é possível devido à relação que existe entre o movimento circular uniforme (MCU) e o oscilatório.

Na medida em que a perturbação oscilante esteja associada ao movimento ondulatório subsequente então podemos usar o MHS para representar seu perfil de onda.

Seja então uma variável independente θ , em radianos, coincidindo com o eixo das abcissas do plano cartesiano, então teríamos a expressão da função seno na forma:

$$\xi(\theta) = \xi_0 \sin \theta,$$

CODE CELL MAXIMA

```
-> kill(all)$
declare_subscripted(xi_0)$
disp(box( xi(theta):= xi_0 * sin(theta)))$
```

```
(%i2) kill(all)$
declare_subscripted(xi_0)$
disp(box( xi(theta):= xi_0 * sin(theta)))$
```

$$\boxed{\xi(\theta) := \xi_0 \sin(\theta)}$$

onde, $\xi(\theta)$ é o valor da função, corresponde a uma ordenada ou cota, como na figura 3.2, e que poderia estar representando a *amplitude local* de uma propriedade física. E o arco, θ , do qual está sendo calculado o seno, é a *fase* onde se apresenta a evolução da onda. É um valor de arco representando um ponto

sobre o eixo de propagação cuja a amplitude associada será sempre de mesmo valor à medida que se evolua a função ξ por meio de ciclos iguais a 2π rad, ou seja, $\Theta + 2\pi$.

CODE CELL MAXIMA

```
→ kill(all)$declare_subscripted(ξ_0)$ ξ(θ):= ξ_0*sin(θ)$ ξ_0: [10, 15,-5]$ξ(θ):= ξ_0[1]*sin(θ)$
wxdraw2d( axis_top = false, axis_bottom = false, xtics_axis = true,
explicit(ξ(θ),θ,0,2*%pi), explicit(0,θ,0,2*%pi), explicit(ξ_0[1],θ,0,2*%pi),
explicit(-ξ_0[1],θ,0,2*%pi), point_type = filled_circle, point_size = 2,
points_joined = true, points([ [2.5,ξ(2.5)], [2.5,0] ]), /* Ordenda da função em θ */
grid=[1,1], color = _blue, font = "Arial", font_size = 20,
label([concat( "y = ξ_0"),1,ξ_0[1]-.2] ), label([concat( "y = - ξ_0"),1,-ξ_0[1]+.5] ),
label([ "θ", 2.5, -0.5 ]), label([concat( "ξ(θ)=", ξ_0[1], "*sin(θ)" ), 3.1,ξ(2.5) ]) )$
```

```
(%i5) kill(all)$declare_subscripted(ξ_0)$ ξ(θ):= ξ_0·sin(θ)$ ξ_0: [10, 15,-5]$ξ(θ):= ξ_0[1]·sin(θ)$
wxdraw2d( axis_top = false, axis_bottom = false, xtics_axis = true,
explicit(ξ(θ),θ,0,2·%pi), explicit(0,θ,0,2·%pi), explicit(ξ_0[1],θ,0,2·%pi),
explicit(-ξ_0[1],θ,0,2·%pi), point_type = filled_circle, point_size = 2,
points_joined = true, points([ [2.5,ξ(2.5)], [2.5,0] ]), /* Ordenda da função em θ */
grid=[1,1], color = _blue, font = "Arial", font_size = 20,
label([concat( "y = ξ_0"),1,ξ_0[1]-.2] ), label([concat( "y = - ξ_0"),1,-ξ_0[1]+.5] ),
label([ "θ", 2.5, -0.5 ]), label([concat( "ξ(θ)=", ξ_0[1], "*sin(θ)" ), 3.1,ξ(2.5) ]) )$
```

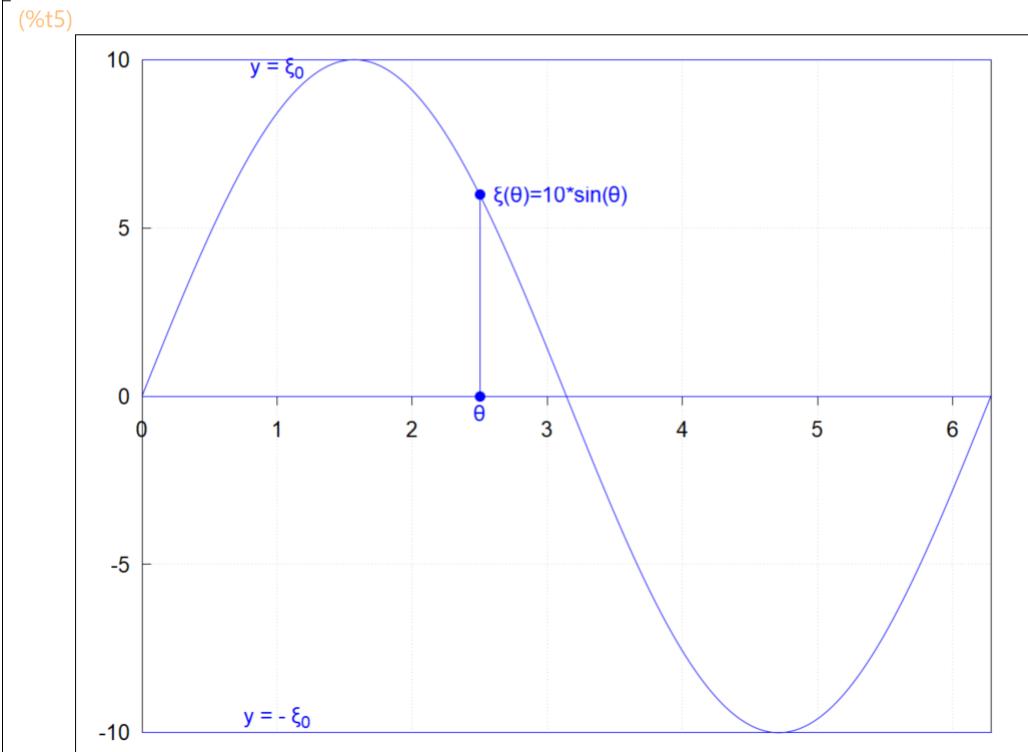


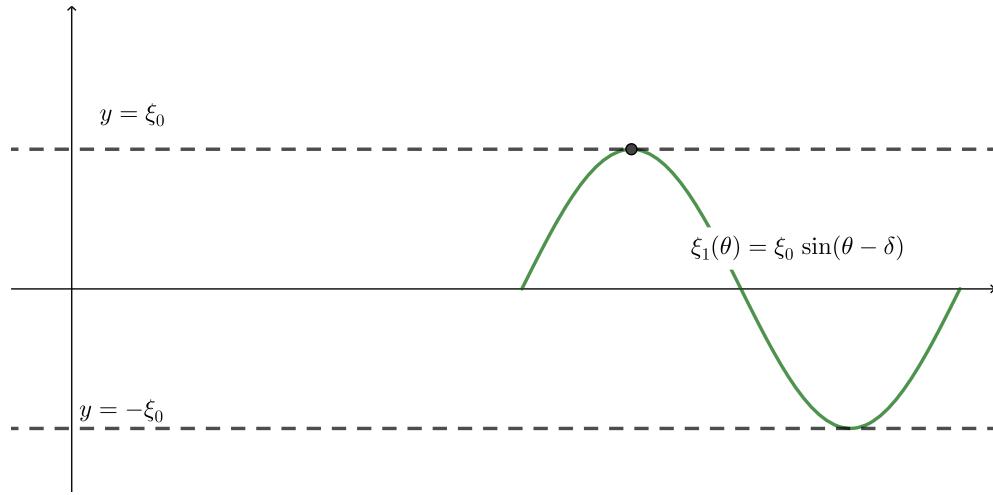
Figura 3.2: Perfil da função harmônica.

Vamos supor que tenhamos agora uma outra função na forma

$$\xi_1(\theta) = \xi_0 \sin (\theta - \delta).$$

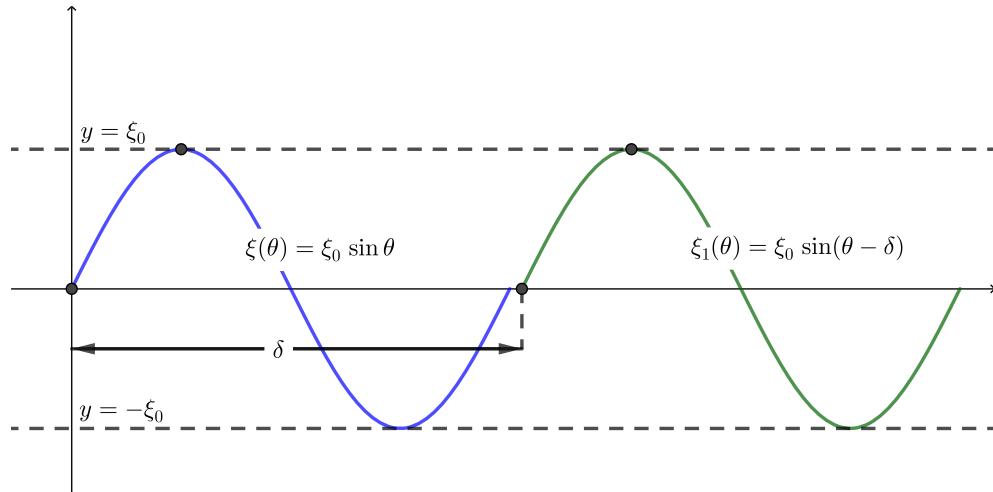
Esta seria representada na forma do gráfico da figura 3.3, ou seja, uma *senoide* deslocada da origem dos eixos à direita.

Figura 3.3: Gráfico de um perfil de uma função harmônica deslocado da origem do eixo das abcissas.



Os dois gráficos anteriores, agora compartilhando do mesmo conjunto de eixos no desenho, como mostrado na figura 3.4, permite-nos vislumbrar que ambos são semelhantes na forma, estão deslocados no espaço (eixo θ) de uma distância igual a δ . Assim, diz-se que a função ξ_1 é uma translação de ξ .

Figura 3.4: Confrontação dos gráficos de duas funções harmônica inter deslocadas.



Ressalte-se que estamos tratando com uma *variável independente* que representa valores em radianos, qual seja, a variável θ . Então daqui em diante temos que associar uma grandeza para melhor representar os sistemas físicos. Podemos fazê-lo por meio da transformação matemática:

$$\theta = k x - \delta,$$

onde a variável x estará com aí presente para representar de fato uma posição, agora, no eixo X. Isso nos permite propor as seguintes descrições físicas:

- o traçado da senoide como que representando o efeito de uma onda tem origem na posição onde uma fonte impõe oscilação ou vibração ao meio.

- a partir daí o meio absorve e ao mesmo tempo transmite a energia e quantidade de movimento para sua vizinhança também na forma de oscilações locais.

A presença de um fenômeno físico ondulatório, cujo perfil da onda estaria sendo esboçado pela senoide do lado direito da figura 3.4, sugere que ele, o perfil, é uma repetição do que está à esquerda, ou seja, o perfil atual é uma repetição gráfica e manterá a sua extensão unidirecional medida em *radianos*, durante as repetições sucessivas. Então, para um determinado valor de x :

- é verdadeira a proposição,

$$\sin(2\pi + kx - \delta) = \sin(kx - \delta),$$

qualquer que seja o valor de x .

- e, consequentemente, estender tal argumentação diretamente para a expressão do perfil de onda senoidal.

De forma algébrica, temos:

$$\xi(x) = \xi_0 \sin(kx - \delta)$$

ou

$$\xi(x) = \xi_0 \sin k\left(x - \frac{\delta}{k}\right), \quad (3.1)$$

e da mesma forma

$$\xi(x) = \xi_0 \sin k\left(\frac{2\pi}{k} + x - \frac{\delta}{k}\right),$$

Se considerarmos neste momento, o k como um coeficiente contante, então a quantidade $\frac{2\pi}{k}$ também o será. Tal quantidade, a qual é do mesmo tipo de grandeza de x , é denominada *comprimento de onda*¹, ou seja, é o comprimento do perfil da onda ao longo do eixo X. Temos então,

$$\lambda = \frac{2\pi}{k}.$$

Vamos a um exemplo prático representado no ambiente Maxima.

CODE CELL MAXIMA

```
→ A: [\xi_0: 7, k: 3, δ :20] /* valores arbitrários. */ λ: 2*%pi/k$  

ξ(x):= A[1]* sin(A[2]*( λ + x - A[3]/A[2]))$ disp( 'λ = λ, ξ(x) = ξ(x))$  

wxplot2d([\xi(x)], [x,0, x_1: 6], [style, lines], [color, brown], nobox )$
```

¹Observar que essa quantidade corresponde à constante T apresentada no preâmbulo do capítulo 2, a qual caracteriza uma função periódica.

```
(%i5) A: [\xi_0: 7, k: 3, \delta :20] /* valores arbitrários. */$ \lambda: 2-%pi/k$  

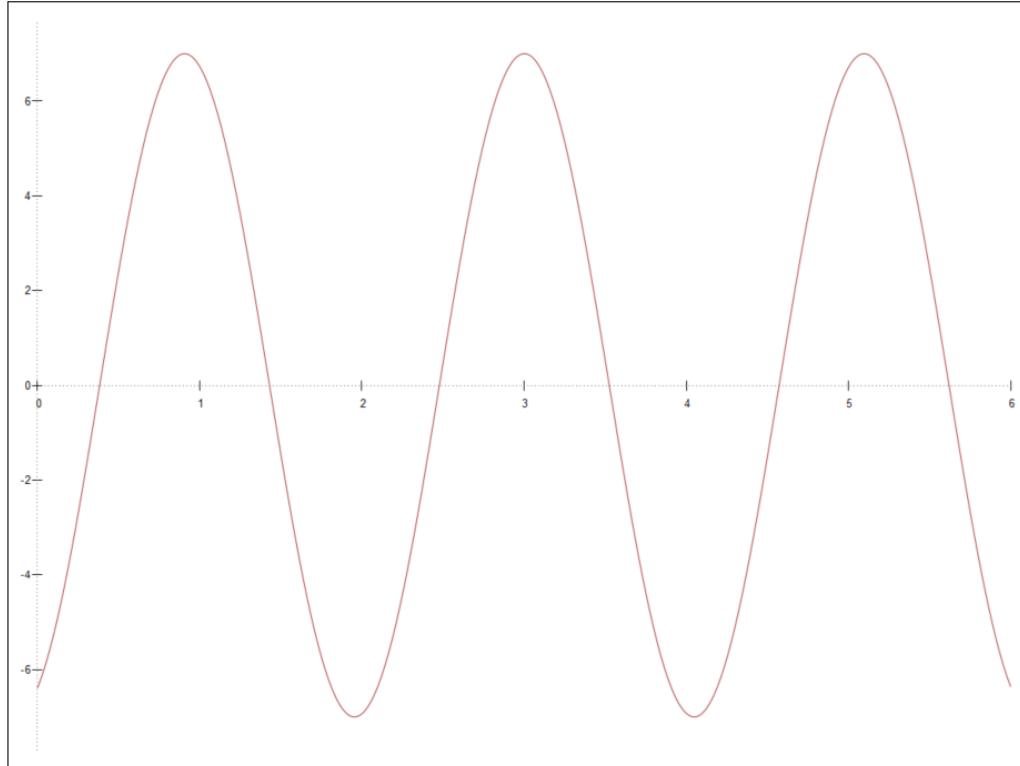
\xi(x):= A[1]* sin(A[2]*( \lambda + x - A[3]/A[2] ))$ disp( '\lambda = \lambda, \xi(x) = \xi(x))$  

wxplot2d([\xi(x)], [x,0, x_1: 6], [style, lines], [color, brown], nobox )$
```

$$\lambda = \frac{2\pi}{3}$$

$$7 \sin\left(3\left(x + \frac{2\pi}{3} - \frac{20}{3}\right)\right) = 7 \sin\left(3\left(x + \frac{2\pi}{3} - \frac{20}{3}\right)\right)$$

(%)



CODE CELL MAXIMA

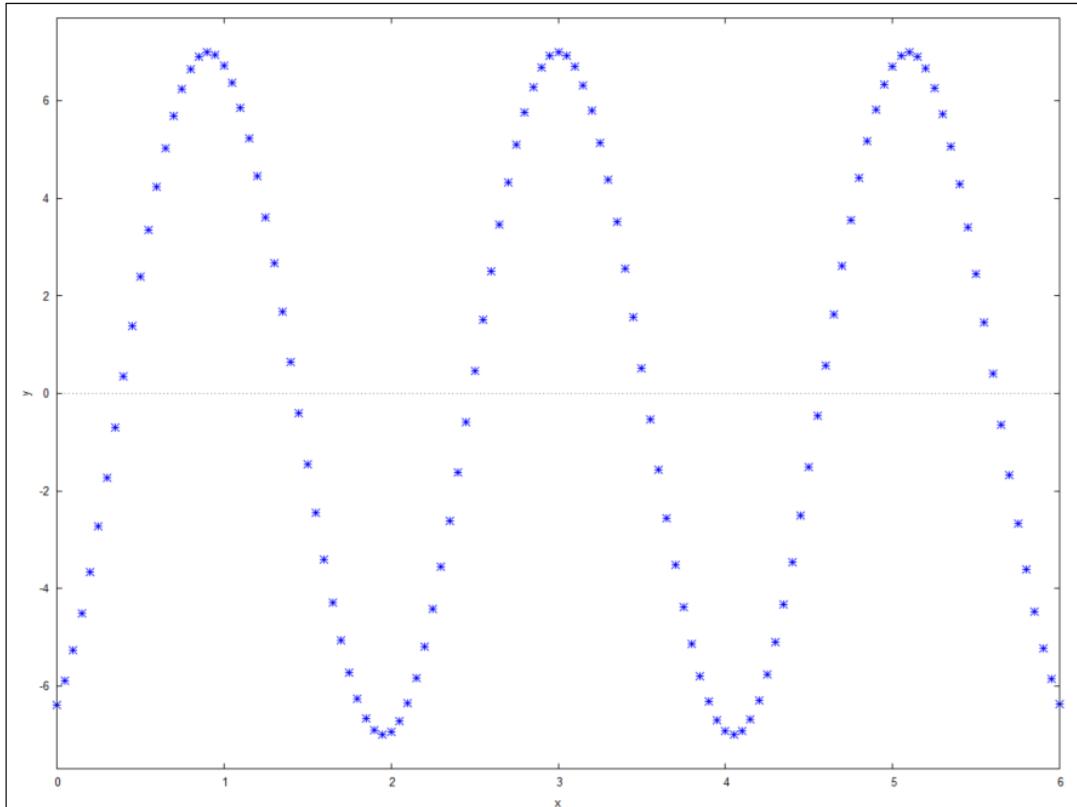
→ A: [\xi_0: 7, k: 3, \delta :20] /* valores arbitrários. */\$ \lambda: 2*%pi/k\$
\xi(x):= A[1]* sin(A[2]*(\lambda + x - A[3]/A[2]))\$ disp('\lambda = \lambda, \xi(x) = \xi(x))\$
/* Conjunto de pontos. */ \xi: makelist([x,\xi(x)],x,0,x_1: 6,0.05)\$
wxplot2d([discrete, \xi], [style, points], [color, blue], [point_type, asterisk])\$

```
(%i29) A: [ $\xi_0: 7$ ,  $k: 3$ ,  $\delta: 20$ ] /* valores arbitrários. */$  $\lambda: 2\pi/k$ $
 $\xi(x):= A[1] \cdot \sin(A[2] \cdot (\lambda + x - A[3]/A[2]))$ $ disp('lambda =  $\lambda$ , 'xi(x) =  $\xi(x)$ )$ 
/* Conjunto de pontos. */  $\xi: makelist([x, \xi(x)], x, 0, x\_1: 6, 0.05)$ $
wxplot2d([discrete,  $\xi$ ], [style, points], [color, blue], [point_type, asterisk])
```

$$\lambda = \frac{2\pi}{3}$$

$$\xi(x) = 7 \sin\left(3\left(x + \frac{2\pi}{3} - \frac{20}{3}\right)\right)$$

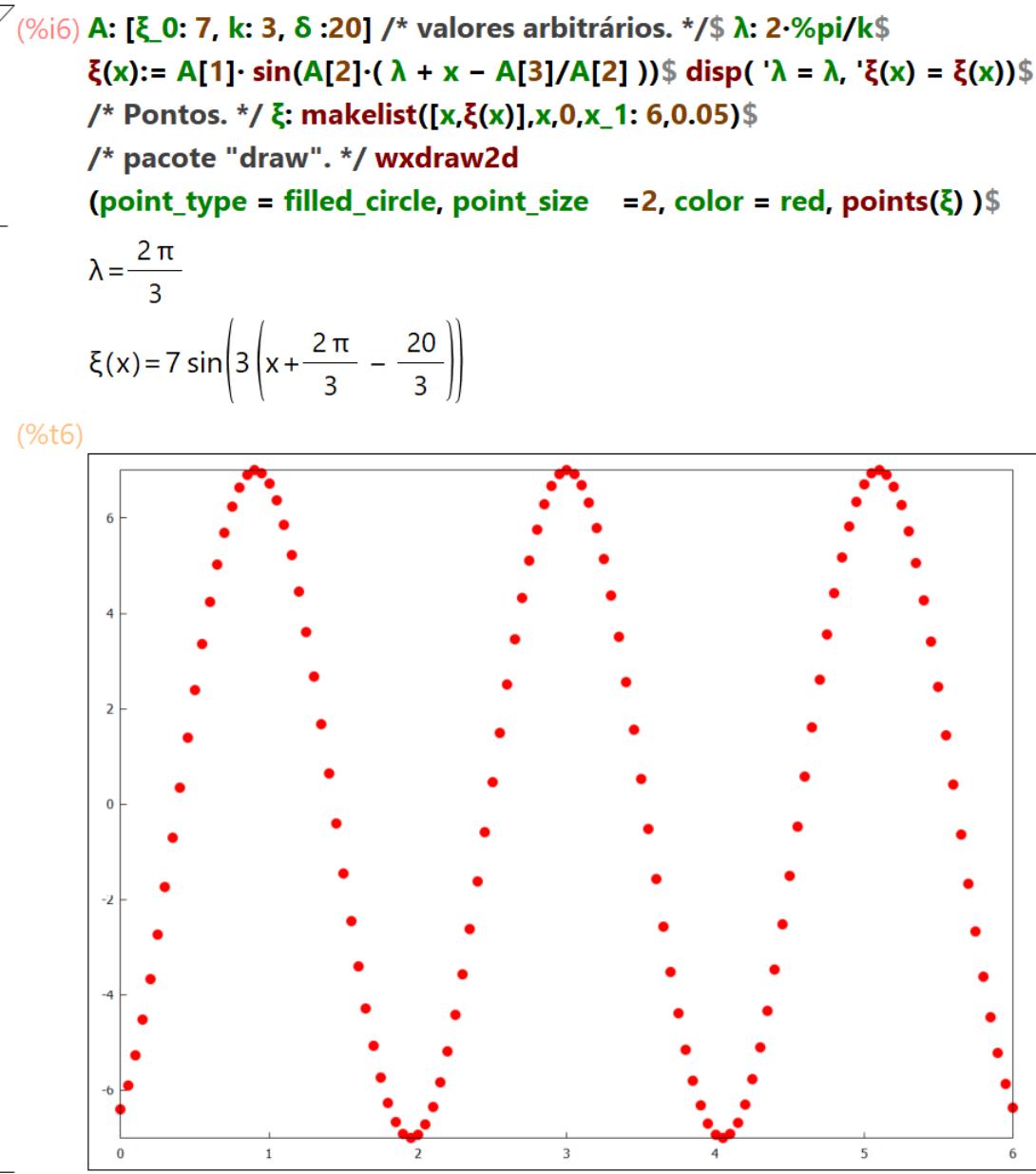
(%t29)



CODE CELL MAXIMA

→

```
A: [ $\xi_0: 7$ ,  $k: 3$ ,  $\delta: 20$ ] /* valores arbitrários. */$  $\lambda: 2\pi/k$ $
 $\xi(x):= A[1] * \sin(A[2] * (\lambda + x - A[3]/A[2]))$ $ disp('lambda =  $\lambda$ , 'xi(x) =  $\xi(x)$ )$ 
/* Pontos. */  $\xi: makelist([x, \xi(x)], x, 0, x\_1: 6, 0.05)$ $
/* pacote "draw". */ wxdraw2d
(point_type = filled_circle, point_size = 2, color = red, points( $\xi$ ))$
```



3.3 Onda Harmônica como um Modelo Físico

Até aqui trabalhamos com desenhos hipotéticos, mas, numa situação real, uma onda o é de fato assim considerada se observadas as características que confirmam sua presença no meio. Uma dessas é o movimento de vai e vem dos elementos que constituem o meio que está sob efeito da suposta onda. Tal movimento é transferido sucessivamente de um ponto para sua vizinhança a cada momento.

Para refletir essa condição há de se fazer o perfil, ora representado por uma senoide, como algo em movimento ou caminhante. Assim, a parcela $\frac{\delta}{k}$ na equação 3.1 passa a ser escrita com a conotação de tempo por meio do produto vt ,

$$\frac{\delta}{k} \equiv vt$$

onde

1. t é a variável independente tempo;
2. v é a velocidade com que o perfil estaria se movimentando, é chamada *velocidade de fase*. No caso da onda harmônica seu perfil não altera, portanto todas as fases da onda caminham à mesma velocidade.

Reescrevendo a equação 3.1, nesses termos,

$$\xi(x, t) = \xi_0 \sin k(x - vt),$$

e, fazendo $k = \frac{2\pi}{\lambda}$, temos

$$\xi(x, t) = \xi_0 \sin \left(kx - \frac{2\pi}{\lambda} vt \right),$$

que reescrevendo fica,

$$\xi(x, t) = \xi_0 \sin \left(kx - \frac{2\pi}{\frac{\lambda}{v}} t \right).$$

O termo que aparece nessa última expressão, na forma λ/v é denominado *período de oscilação*, P , associado a cada ponto x sujeito ao efeito da onda ("os vaivéns"), é expresso em unidades de tempo. Consequentemente define-se a *frequência angular* da onda, ω , em radianos por unidade de tempo

$$\omega = \frac{2\pi}{P}. \quad (3.2)$$

Portanto, para um onda harmônica, movendo-se (propagando-se) no sentido positivo do eixo X escreve-se, suas função de onda na forma:

$$\xi(x, t) = \xi_0 \sin(kx - \omega t). \quad (3.3)$$

Uma vez que ciclo completo de oscilação se associa o valor de 2π rad, podemos definir uma grandeza ν por meio da expressão,

$$\nu = \frac{\omega}{2\pi}, \quad (3.4)$$

que corresponderá à taxa em *ciclos (ou fração de ciclo) por unidade de tempo* com que oscilações (efeito local da onda) estão perpassando no ponto x . A partir da equação 3.3, podemos fazer $\omega = 2\pi\nu$ e comparar com o segundo membro da equação 3.2, do que se deduz $P = \frac{1}{\nu}$. Assim temos,

$$P = \frac{\lambda}{v} = \frac{1}{\nu},$$

de onde se deduz

$$\lambda\nu = v, \quad (3.5)$$

e, também a equação

$$\lambda = vP, \quad (3.6)$$

a qual permite a interpretação de λ como a distância que é percorrida pelo movimento ondulatório ao tempo de um período, P . Uma vez que apliquemos $x = 0$ na equação 3.3, resulta $\xi(0, t) = \xi_0 \sin(-\omega t)$. Se a fonte da oscilação estiver a começar a atuar nessa posição, ou seja, $t = 0$, então teremos $\xi(0, 0) = 0$. O valor de ξ , em $t = 0$ chama-se *condição inicial* e, não necessariamente, terá o valor sempre igual a zero. Por conta disso podemos escrever tal quantidade na forma literal, $\xi(0, 0) = \xi_0 \sin \phi$. Assim a equação 3.3 passa a ser escrita de forma mais geral como:

$$\xi(x, t) = \xi_0 \sin(kx - \omega t + \phi), \quad (3.7)$$

onde ϕ é medida em radianos e chamada *constante de fase*. A presença dessa contante de fase não altera a forma da onda, apenas a desloca sobre o eixo X ou no eixo dos tempos. O valor k mencionado inicialmente possui um significado importante, ele é o *número de onda*. A partir da equação 3.2 podemos fazer $k\lambda = 2\pi$, de onde se tira que k indica quantos comprimentos de onda estão contidos na distância 2π radianos.

Seja uma função senoidal $y = 5 \sin x - 2$, que representa uma grandeza a qual iniciou um movimento ondulatório na direção positiva do eixo X. A velocidade angular foi determinada no valor de dois radiano por segundo. Vamos representar essa condição no wxMaxima.

CODE CELL MAXIMA

```

-> A: [ xi_0: 5, k: 1, omega: 2 , phi: 2 ] /* parâmetros do movimento ondulatório em t=0. */$  

xi(x, t):= A[1]*sin( A[2]*x -A[3]*t +A[4] )$  

print("Onda caminhante: ", "xi(x, t) =", xi(x, t) )$  

(%i3) A: [ xi_0: 5, k: 1, omega: 2 , phi: 2 ] /* parâmetros do movimento ondulatório em t=0. */$  

xi(x, t):= A[1]*sin( A[2]*x -A[3]*t +A[4] )$  

print("Onda caminhante: ", "xi(x, t) =", xi(x, t) )$  

Onda caminhante: xi(x, t) = 5 sin(x - 2 t + 2)

```

Supondo que a fonte de oscilação do exemplo anterior manteve-se continuamente em ação². E em algum momento ocorreu um processo sobre movimento que ‘deslocou’ a onda para frente. De tal forma que no ponto de fase igual a $\frac{3\pi}{2}$ rad a amplitude passou a ser 80% do valor original. Vamos analisar tal condição no wxMaxima.

CODE CELL MAXIMA

```

-> kill(all)$ A: [ ξ_0: 5, k: 1, ω: 2 , φ: 2 ] /* parâmetros do movimento em t=0. */$ 
numer: true$ fpprintprec: 3$ ξ(x, t):= A[1]*sin( A[2]*x -A[3]*t +A[4])$ 
/* Fase inicial.*/ θ: 3*%pi/2$ /* Δδ, deslocamento da fase. */ 
/* Equação de balanço.*/ EQ: A[1]* sin(θ+Δδ) = 0.5 *A[1]* sin(θ)$ 
SOL: rat(solve(EQ, Δδ))$ /* SOL retorna como lista. */ 
Δδ: float(pickapart(part(SOL[1],2),3))$ /* retorna apenas o valor. */ 
print(« "Onda inicial: ", "ξ(x, t) =", ξ(x, t) )$ 
ξ_1(x, t):= A[1]*sin( A[2]*x -A[3]*t +(A[4]+Δδ) )$ 
print(« "Onda final: ", "ξ(x, t) =", ξ_1(x, t) )$ 
/* Plotagem. */ H(x) := if x < (3*%pi/2 - 2) then ξ(x, 0) else ξ_1(x, 0)$ 
wxplot2d( H(x), [x,-10,10] , nobox )$
```

²Neste caso, disse-se que há um *trem de ondas*.

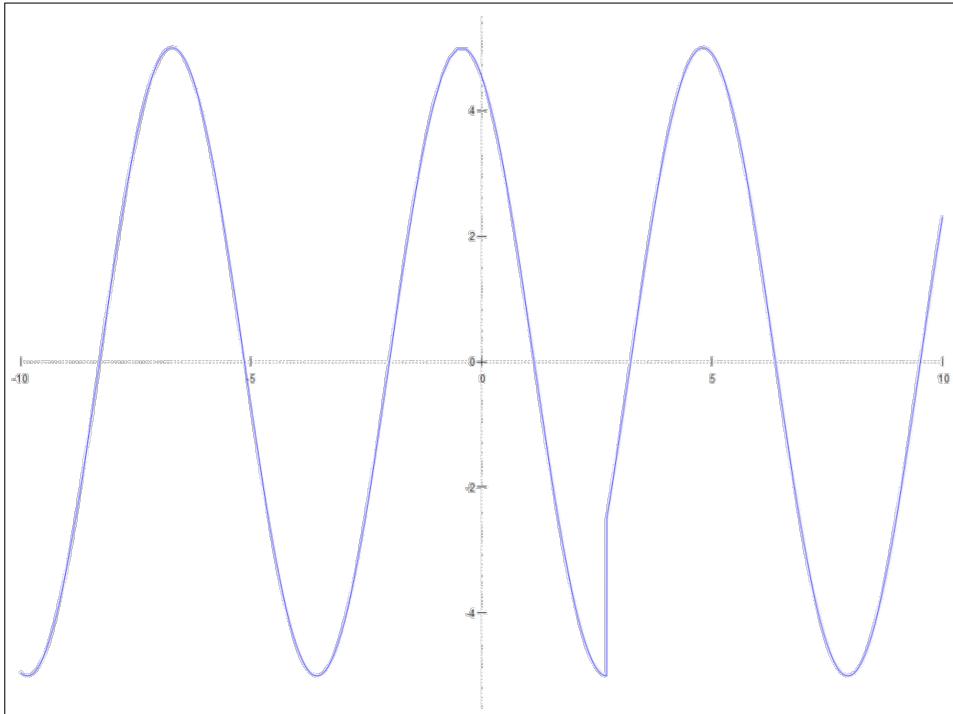
```
(%i13)kill(all)$ A: [ ξ_0: 5, k: 1, ω: 2 , φ: 2 ] /* parâmetros do movimento em t=0. */
numer: true$ fpprintprec: 3$ ξ(x, t):= A[1]·sin( A[2]·x - A[3]·t + A[4])$
/* Fase inicial.*/ θ: 3·%pi/2$ /* Δδ, deslocamento da fase. */
/* Equação de balanço.*/ EQ: A[1]·sin(θ+Δδ) = 0.5 ·A[1]·sin(θ)$
SOL: rat(solve(EQ, Δδ))$ /* SOL retorna como lista. */
Δδ: float(pickapart(part(SOL[1],2),3))$ /* retorna apenas o valor. */
print("► Onda inicial: ", "ξ(x, t) =", ξ(x, t))$
ξ_1(x, t):= A[1]·sin( A[2]·x - A[3]·t +(A[4]+Δδ) )$
print("► Onda final: ", "ξ(x, t) =", ξ_1(x, t))$
/* Plotagem. */ H(x) := if x < (3·%pi/2 - 2) then ξ(x, 0) else ξ_1(x, 0)$
wxplot2d( H(x), [x,-10,10] , nobox )$
```

rat: replaced 2.5 by 5/2 = 2.5
rat: replaced 4.71 by 117506110/24935571 = 4.71
rat: replaced 2.5 by 5/2 = 2.5
rat: replaced 2.5 by 5/2 = 2.5
rat: replaced 4.71 by 112086759/23785549 = 4.71
rat: replaced 4.71 by 106667408/22635527 = 4.71
rat: replaced 4.71 by 106667408/22635527 = 4.71
solve: using arc-trig functions to get a solution.
Some solutions will be lost.

rat: replaced -0.524 by -16066985/30685681 = -0.524
rat: replaced -4.42e-8 by -1/22635527 = -4.42e-8
rat: replaced -0.524 by -16066985/30685681 = -0.524
rat: replaced -4.42e-8 by -1/22635527 = -4.42e-8
rat: replaced -1.44e-15 by -1/694586560788887 = -1.44e-15
rat: replaced 5.24 by 48009887/9169213 = 5.24
rat: replaced -5.24 by -48009887/9169213 = -5.24

- Onda inicial: $\xi(x, t) = 5 \sin(x - 2t + 2)$
- Onda final: $\xi(x, t) = 5 \sin(x - 2t - 3.24)$

(%%t13)



Um determinado movimento ondulatório arbitrário tem como função de onda a expressão dada por

$$\xi(x, t) = \sqrt{2} \sin(2x - t),$$

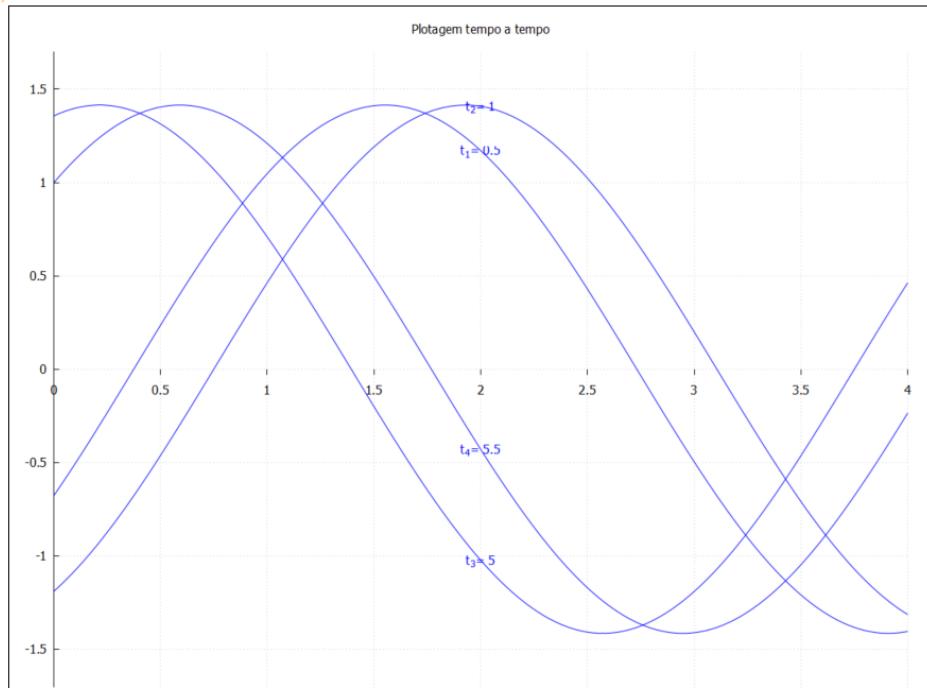
vamos analisar a posição do ponto em $x = 2$ para quatro momentos de tempo diferentes e arbitrários.

CODE CELL MAXIMA

```
→ kill(all)$declare_subscripted(ξ_0)$ ξ_0: sqrt(2)$
t: [t_1: 0.5, t_2: 1, t_3: 5, t_4: 5.5]$ ξ[i](x):= ξ_0*sin( 4*x/3 - t[i] ) $
set_draw_defaults( axis_top=false, axis_bottom=false, axis_right = false, xtics_axis=true,
xrange = [0,4], yrange = [-1.7,1.7], point_type = ~ filled_circle, point_size =~ 0 ,
ip_grid_in=[1,1], ip_grid=[1,1], grid = true, title = "Plotagem tempo a tempo" )$
wxdraw2d(color=blue,
explicit(ξ[1],x,0,4), points( [2,ξ[1](2)] ), label( [concat("t_1= ", t[1]), 2,ξ[1](2) ] ),
explicit(ξ[2],x,0,4), points( [2,ξ[2](2)] ), label( [concat("t_2= ", t[2]), 2,ξ[2](2) ] ),
explicit(ξ[3],x,0,4), points( [2,ξ[3](2)] ), label( [concat("t_3= ", t[3]), 2,ξ[3](2) ] ),
explicit(ξ[4],x,0,4), points( [2,ξ[4](2)] ), label( [concat("t_4= ", t[4]), 2,ξ[4](2) ] ) )$
```

```
(%i6) kill(all)$declare_subscripted(ξ_0)$ ξ_0: sqrt(2)$
t: [t_1: 0.5, t_2: 1, t_3: 5, t_4: 5.5]$ ξ[i](x):= ξ_0*sin( 4*x/3 - t[i] ) $
set_draw_defaults( axis_top=false, axis_bottom=false, axis_right = false, xtics_axis=true,
xrange = [0,4], yrange = [-1.7,1.7], point_type = filled_circle, point_size = 0 ,
ip_grid_in=[1,1], ip_grid=[1,1], grid = true, title = "Plotagem tempo a tempo" )$
wxdraw2d(color=blue,
explicit(ξ[1],x,0,4), points( [2,ξ[1](2)] ), label( [concat("t_1= ", t[1]), 2,ξ[1](2) ] ),
explicit(ξ[2],x,0,4), points( [2,ξ[2](2)] ), label( [concat("t_2= ", t[2]), 2,ξ[2](2) ] ),
explicit(ξ[3],x,0,4), points( [2,ξ[3](2)] ), label( [concat("t_3= ", t[3]), 2,ξ[3](2) ] ),
explicit(ξ[4],x,0,4), points( [2,ξ[4](2)] ), label( [concat("t_4= ", t[4]), 2,ξ[4](2) ] ) )$
```

(%t6)



Duas ondas propagam-se em uma mesma direção ao longo de uma dimensão retilínea de um meio perfeitamente elástico e acontece interferência. Supondo elas estando com um mesmo comprimento de onda e mesma velocidade de propagação. A amplitude de cada onda é de 9,7mm e há uma diferença de fase de 110°. Vamos analisar essa interferência no Maxima.

CODE CELL MAXIMA

```
→ load("trigtools" ) $ fpprintprec : 3$  
A:[ k:1, ω: 1, φ_1: 0, φ_2: 110 , ξ_1: 9.7 , /* para k e ω aqui os valores são hipotéticos. */  
ξ_2: 9.7 ]$  
A: append( A, [A[3]*%pi/180, A[4]*%pi/180] )$  
/* Ondas superpostas. */ ξ(x,t):= [ A[5]* sin( A[1]*x - A[2]* t -A[7] ),  
A[6]* sin( A[1]*x - A[2]* t - A[8]),  
/*Onda resultante.*/ A[5]* sin( A[1]*x - A[2]* t -A[7] ) +A[6]* sin( A[1]*x - A[2]* t - A[8]) ]$  
/* Amplitude. */ ξ_3: float( sqrt( A[5]^2 + A[6]^2 + 2* A[5]*A[6]*cos(A[8]-A[7]) ));  
numer:false$  
/* Solução no ponto de máximo */ sol: float ( trigsolve ( float( ξ(x,0)[3])=ξ_3 , 2,3 ) );  
x_max: first(sol);  
wxdraw2d(grid = true, xtics_axis = true, xaxis = true, axis_top=false, axis_right=false,  
axis_bottom = false, explicit( ξ(x,0)[1], x,0,4*%pi), explicit( ξ(x,0)[2], x,0,4*%pi),  
line_width=5, color=red, line_type = solid, explicit( ξ(x,0)[3], x,0,4*%pi),  
point_type = filled_circle, point_size = 2, points_joined = false, color=blue,  
points( [ [ x_max, 0], [x_max, ξ_3] ] ), font_size = 20, font = "Arial",  
label( ["amplitude ξ_3 ", x_max+1, ξ_3 ] ), label( ["x_max ", x_max+.5, 0 ] )  
)$
```

```
(%i10)load("trigtools") $ fpprintprec:3$  

A:[ k:1, ω: 1, φ_1: 0, φ_2: 110 , ξ_1: 9.7 , /* para k e ω aqui os valores são hipotéticos. */  

ξ_2: 9.7 ]$  

A:append( A, [A[3]·%pi/180, A[4]·%pi/180] )$  

/* Ondas superpostas. */ ξ(x,t):= [ A[5]·sin( A[1]·x - A[2]·t -A[7] ),  

A[6]·sin( A[1]·x - A[2]·t - A[8] ),  

/*Onda resultante.*/ A[5]·sin( A[1]·x - A[2]·t -A[7] ) +A[6]·sin( A[1]·x - A[2]·t - A[8] ) ]$  

/* Amplitude. */ ξ_3: float(sqrt( A[5]^2 + A[6]^2 + 2·A[5]·A[6]·cos(A[8]-A[7]) ))$  

numer:false$  

/* Solução no ponto de máximo */ sol: float(trgsolve(float(ξ(x,0)[3])=ξ_3, 2,3 ))$  

x_max:first(sol);  

wxdraw2d(grid = true, xtics_axis = true, xaxis = true, axis_top=false, axis_right=false,  

axis_bottom = false, explicit(ξ(x,0)[1], x,0,4·%pi), explicit(ξ(x,0)[2], x,0,4·%pi),  

line_width=5, color=red, line_type = solid, explicit(ξ(x,0)[3], x,0,4·%pi),  

point_type = filled_circle, point_size = 2, points_joined = false, color=blue,  

points( [[x_max, 0], [x_max, ξ_3]] ), font_size = 20, font = "Arial",  

label( ["amplitude ξ_3 ", x_max+1, ξ_3] ), label( ["x_(max)", x_max+.5, 0] )  

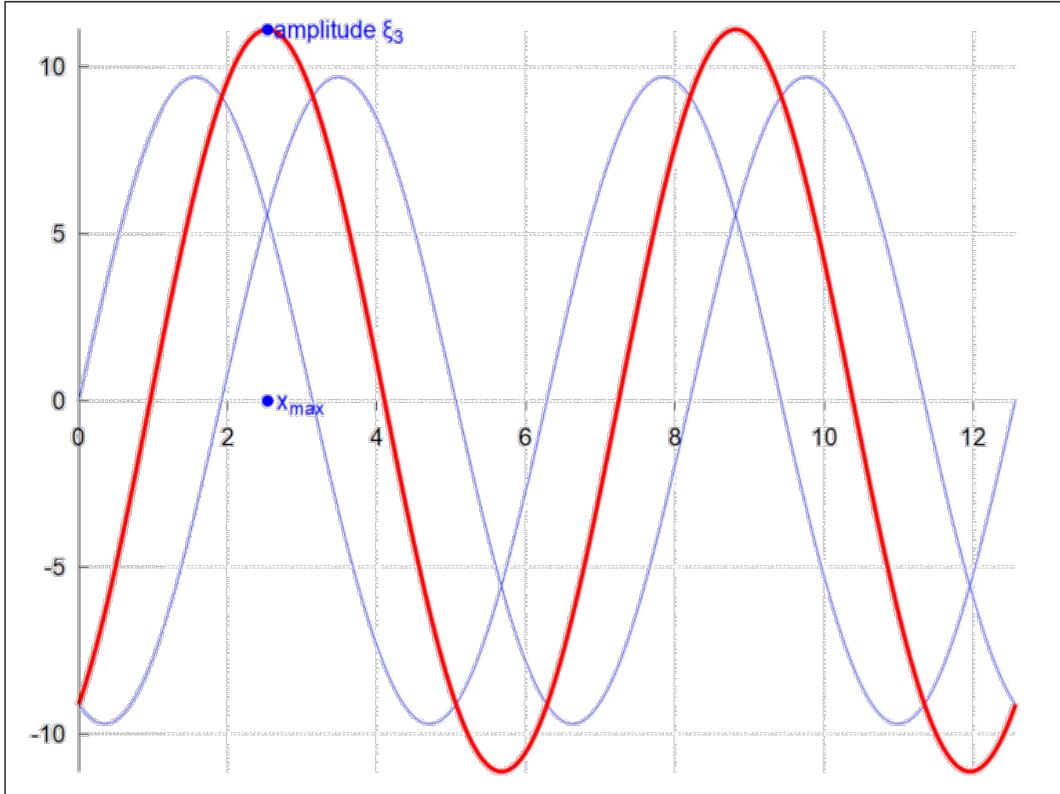
)$
```

ξ_3 11.1

sol {2.53,2.53}

x_max2.53

(%i10)



Ondas estacionárias podem ser produzidas em uma corda usando um gerador de pulsos mecânicos com determinada frequência. Vamos supor um tal mecanismo onde os ajustes sejam dados pela tração fornecida por uma massa dependurada em uma extremidade e na outra um motor fornece o movimento oscilatório.

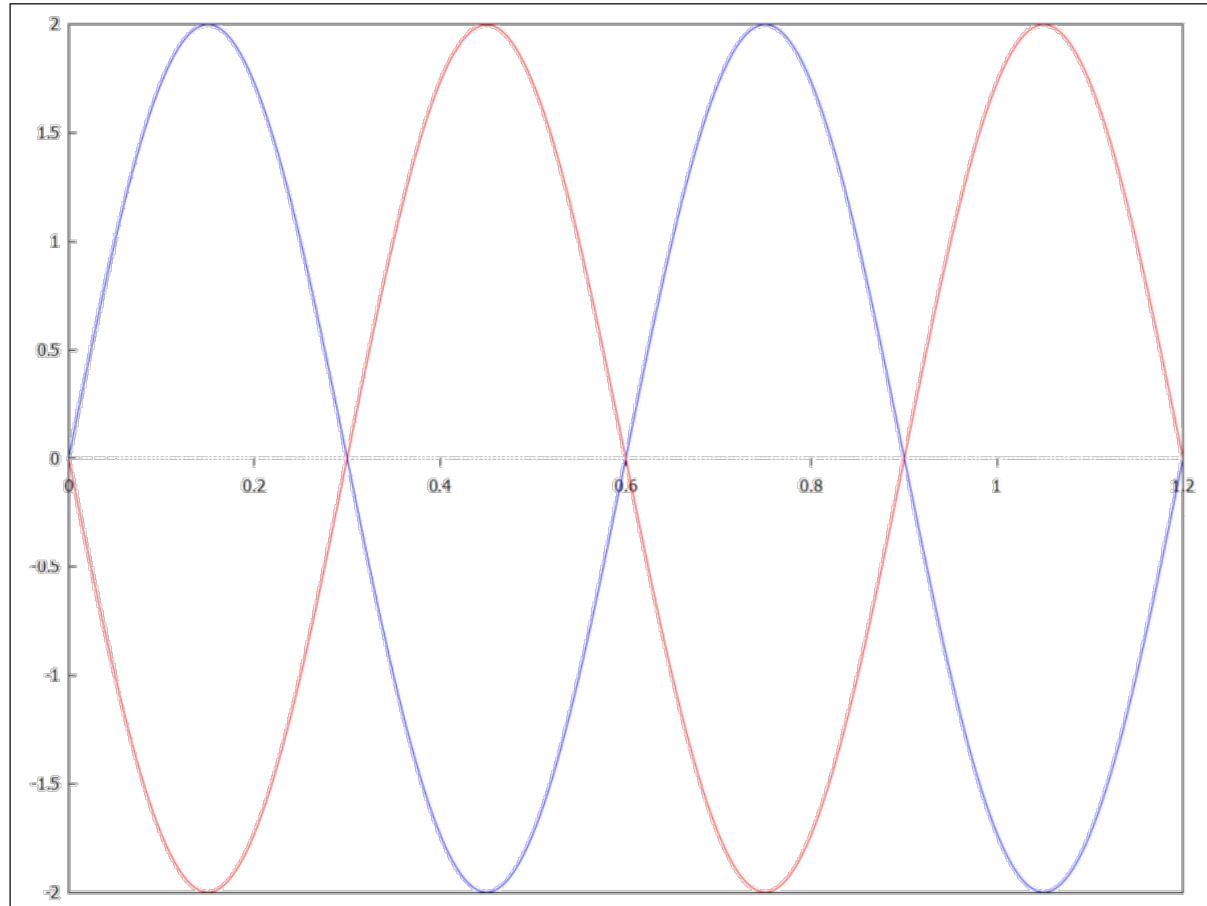
CODE CELL MAXIMA

→ A: /*Frequência do motor*/ ν : 120, /*Comprimento da corda.*/ L: 1.2,
/*massa específica.*/ μ : 1.6, /*padrão de movimento.*/ n: 4, ξ_0 : 1 /*(arbitrário).*/]
\$fpprintprec:4\$ /*velocidade angular*/ ω : float(2*%pi*A[1])\$
/*comprimento de onda*/ λ_n : 2* A[2]/A[4]\$ /*número de onda*/ k: float(2*%pi/ λ_n)\$
/*frequência estacionária.*/ ν_n : A[1]\$ /*velocidade de onda*/ v: $\lambda_n * \nu_n$ \$
/*Amplitude da onda incidente.*/ \$ display([ν , L, μ , ω , λ_n , k, v, ν_n , ξ_0])\$
/*amplitudes.*/A[5]: 2*A[5]\$ $\xi_m(x)$:= A[5]*sin(k*x)\$ display($\xi_m(x)$)\$
/*Equação da onda estacionária.*/ $\xi(x,t)$:= $\xi_m(x)$ *cos($\omega*t$)\$ simp: false\$ display($\xi(x,t)$)\$
simp: true\$ /*Força de tração*/ F: float(v^ 2* μ *10^-3)\$ display(F)\$
wxdraw2d(xtics_axis=true, xaxis =true,
explicit($\xi_m(x)$, x,0,L), color = red, explicit(- $\xi_m(x)$, x,0,L))\$

```
(%i18) A: /*Frequência do motor*/ v: 120, /*Comprimento da corda.*/ L: 1.2,
/*massa específica.*/ μ: 1.6, /*padrão de movimento.*/ n: 4, ξ_0: 1 /*(arbitrário)*/
$fpprintprec:4$ /*velocidade angular*/ω: float(2·%pi·A[1])$
/*comprimento de onda*/ λ_n: 2· A[2]/A[4] /*número de onda*/ k: float(2·%pi/λ_n)$
/*frequência estacionária.*/v_n: A[1]$ /*velocidade de onda*/ v: λ_n · v_n$
/*Amplitude da onda incidente.*/ $ display([v, L, μ, ω, λ_n, k, v, v_n, ξ_0])$
/*amplitudes.*/A[5]: 2·A[5]$ ξ_m(x):= A[5]·sin(k·x )$ display(ξ_m(x ))$
/*Equação da onda estacionária.*/ ξ(x,t):=ξ_m(x)·cos(ω·t)$ simp: false$ display(ξ(x,t))$
simp: true$ /*Força de tração*/ F: float(v^2·μ·10^-3)$ display(F)$
wxdraw2d( xtics_axis=true, xaxis =true,
           explicit( ξ_m(x), x,0,L ), color = red, explicit(- ξ_m(x), x,0,L ) )$
```

[v,L,μ,ω,λ_n,k,v,v_n,ξ_0]=[120,1.2,1.6,754.0,0.6,10.47,72.0,120,1]
 $\xi_m(x)=2 \sin(10.47 x)$
 $\xi(x,t)=2 \sin(10.47 x) \cos(754.0 t)$
 $F=8.294$

(%t18)



O campo elétrico e o campo magnético são grandezas acopladas formando o chamado de *campo eletromagnético*. Isso pode ser evidenciado mediante a equação vetorial $\vec{B} = \mu_0 \epsilon_0 \vec{v} \times \vec{E}$ associada a

uma carga puntual em movimento (não acelerada).

Ao aplicar as equações de Maxwell a campos elétricos e magnéticos cruzados, ou seja, o campo eletromagnético, chega-se a uma expressão na forma diferencial:

$$\frac{\partial^2 E}{\partial t^2} = \frac{1}{\mu_0 \epsilon_0} \frac{\partial^2 E}{\partial x^2},$$

e tal equação indica que o campo eletromagnético propaga-se, abstraindo-se das particularidades, como uma onda plana no espaço, isso porque tal expressão é compatível com a equação diferencial da onda (o mesmo valerá para o vetor \vec{B}).

Podemos também aqui fazer uso da função harmônica como solução da equação diferencial dessa propagação do campo eletromagnético. Assim, temos duas expressões de funções harmônicas aplicáveis:

$$E_y = E_0 \sin(kx - \omega t), \quad (3.8)$$

$$B_z = B_0 \sin(kx - \omega t), \quad (3.9)$$

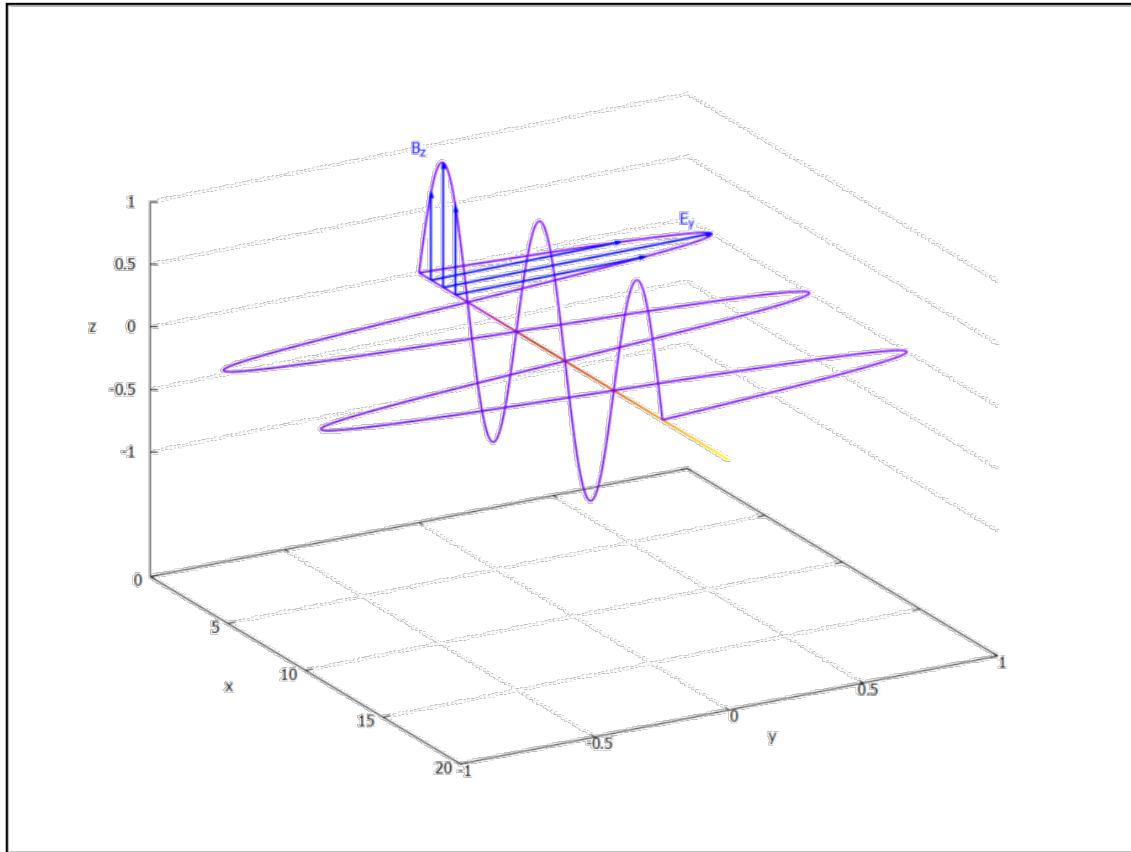
as quais podem ser representadas graficamente.

CODE CELL MAXIMA

```
→ A:[ k:1, c:1 , E_0:1 , B_0: E_0/c ]$ ω: k*c$ t:0 $
wxdraw3d( xlabel="x", ylabel="y", zlabel="z", grid=true,
yrange = [-1,1], zrange = [-1,1], xrange = [0,20], nticks = 200,
view = [60,60], colorbox= false, line_width = 2,
enhanced3d = [x, x, y,z], parametric(u,0,0,u,-1,11*pi),
enhanced3d = [y, x, y,z], parametric(u, E_0* sin(k*u-ω*t) ,0,u,0,5*pi),
enhanced3d = [z, x, y,z], parametric(u,0, B_0* sin(k*u-ω*t) ,u,0,5*pi),
head_length = 0.1, head_angle = 15,
label(["E_y",0, E_0, 0]),
vector( [%pi/4, 0, 0], [0, E_0* sin(k*pi/4-ω*t), 0 ] ),
vector( [%pi/2, 0, 0], [0, E_0, 0 ] ),
vector( [3*pi/4, 0, 0], [0, sin(k*3*pi/4-ω*t), 0 ] ),
label(["B_z",0, 0,E_0]),
vector( [%pi/4, 0, 0], [0, 0,B_0* sin(k*pi/4-ω*t) ] ),
vector( [%pi/2, 0, 0], [0, 0,E_0 ] ),
vector( [3*pi/4, 0, 0], [0, sin(k*3*pi/4-ω*t) ] )
) $
```

```
(%i4) A:[ k:1, c:1 , E_0:1 , B_0:E_0/c ]$ ω: k·c$ t:0 $
wxdraw3d( xlabel="x",ylabel="y", zlabel="z", grid=true,
yrange = [-1,1], zrange = [-1,1], xrange = [0,20], nticks = 200,
view = [60,60], colorbox= false, line_width = 2,
enhanced3d = [x, x, y,z], parametric(u,0,0,u,-1,11·%pi),
enhanced3d = [y, x, y,z], parametric(u, E_0·sin(k·u-ω·t),0,u,0,5·%pi),
enhanced3d = [z, x, y,z], parametric(u,0, B_0·sin(k·u-ω·t),u,0,5·%pi),
head_length = 0.1, head_angle = 15,
label(["E_y",0, E_0, 0]),
vector( [%pi/4, 0, 0], [0, E_0·sin(k·%pi/4-ω·t), 0] ),
vector( [%pi/2, 0, 0], [0, E_0, 0] ),
vector( [3·%pi/4, 0, 0], [0, sin(k·3·%pi/4-ω·t), 0] ),
label(["B_z",0, 0,E_0]),
vector( [%pi/4, 0, 0], [0, 0,B_0·sin(k·%pi/4-ω·t) ] ),
vector( [%pi/2, 0, 0], [0, 0,E_0 ] ),
vector( [3·%pi/4, 0, 0], [0, 0,sin(k·3·%pi/4-ω·t) ] )
)$
```

(%t4)



Vamos considerar agora uma superfície, tal como uma membrana retangular com suas bordas fixadas, uma vez que essa superfície seja perturbada, são produzidas ondas que se propagam em todas as direções

e refletidas nas bordas, acarretando uma interferência. A equação que governa essa situação na condição estacionária, correspondendo a um sistema de quatro ondas será:

$$\xi = -4 \xi_0 \sin k_1 x \sin k_2 y \sin \omega t,$$

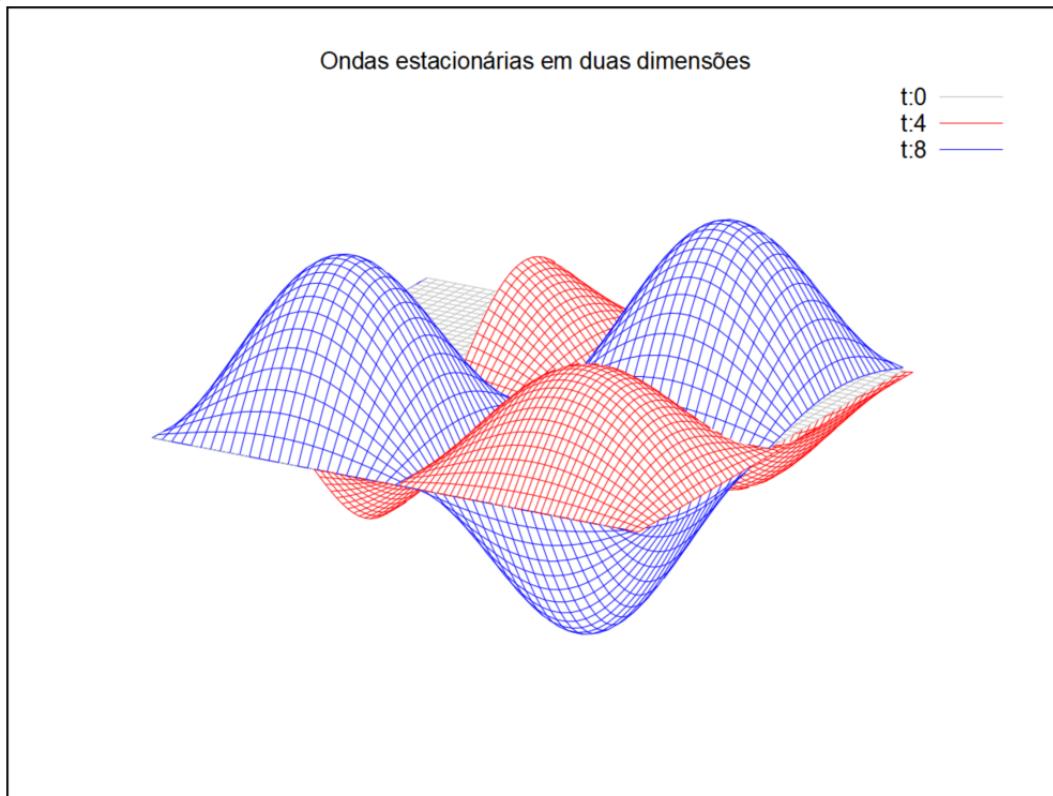
onde $k_1 = \frac{n_1 \pi}{a}$, $k_2 = \frac{n_2 \pi}{b}$, n_1, n_2 , são números inteiros, a e b correspondem às dimensões da membrana. Supondo-se ondas produzidas com uma única frequência esta será dada por $\omega = \sqrt{k_1^2 + k_2^2} c$.

CODE CELL MAXIMA

```
-> A: [a: 2, b: 3, n_1: 2, n_2: 2, c:1, ξ_0: 10 ]$ k_1: A[3]*%pi/A[1]$ k_2: A[4]*%pi/A[2]$  
ω: sqrt(k_1^2 + k_2^2)*A[5]$ t:[0,4,8]$ ξ(x,y,t):=-4*A[3]* sin(k_1*x)*sin(k_2*y)*sin(ω*t)$  
wxdraw3d(font = "Arial", font_size = 20, title ="Ondas estacionárias em duas dimensões",  
key = concat("t:", t[1]), color = gray, explicit(ξ(x,y,t[1])),x,0,a,y,0,b),  
key = concat("t:", t[2]), color = red, explicit(ξ(x,y,t[2])),x,a/3,a,y,0,b),  
key = concat("t:", t[3]), color = blue, explicit(ξ(x,y,t[3])),x,0,a,y,0,b), axis_3d = false,  
contour = none, surface_hide = true, ip_grid=[1,1])$
```

```
(%i7) A: [a: 2, b: 3, n_1: 2, n_2: 2, c:1, ξ_0: 10 ]$ k_1: A[3]*%pi/A[1]$ k_2: A[4]*%pi/A[2]$  
ω: sqrt(k_1^2 + k_2^2)*A[5]$ t:[0,4,8]$ ξ(x,y,t):=-4*A[3]* sin(k_1*x)*sin(k_2*y)*sin(ω*t)$  
wxdraw3d(font = "Arial", font_size = 20, title ="Ondas estacionárias em duas dimensões",  
key = concat("t:", t[1]), color = gray, explicit(ξ(x,y,t[1])),x,0,a,y,0,b),  
key = concat("t:", t[2]), color = red, explicit(ξ(x,y,t[2])),x,a/3,a,y,0,b),  
key = concat("t:", t[3]), color = blue, explicit(ξ(x,y,t[3])),x,0,a,y,0,b), axis_3d = false,  
contour = none, surface_hide = true, ip_grid=[1,1])$
```

(%t7)



APLICAÇÕES ONDULATÓRIAS

De modo mais geral, o estudo do movimento ondulatório é feito do ponto de vista da dinâmica do processo, ou seja, é assumindo que haja uma perturbação em um ponto e ela é propagada até outro ponto, o que conduz tal processo a ser descrito por uma equação dinâmica, ou diferencial.

4.1 Corda Estendida

Aqui consideramos uma corda como um meio sobre o qual se propagará um movimento oscilatório periódico. Por hipótese será uma corda ideal então ela será inextensível e perfeitamente homogênea, com densidade linear de valor m em unidades de massa por unidade de comprimento.

No momento inicial a corda estará tensionada e esticada na horizontal por uma força mecânica T .

Considerando que a corda seja constituída de pequenos segmentos interligados. Tais elementos entrarão em desequilíbrio sob efeito da perturbação impostas a eles, verticalmente e consequentemente propagarão um movimento ondulatório transversal, gerando um perfil induzido pela fonte de perturbação. A velocidade dessa propagação é dada pela expressão:

$$v = \sqrt{\frac{T}{m}}. \quad (4.1)$$

Em nossa simulação consideraremos em tese o princípio da composição do movimento, ou seja, a posição de um ponto da corda será correspondente ao efeito composto pela soma das posições decorrentes dos perfis estático e dinâmico (senoidal).

Para representar o perfil estático utilizaremos a curva catenária, a qual é descrita algebraicamente pela função hiperbólica na forma

$$f(x) = a \cosh\left(\frac{x}{a}\right) \quad (4.2)$$

onde a é a ordenada do ponto crítico ou de equilíbrio dado por

$$a = \frac{H}{mg},$$

e H representa o esforço horizontal no ponto crítico. Então a posição de um determinado ponto (x, y) da corda será dada pela expressão

$$y = a \cosh\left(\frac{x}{a}\right) [2 \sin(kx) \cos(\omega t)],$$

ressaltando os seguintes aspectos:

- o parâmetro a nem sempre corresponderá a um significado de medida de altura em uma situação real.
- a parte trigonométrica já está expressa tendo sido levado em consideração o efeito ondulatório sobre o fio preso nas suas duas extremidades, que induziria-o a um estado ondulatório estacionário, assumindo-se o modelo harmônico a governar o processo.

EXEMPLO 4.1.1: Seja uma corda 3 unidades de comprimento e 0,5 de peso unitário (linear), estendida e fixada pelas suas extremidades a uma altura de 2 unidades de comprimento, tal forma que apresenta um esforço horizontal no ponto de equilíbrio de 11 unidades e que, após ser submetido a um movimento ondulatório senoidal transversal com velocidade angular de 1rad/s apresentou deslocamento máximo de 10%. Desenhe por meio do wxMaxima os perfis de onda para quatro instantes de tempo.

Solução.

Esse caso envolverá duas etapas, pois temos que determinar primeiramente os parâmetros da curva catenária. E em seguida juntar esta ao modelo indicado. É conveniente estruturar os parâmetros da catenária para que esta esteja posicionada sobre o primeiro quadrante de um plano cartesiano. Isso pode ser feito no contexto das linhas de programação.

Metodologia.

```

-> kill(all)$ /* limpar a memória*/

/* Comprimento L do fio: */ L: 5$
/* Tensão horizontal no ponto crítico (ponto de equilíbrio)*/ H:11$
/* Peso unitário */ w:0.5$
/* Ordenada do ponto de sustentação */ y_0:2$

/* Calcular a curva catenária genérica */
/* Diretriz da catenária */ a: H/w$
/* Determinar a abscissa do seu ponto de sustentação */
numer: true$ 
sol1: solve(a* sinh(x_1/a)=L/2, x_1)$
x_1: last(last(sol1))$ 
cat1(x):=a* cosh((x)/a) $

ajuste: cat1(x_1) - y_0$ 
/* ajuste da ordenada para o ponto de sustentação*/
cat(x):= a* cosh((x-x_1)/a) - ajuste$ 
/* transformação da catenária para o primeiro quadrante */

/* Gerar gráficos e plotar */
wxplot_size:[1900,400]$
wxdraw2d(
title="Fio Estendido",
xlabel="x",ylabel="y",grid=true,
color=blue,line_type=dots,key="Catenária",
line_width = 8,
grid=[5,5],
nticks=300,
line_type = solid,
explicit( cat(x) ,x, 0, 2*x_1))$ 

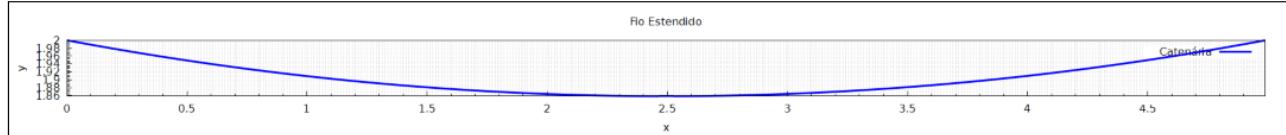
/* Representar o efeito ondulatório sobre o perfil do fio */
/* tempo */ t: 1$
/* velocidade angular */ omega: 1$
/* número de onda */ k: 2%pi/(2*x_1)$
for i: 1 while i <= 4 do (
/* estrutura da repetição */
wxdraw2d(
title= concat("Instante=",float(t)) ,
xlabel="x",ylabel="y",grid=true,
color=blue,line_type=dots,key="Perfil de onda",
line_width = 8,
grid=[5,5],
nticks=300,
line_type = dots,
explicit( cat(x) + 0.1*cat(x) * 2* (sin (k*x)) *cos (omega*t) ,x, 0, 2*x_1)),
t: t+1 )$ 
/*Fim*/

```

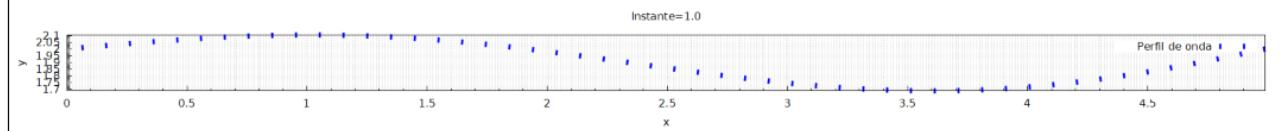
Figura 4.1: Simulação dos perfis correspondentes a quatro tempos diferentes. A primeira imagem corresponde ao perfil estático da corda

```
rat: replaced -2.4946505180016283 by -204985463/82170012 = -2.4946505180016283
```

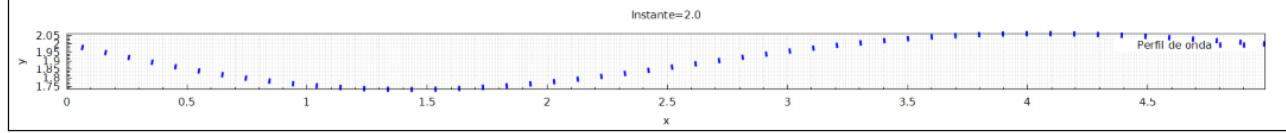
(%t13)



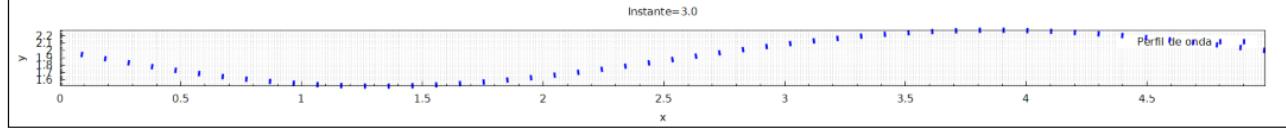
(%t17)



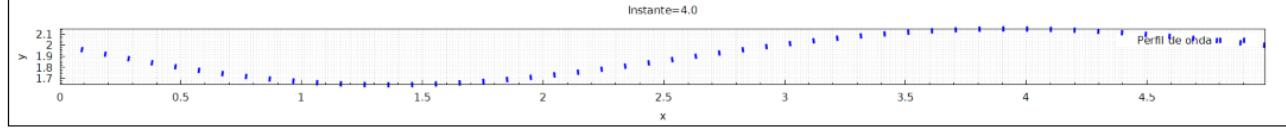
(%t18)



(%t19)



(%t20)



4.2 Efeito Doppler

Antes propriamente de falar sobre efeito Doppler há de se destacar o significado do termo *efeito* no contexto da física; é um sinônimo de fenômeno físico. Muitas descobertas da física surgiram como o resultado de efeitos observados casualmente. Por exemplo: a famosa "maçã de Newton".

No caso do efeito Doppler não foi diferente. Consta que quem primeiro observou tal efeito foi C. J. Doppler (1802–1853), um físico alemão, cujo nome foi atribuído ao nome do fenômeno. A abordagem do efeito Doppler se dá dentro do estudo do movimento ondulatório.

Tanto perturbações mecânicas como os campos eletromagnéticos se enquadram nesse conceito, e, portanto, são passíveis de efeito Doppler.

O efeito Doppler consiste na variação da frequência do movimento ondulatório que é percebida pelo observador em movimento relativo. Nesse sentido há certos fatores importantes a considerar, quais sejam:

- a fonte de perturbação ou de ondas.
- o movimento relativo da fonte de ondas.
- os observadores e seus sistemas de referência.
- o movimento relativo dos observadores.

- o tipo de onda, condicionada ou não ao meio material.

No caso de ondas mecânicas o interesse é obter a relação entre a frequência ν produzida pela fonte de ondas e outra, ν' , registrada pelo observador.

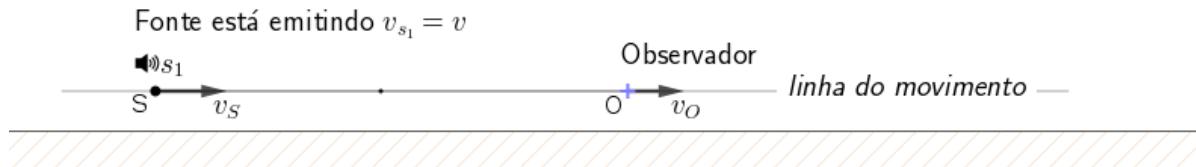
Para obter tal relação nos valeremos da relação inerente da frequência ν :

$$\nu = \frac{\text{quantidade de sinais emitidos}}{\text{tempo de amostragem}},$$

e também do esquema apresentado na figura abaixo.

Figura 4.2: Condições hipotéticas para equacionar o efeito Doppler.

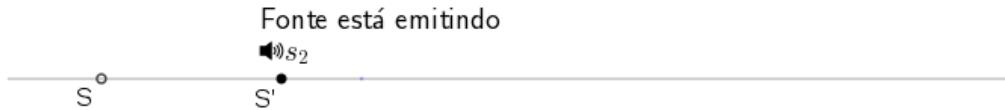
Situação inicial em movimento. Tempo $t = 0$



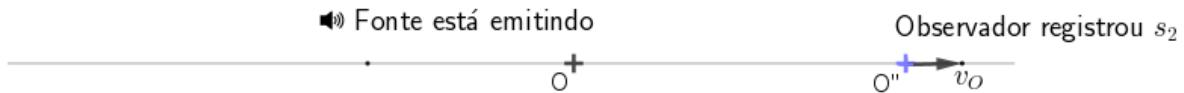
1º registro



Situação em progresso:



2º registro



Os elementos presentes na figura 4.2 preparam-nos escrever equações horárias do movimento, assim temos

$$v t_{OO'} = \overline{SO} + v_O t_{OO'},$$

que resolvendo para o tempo de observação da primeira onda, $t_{OO'}$, temos

$$t_{OO'} = \frac{\overline{SO}}{v - v_O}$$

e também

$$v (t_{OO''} - t_{SS'}) = \overline{SO} - v_S t_{SS'} + v_O t_{OO''},$$

que resolvendo para o tempo de observação da segunda onda, $t_{OO''}$, temos

$$t_{OO''} = \frac{\overline{SO} + (v - v_S) t_{SS'}}{v - v_O}.$$

Então o nosso tempo de amostragem τ será

$$\begin{aligned}\tau &= t_{OO''} - t_{OO'} \\ &= \frac{v - v_S}{v - v_O} t_{SS'}.\end{aligned}$$

E, a frequência desconhecida será dada por

$$\begin{aligned}\nu' &= \frac{\nu t_{SS'}}{\frac{v - v_S}{v - v_O} t_{SS'}} \\ &= \nu \left(\frac{v - v_O}{v - v_S} \right).\end{aligned}$$

Quando uma onda mecânica é ultrapassada pela sua própria fonte ocorre um efeito superveniente ao Doppler, é a *onda de choque* que fica segregada em um cone de propagação cujo eixo é a linha de movimento da fonte.

É possível provar que, de modo geral, o efeito Doppler mecânico pode ser descrito para uma condição de *convergência* fonte-observador conforme a expressão:

$$\nu' = \nu \left(\frac{v + v_O}{v - v_S} \right),$$

e para o caso de *afastamento recíproco* vale a relação:

$$\nu' = \nu \left(\frac{v - v_O}{v + v_S} \right).$$

EXEMPLO 4.2.1: O sonar Doppler envia um pulso sonoro (ou ‘ping’ de saída) com o objetivo de analisar um determinado alvo, e obter informações como a velocidade e sentido de movimento. O sonar irá receber um pulso refletido (‘ping’ refletido) pelo alvo que estará em movimento unidirecional. Simule, por meio de um código de programa, um sonar Doppler estacionário ($v_S = 0$).

Solução.

Grandezas conhecidas serão: a velocidade do som, v , a frequência do pulso emitido, ν , a do pulso refletido, ν' , bem como o tempo de retardo entre ‘pings’, t_p .

A condição de observador estará posicionada no alvo em movimento e assim o objetivo é determinar sua velocidade v_O . A origem do sistema de referência estará posicionada do radar estacionário.

Chamando-se $b > 0$ a abscissa do ponto de reflexão do pulso e de τ o valor do tempo de retardo então teremos:

$$b = v \left(\frac{\tau}{2} \right).$$

Observe que v' resulta de efeito Doppler sobre a frequência ν'_O emitida pelo alvo (fonte secundária). Assim temos:

$$\begin{aligned}\nu'_O &= \nu \left(\frac{v + v_O}{v - 0} \right) \\ &= \nu \left(\frac{v + v_O}{v} \right),\end{aligned}$$

e

$$\begin{aligned}\nu' &= \nu'_O \left(\frac{v + 0}{v - v_O} \right) \\ &= \nu'_O \left(\frac{v}{v - v_O} \right).\end{aligned}$$

Portanto,

$$\begin{aligned}\nu' &= \nu \left(\frac{v + v_O}{v} \right) \left(\frac{v}{v - v_O} \right) \\ &= \nu \left(\frac{v + v_O}{v - v_O} \right).\end{aligned}$$

ou

$$v_O = \nu \frac{\nu' - \nu}{\nu' + \nu},$$

sendo que o resultado calculado é que determinará o sinal algébrico de v_O .

Metodologia.

```

→ /* Simulação do radar Doppler estacionário - movimento unidirecional */

kill(all)$
fpprintprec : 4 $

/* Dados de entrada */
promptComum : "ou digite um valor."$
input : "A velocidade do pulso de saída enviado"$
input : concat(input, " pelo sonar está em" )$
v : 340 $ read(input, v , "m/s", promptComum)$

input : "A frequência do pulso de saída " $
input : concat(input, " enviado pelo sonar está em" )$
ν : 20$ read(input, ν , "kHz", promptComum)$

input : "A frequência Doppler do pulso refletido " $
input : concat(input, " pelo alvo está em" )$
wxdeclare_subscripted(ν_D)$
ν_D: 20.612$ read(input, ν_D, "kHz", promptComum)$

input : "O valor do tempo de retardo " $
input : concat(input,"está em" )$
τ : 0.230$ read(input, τ, "s", promptComum)$

/* Velocidade do observador */ wxdeclare_subscripted(v_O)$
v_O : (ν_D - ν)/(ν_D + ν)$
v_O : % * v$
if ν_D > ν then v_O : v_O*-1 $
/* imposição de um sinal algébrico */
/* para corresponder à situação de movimento relativo */

/* Abscissa do ponto de reflexão do pulso */ b: v * τ/2 $

/* Indicação de movimento do alvo */ Ind:$
block(
if ν_D > ν and v_O<0 then newline(),
Ind: "alvo aproximando-se pela direita.", go(loop),
if ν_D > ν and v_O>0 then newline(),
Ind: "alvo aproximando-se pela esquerda.", go(loop),
if ν_D < ν and v_O>0 then newline(),
Ind: "alvo afastando-se para a direita.", go(loop),
if ν_D < ν and v_O<0 then newline(),
Ind: "alvo afastando-se para a esquerda.", loop
)$

```

```

/* Resultados da simulação */
objetivo : "Sonar Doppler estacionário"
caract : "A origem do sistema de referência "
caract : concat(caract , "corresponde ao sonar estacionário.")$

disp("RESULTADO DA SIMULAÇÃO")$
disp(objetivo)$
disp(caract)$
disp(concat(- Distância até o alvo: ", b,"m." ) )$
disp(concat(- Velocidade do alvo: ", v_O,"m/s."))$
disp(concat(- Indicação: ",Ind)  )$
```

Figura 4.3: Resultado para a simulação do sonar Doppler estacionário.

```

/* Resultados da simulação */
objetivo : "Sonar Doppler estacionário"
caract : "A origem do sistema de referência estará "
caract : concat(caract , "na posição do sonar estacionário.")$

disp("RESULTADO DA SIMULAÇÃO")$
disp(objetivo)$
disp(caract)$
disp(concat("- Distância até o alvo: ", b,"m." ) )$
disp(concat("- Velocidade do alvo: ", v_O,"m/s."))$
disp(concat("- Indicação: ",Ind)  )$
```

A velocidade do pulso de saída enviado pelo sonar está em 340 m/s ou digite um valor. 340;

A frequência do pulso de saída enviado pelo sonar está em 20 kHz ou digite um valor. 20;

A frequência Doppler do pulso refletido pelo alvo está em 20.61 kHz ou digite um valor. 20.62;

O valor do tempo de retardo está em 0.23 s ou digite um valor. 0.23;

RESULTADO DA SIMULAÇÃO

Sonar Doppler estacionário

A origem do sistema de referência estará na posição do sonar estacionário.

- Distância até o alvo: 39.1m.
- Velocidade do alvo: -5.124m/s.
- Indicação: alvo aproximando-se pela direita.

4.3 Radiação de um Dipolo Elétrico Oscilante

Um dipolo elétrico é um dispositivo que representa o efeito físico de duas cargas elétricas simétricas e separadas a uma pequena distância. Polaridade é a propriedade que determina a intensidade de um dipolo e é representada pelo vetor momento de dipolo. Uma forma de gerar ondas eletromagnéticas é fazendo oscilar o valor da polarização. O valor do campo elétrico irradiado é determinado pelas equações

de Maxwell. Uma possível solução leva à expressão:

$$\xi = \frac{\Pi_0 \sin \theta}{4\pi \epsilon_0 r} \left(\frac{\omega}{c}\right)^2 \sin(kr - \omega t),$$

onde Π_0 corresponde ao valor de pico do momento de dipolo elétrico e θ é a coordenada angular em relação ao eixo do dipolo.

EXEMPLO 4.3.1: Para manter um dipolo elétrico irradiando devemos fornecer-lhe energia. Tal energia é transportada pela onda irradiada que ter-lhe-á associada a grandeza *intensidade de onda* à medida em que é estabelecida uma distribuição espacial de sua potência. Represente graficamente a intensidade da onda irradiada em coordenadas polares. Considere a intensidade a partir da relação $I(\theta) = cE_{\text{med}}$, onde corresponde à densidade média de energia irradiada por uma onda.

Solução.

A densidade total de energia de uma onda eletromagnética é dada por;

$$E = c\xi^2.$$

Substituindo pela expressão sugerida para ξ

$$E = c \left[\frac{\Pi_0 \sin \theta}{4\pi \epsilon_0 r} \left(\frac{\omega}{c}\right)^2 \sin(kr - \omega t) \right]^2,$$

e aplicando o teorema do valor médio então obtém-se

$$E_{\text{med}} = \frac{\Pi_0^2 \omega^4}{32\pi^2 c^4 \epsilon_0 r^2} \sin^2 \theta,$$

para obter a expressão

$$I(\theta) = \frac{\Pi_0^2 \omega^4}{32\pi^2 c^3 \epsilon_0 r^2} \sin^2 \theta.$$

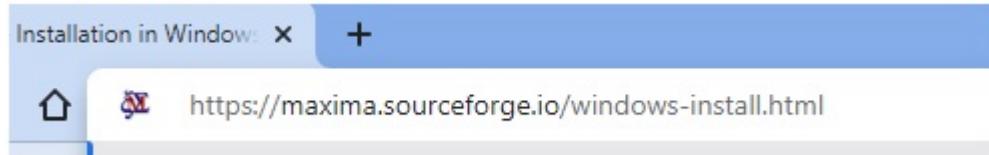
APÊNDICE A

DICAS SOBRE O MAXIMA

A.1 Como Instalar o wxMaxima®

Instalar um software é uma tarefa que a maioria de usuários de PC já fizeram alguma vez, e instalar o Maxima é uma tarefa que não haverá muita dificuldade. O primeiro passo é obter o arquivo instalador. Uma opção é fazer o processo de *download* do endereço da internet ou URL (sugestão do autor): maxima.sourceforge.io/windows-install.html (A.1). A página na figura A.2 contém um passo a passo,

Figura A.1: Entrando com a URL, para buscar a página do Maxima no *Google*.



que é recomendável que seja lido. Entretanto, por ora seguiremos mais focados no processo operacional de instalação em si. Assim, localize o link, nessa página com o texto: 5.46.0-Windows, como mostrado nessa imagem.

Figura A.2: Página encontrada do Maxima no navegador da Internet.



Como forma de gerenciar o processo de instalação são apresentadas a seguir as janelas que interagem nas diversas etapas dessa instalação.

Figura A.3: Janela que apresenta o início do processo de instalação.



Figura A.4: Janela que apresenta os termos de uso.



Figura A.5: Janela que apresenta o diretório local para instalação de arquivos.

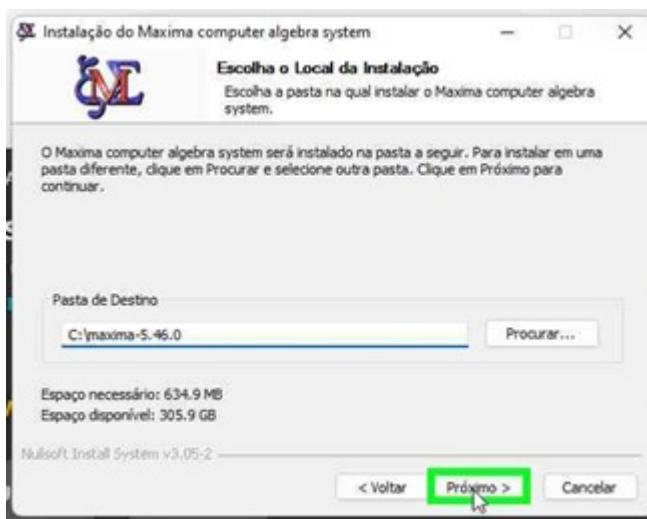


Figura A.6: Janela de configuração de atalhos.



Figura A.7: Janela de configuração do escopo de funções do programa.

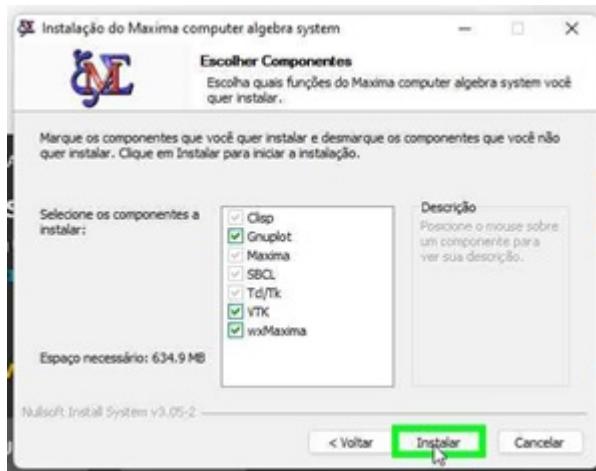


Figura A.8: Janela de status do processo de instalação.

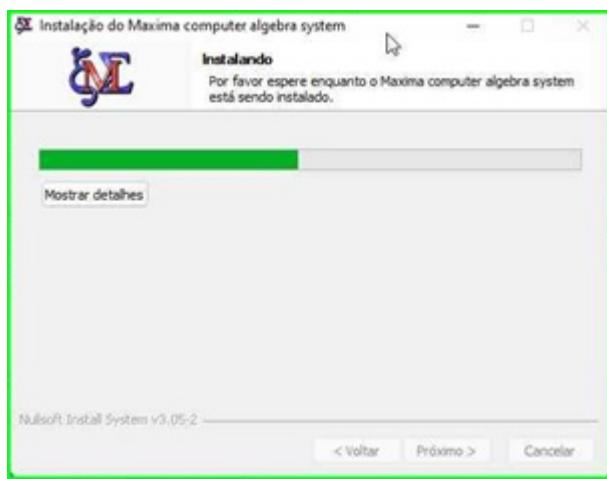


Figura A.9: Janela de confirmação da instalação.

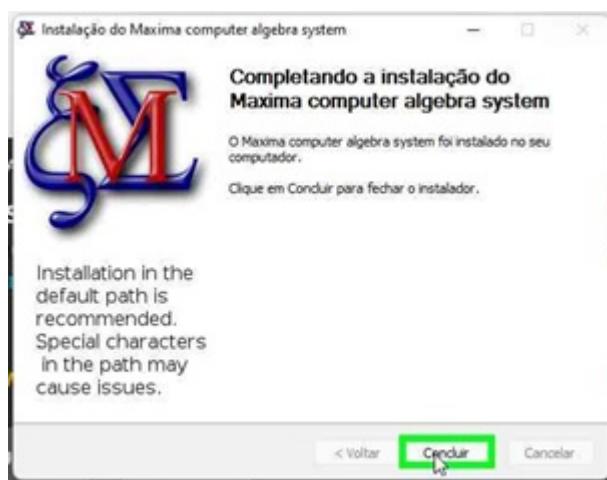
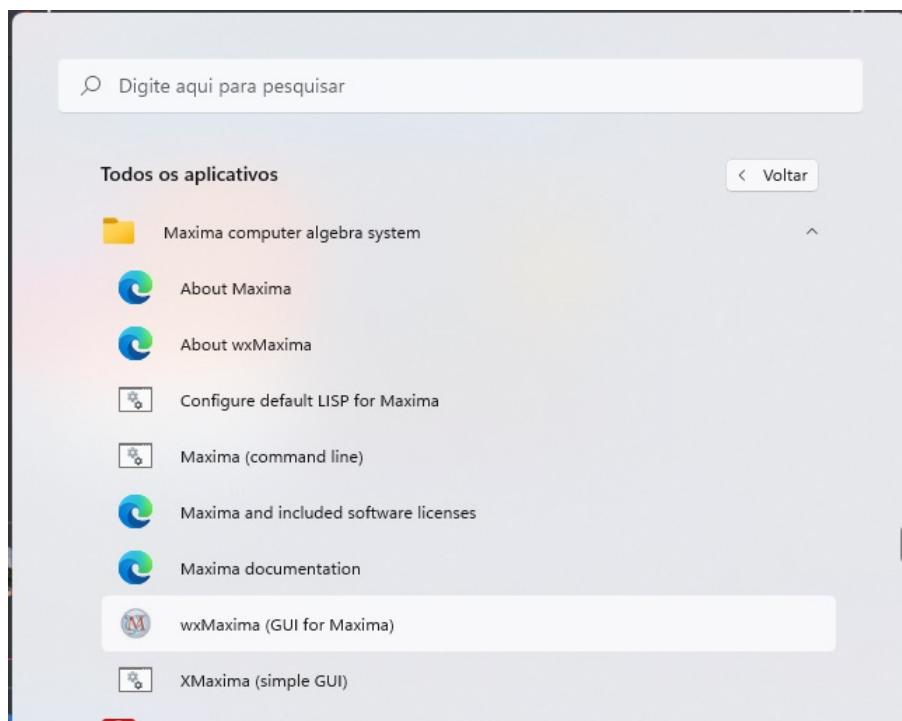


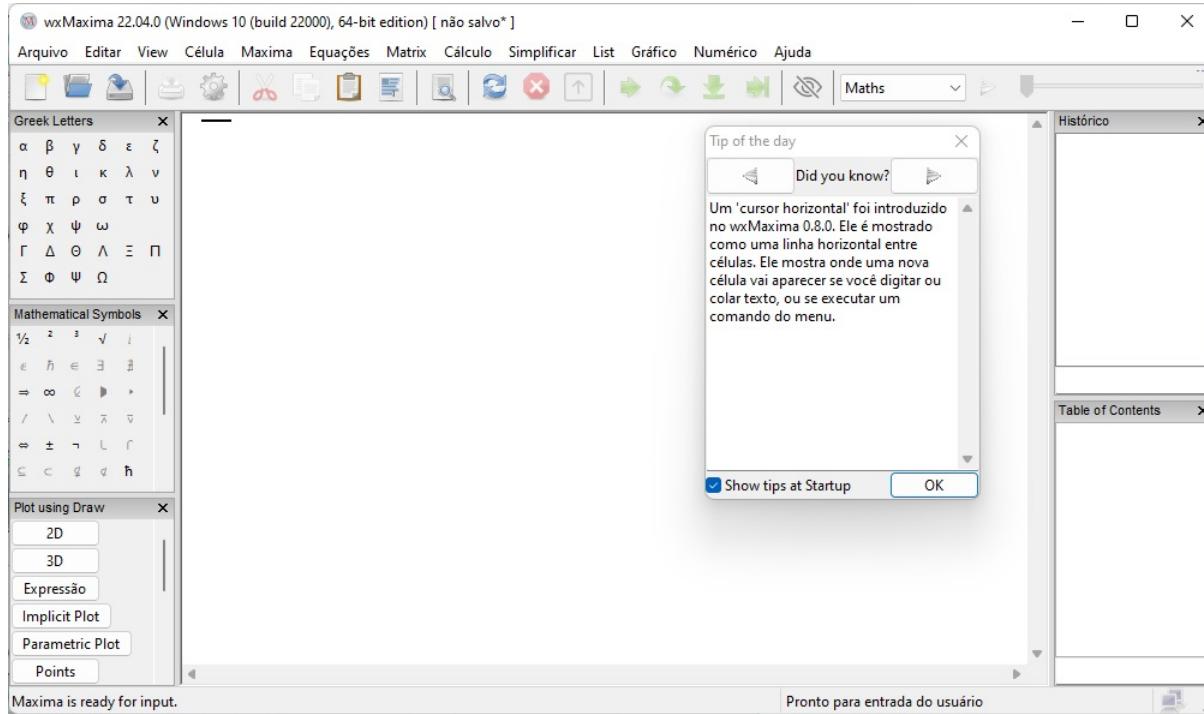
Figura A.10: Aspecto do comando de menu na janela inicial do Windows.



A.2 Conhecendo o *Interface Gráfico* do wxMaxima

A janela do aplicativo wxMaxima é idêntica a um editor de texto, e permite criar um arquivo, abrir um existente e salvar suas alterações, tudo a partir de um menu suspenso de comandos. Ao abrir o aplicativo, este já fornece de imediato, em sua janela, um prompt para a entrada de comandos, ou textos descritivos, figura A.11.

Figura A.11: *Interface* do wxMaxima.

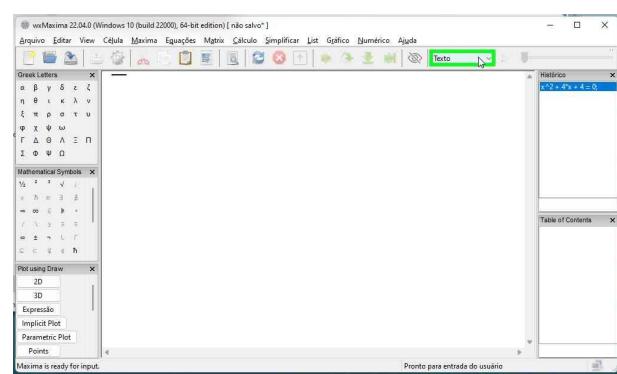


O wxMaxima tem a funcionalidade lógica de uma planilha de cálculo, ou seja, as janelas possuem uma estrutura baseada em células de cálculo.

A figura A.12 mostra o recurso para escolher, a partir de uma lista suspensa, a opção de tipo de entrada no prompt, ou seja,

- tipo Texto
- tipo Math (expressão algébrica)
- tipo Título
- tipo Seção
- tipo Subseção
- tipo Subsubseção
- head 5
- head 6

Figura A.12: Pré-seleção de tipo de entrada a ser inserida no prompt.



A figura A.13 mostra como escolher as opções de tipo de entrada no prompt para o caso de texto. Já a figura A.14 mostra como escolher as opções de tipo de entrada no prompt para o caso de expressão algébrica.

Figura A.13: Pré-seleção para inserir texto no prompt.

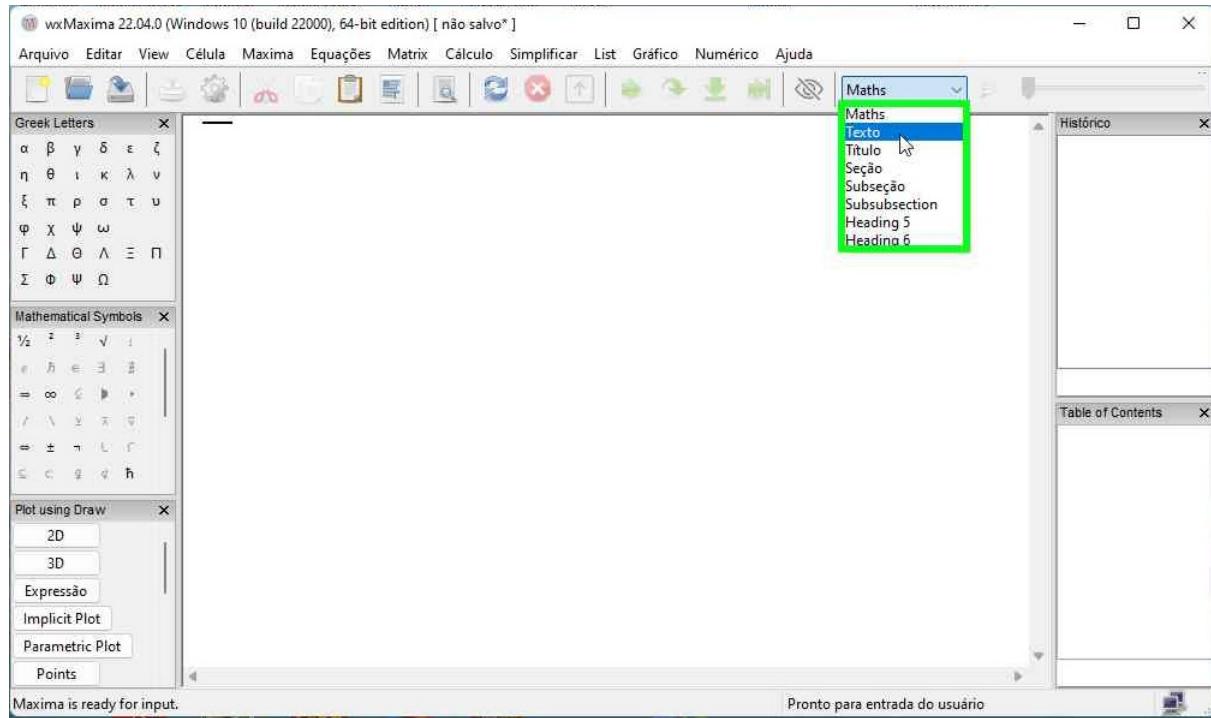


Figura A.14: Pré-seleção para inserir expressão algébrica no prompt.

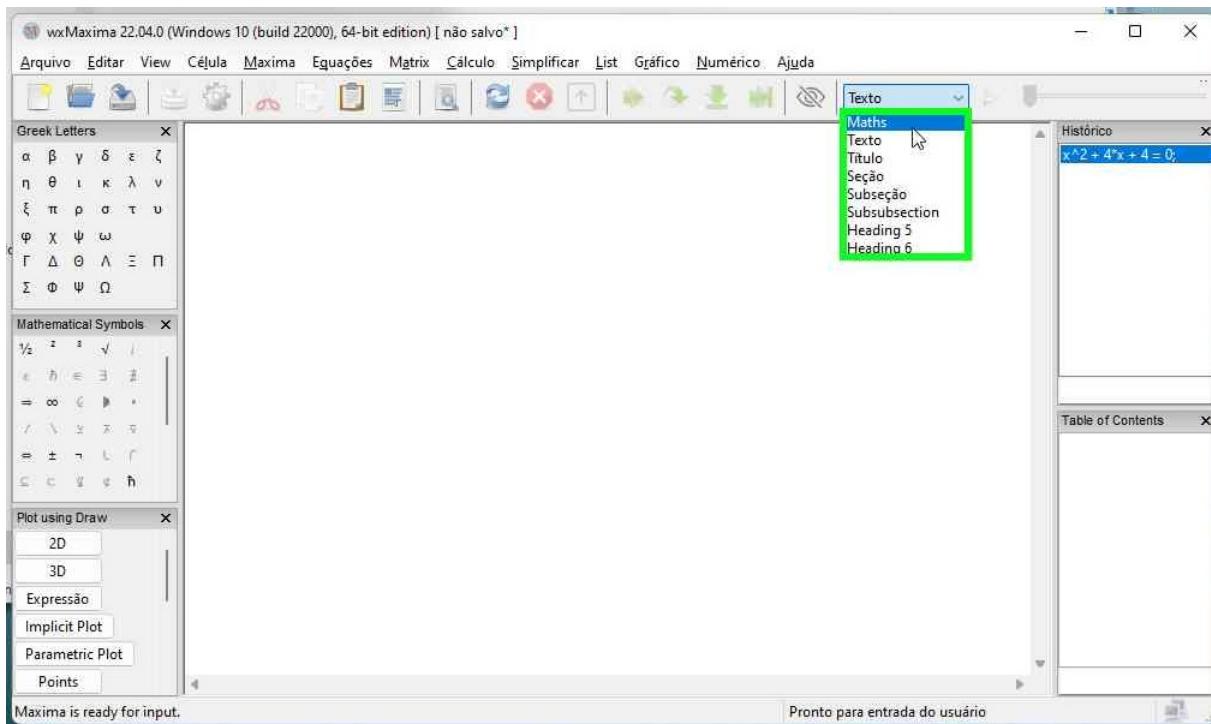
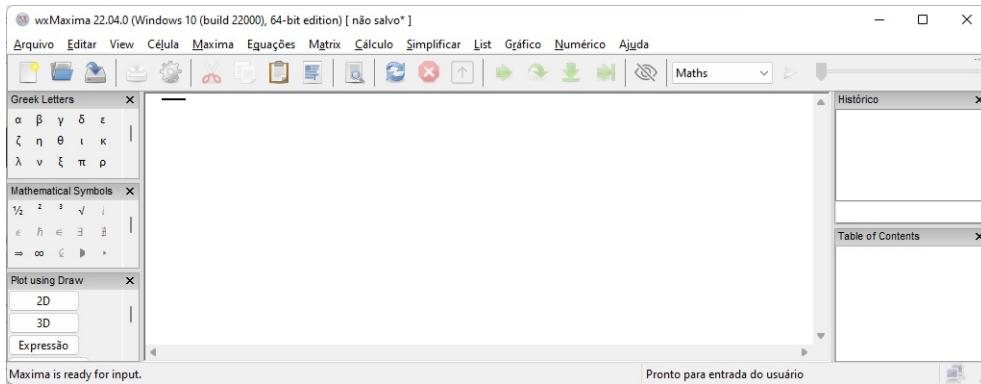


Figura A.15: Cursor intermitente indica a posição para inserir uma célula.



Um cursor, na forma de um traço horizontal intermitente, marca a posição inicial para criar uma célula, figura A.15. Uma vez iniciada digitação, o cursor torna-se vertical, e um símbolo marcador na forma de um parêntese aparece à esquerda no início da linha iniciada, e se a expressão for do tipo *Math* será acompanhado de uma seta, figura A.16. Após a execução da expressão, caso haja um resultado o marcador de célula se redesenha na forma de uma chave, figura A.17.

Figura A.16: Marcador de célula.

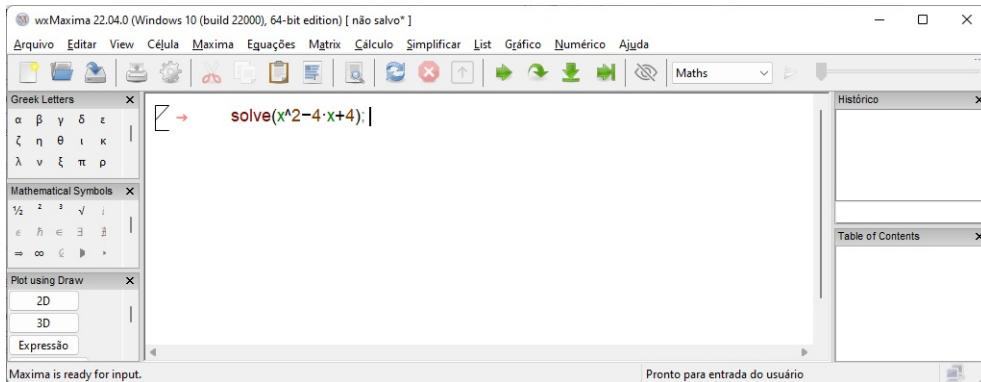
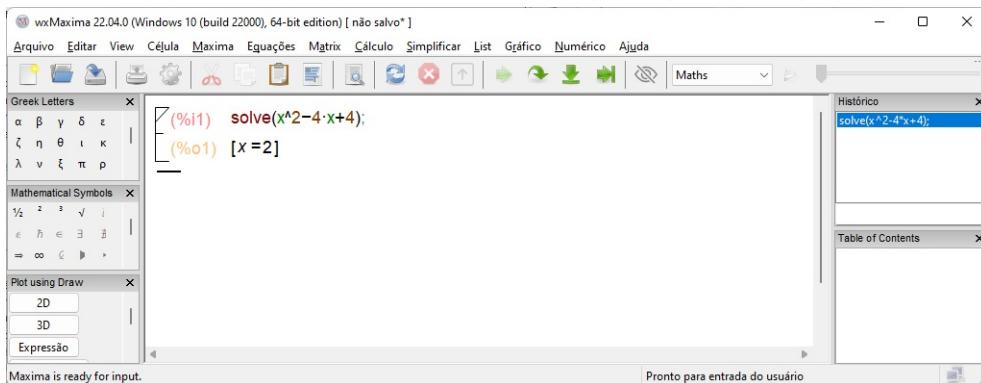


Figura A.17: Marcador estendido ao resultado do Maxima.



Até agora nos ativemos a apresentar generalidades que subsidiarão o leitor durante sua jornada no aprendizado ou mesmo na utilização das técnicas de simulação aqui apresentadas no Maxima. Assunto que será objeto dos próximos capítulos.

A.3 Idioma do *Interface*

Para alterar o idioma da interface acesse o comando "Configurações", e altere para o idioma conveniente para uso do interface. Os passos a seguir alteram para o idioma português.

Figura A.18: Passo inicial ao processo de configurar o idioma

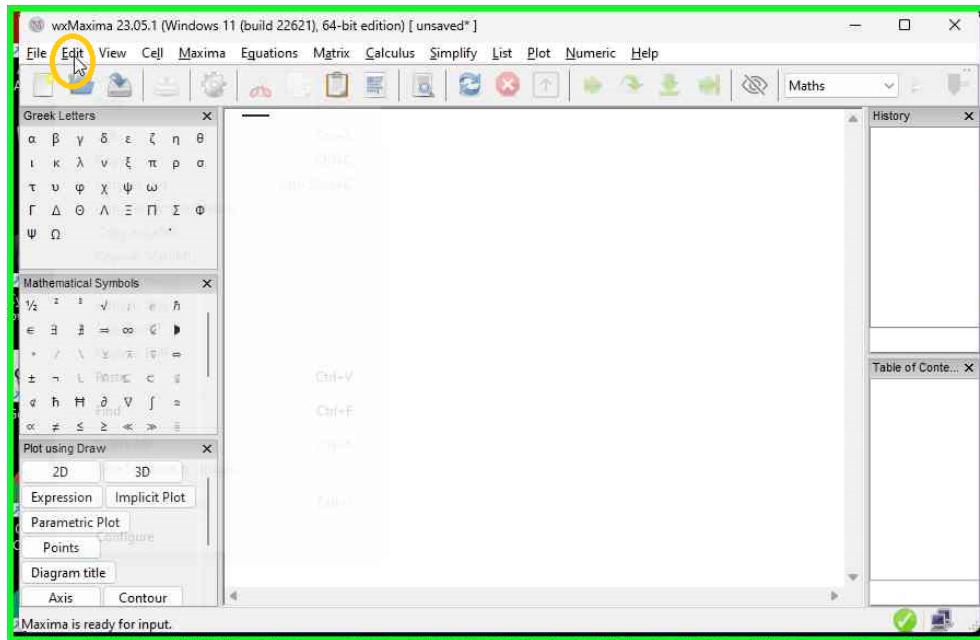


Figura A.19: O comando *Configurar* irá guiar para uma janela de diálogo

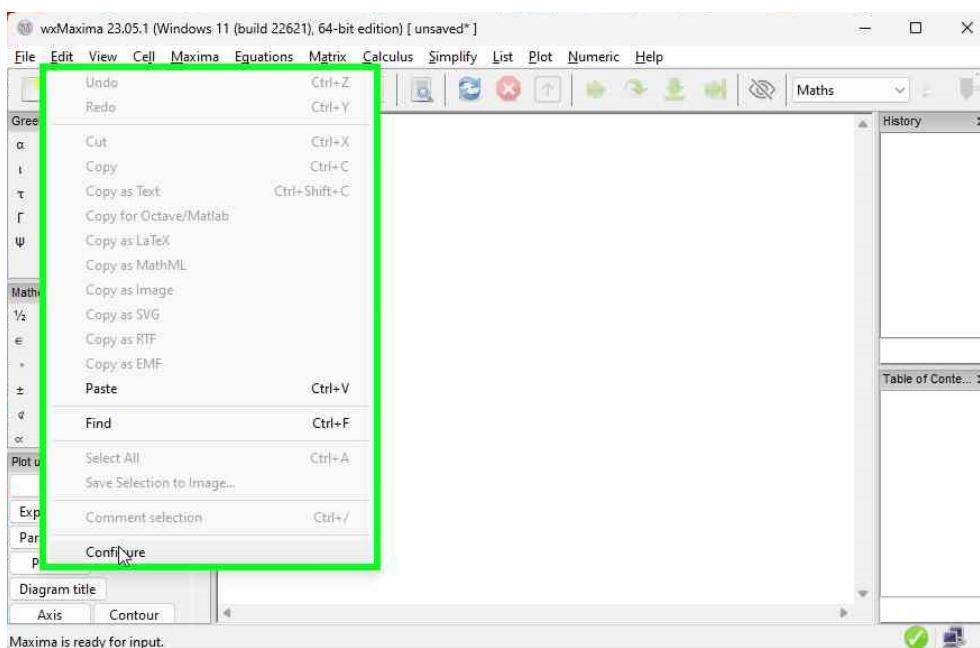


Figura A.20: Na janela de diálogo, acessar o campo indicado para escolha na lista de opções

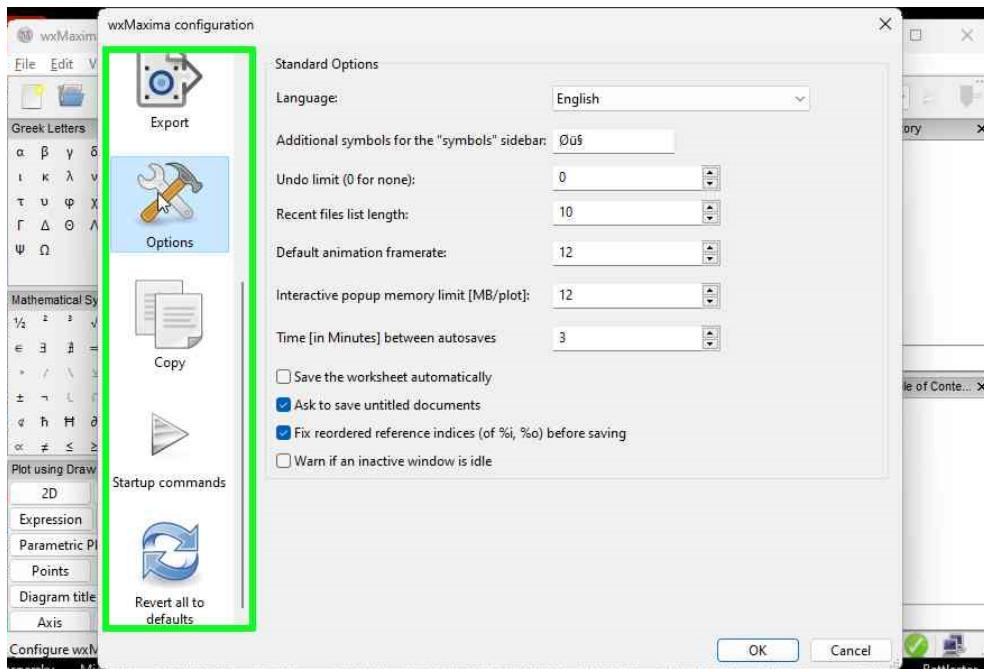


Figura A.21: Selecione o idioma Português (ou o que for de preferência)

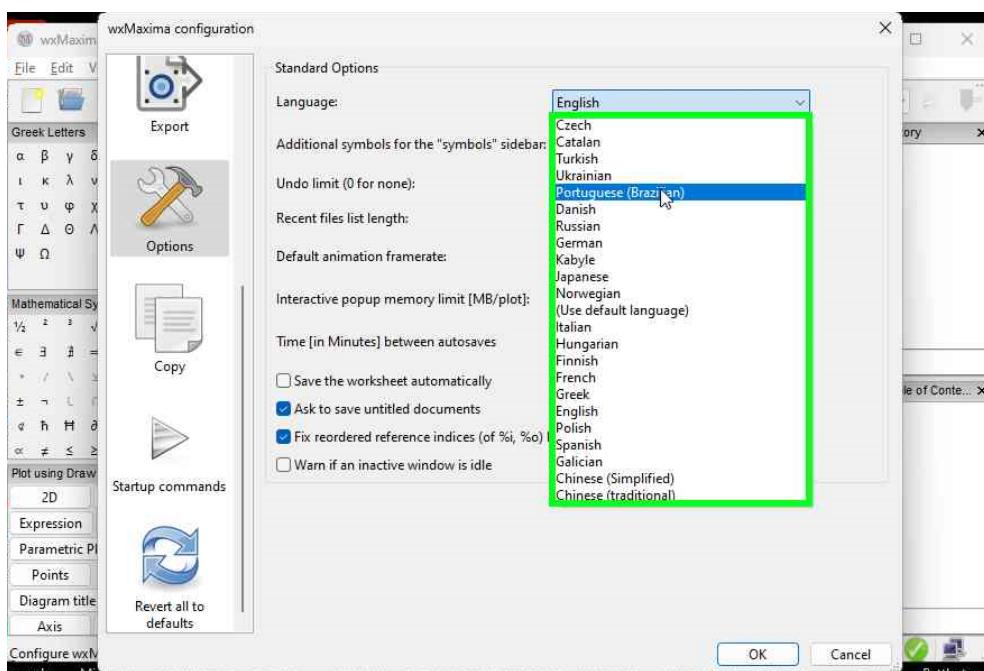


Figura A.22: Visão do tipo de idioma após alteração correspondente

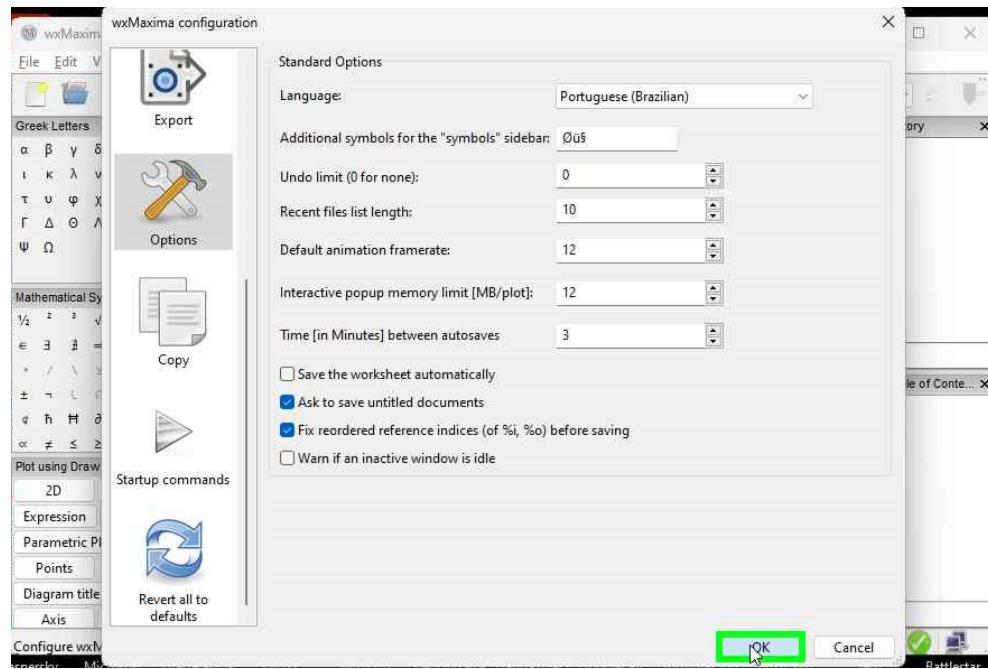
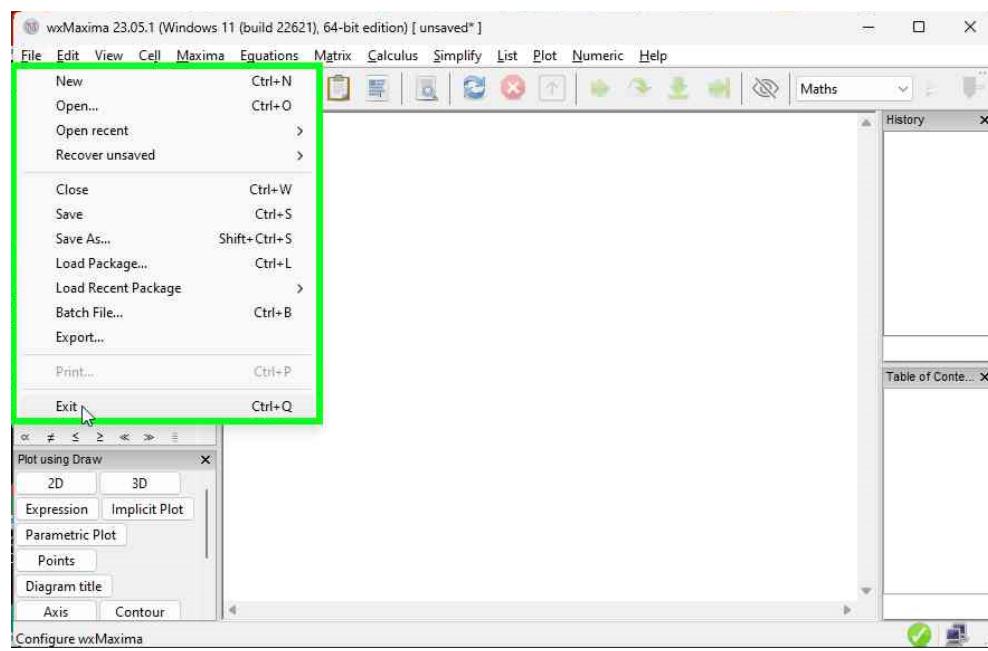


Figura A.23: É necessário reiniciar o wxMaxima para que a alteração feita surta efeito daí em diante.



A.4 Selecão do Compilador

O processamento do wxMaxima é baseado em dois tipos de compiladores: o SBCL e o CLISP. Utilizaremos o primeiro, que disponibiliza maior versatilidade no processamento com caracteres do alfabeto grego. Para configurar tal opção siga a sequênciade telas a seguir.

Figura A.24: Chamar o menu Iniciar do Windows e aceder ao portifólio de atalhos do wxMaxima

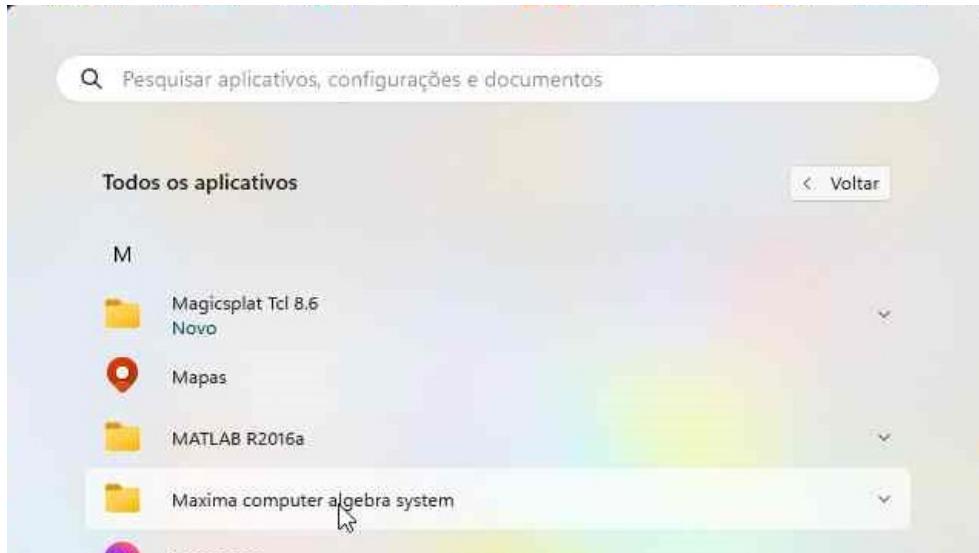


Figura A.25: Execute o comando indicado na lista suspensa

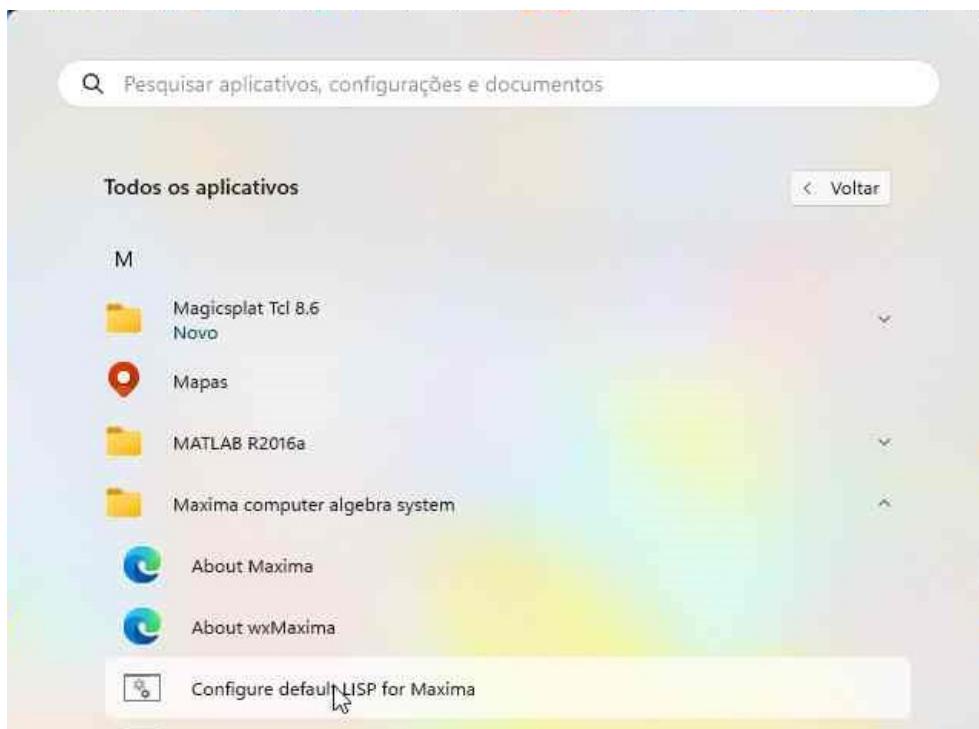
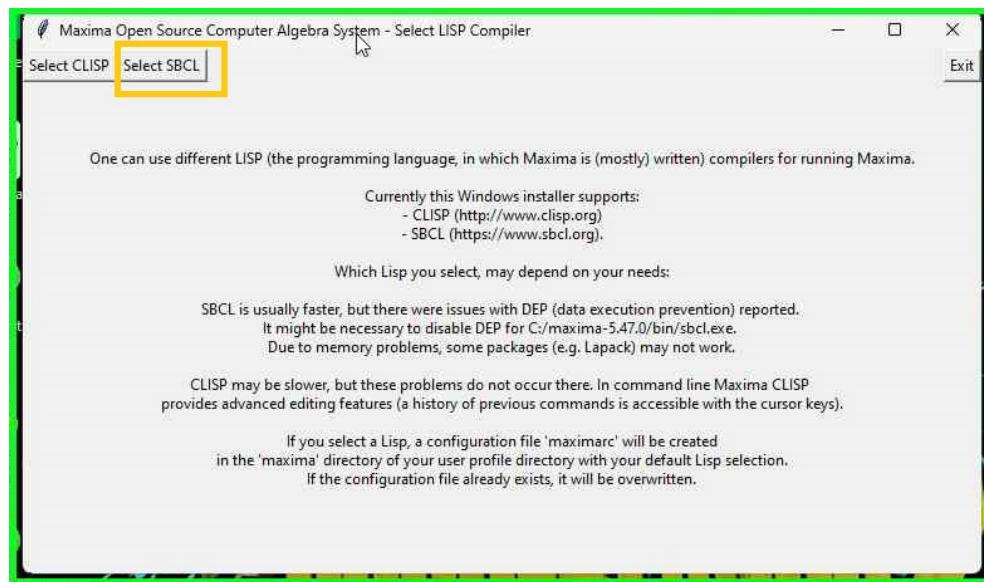
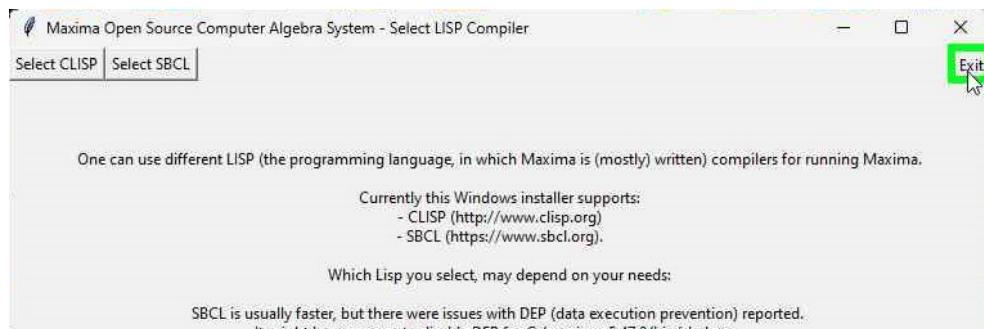


Figura A.26: Na janela de diálogo escolha a opção indentificada com o termo SBCL

Figura A.27: Finalize por meio do comando (botão) *Exit*

A.5 Dicas Diversas

Letras do alfabeto grego.

WxMaxima provides a method of entering Greek characters using the keyboard:

A Greek letter can be entered by pressing the ESC key and then starting to type the Greek character's name.

- Se seu cálculo está demorando de mais para executar, você pode tentar os menus 'Maxima->Interromper' ou 'Maxima->Reiniciar o Maxima'.
- Para fazer gráficos em coordenadas polares, selecione 'Polar' em Opções na janela de gráficos bidimensionais. Você também pode fazer gráficos em coordenadas esféricas e cilíndricas em 3D.
- As janelas do wxMaxima têm valores padrão para as entradas, um dos quais é '%'. Se você selecionou algo no documento, a seleção será usada no lugar de '%'.
- Ao aplicar funções com um argumento a partir dos menus, o argumento padrão é '%'. Para aplicar a função a um outro valor, selecione-o do documento antes de executar o comando do menu.
- Para salvar o tamanho e posição da janela do wxMaxima entre sessões, use a janela 'Maxima->Configurações'.
- Você pode acessar a última saída com a variável '%'. Você pode acessar a saída de comandos anteriores usando a variável '%on' onde n é o número da saída.
- Células de título, seção e subseção podem ser recolhidas para ocultar seu conteúdo. Para recolher ou expandir, clique no quadrado próximo à celula. Se você segurar Shift enquanto clica, todos os subníveis daquela célula também serão recolhidos/expandidos.
- Você pode esconder a saída das células clicando no triângulo no lado esquerdo das células. Isso funciona também em células de texto.
- Um 'cursor horizontal' foi introduzido no wxMaxima 0.8.0. Ele é mostrado como uma linha horizontal entre células. Ele mostra onde uma nova célula vai aparecer se você digitar ou colar texto, ou se executar um comando do menu.
- O cursor horizontal funciona como um cursor normal, mas ele opera em células: pressione as setas para cima ou para baixo para movê-lo; segure Shift enquanto move para selecionar células; pressione Backspace ou Delete duas vezes apaga a célula próxima ao cursor.
- Você pode selecionar vários células com o mouse - clique e arraste desde algum ponto entre células ou dos marcadores à esquerda - ou com o teclado - segure Shift enquanto move o cursor horizontal - e então operar na seleção. Isso é útil quando você quer apagar ou calcular múltiplas células.
- Você pode avaliar todo o documento usando o menu 'Célula->Avaliar todas as células' ou a tecla de atalho correspondente. As células serão avaliadas na ordem em que aparecem no documento.
- As janelas do wxMaxima têm valores padrão para as entradas, um dos quais é '%'. Se você selecionou algo no documento, a seleção será usada no lugar de '%'.

- Equations have several advantages over functions. For example they can be manipulated with factor(), expand() and similar functions. They can easily be introduced one into another. Also they are always printed out as 2D maths.
- In text cells bullet lists can be created by beginning a line with “*”. The number of spaces in front of the “*” determines the indentation level; Indentation can be continued in the next line by indenting the line using spaces.
- The key combination Shift+Space results in a non-breakable space.
- A plot to be embedded into the work sheet by preceding its name with a “wx”. “draw” can be replaced by “wxdraw”, plot by “wxplot” etc.

A.6 Dicas de Maxima

Maxima’s strengths are manipulating equations and in symbolic calculations. It therefore makes sense to use functions (as opposed to equations with labels) sparingly and to keep the actual values of variables in a list, instead of directly assigning them values. An example session that does do so would be:

```
/* We keep the actual values in a list so we can use them later on */
Values:[a=10,c=100];
Pyth:a^2+b^2=c^2;
solve(%,b);
result:%[2];
at(result,Values);
float(%);
```

Maxima’s “at” function allows to access to arbitrary variables in a list of results:

```
g1:a*x+y=0;
g2:b*y+x*x=1;
solve([g1,g2],[a,b]);
%[1];
result_b:b=at(b,%);
```

The “at” function allows to introduce one equation into another:

```
ohm:U=R*I;
r_parallel:R=R_1*R_2/(R_1+R_2);
result:at(ohm,r_parallel);
```

The rhs() (“right hand side”) command allows to retrieve the result of an equation in exactly the format a function would have:

```
Values:[
/* m=1.2 tons */
m=1.2*10^3,
/* 100 km/h*/
v=100*10^3/(60*60)
];
```

```
Energy:W=1/2*m*v^2;
at(Energy,Values);
W_mech:rhs(%);

g1:a*x+y=0;
g2:b*y+x*x=1;
solve([g1,g2],[a,b]);
%[1];
result_b:b=at(b,%);
```