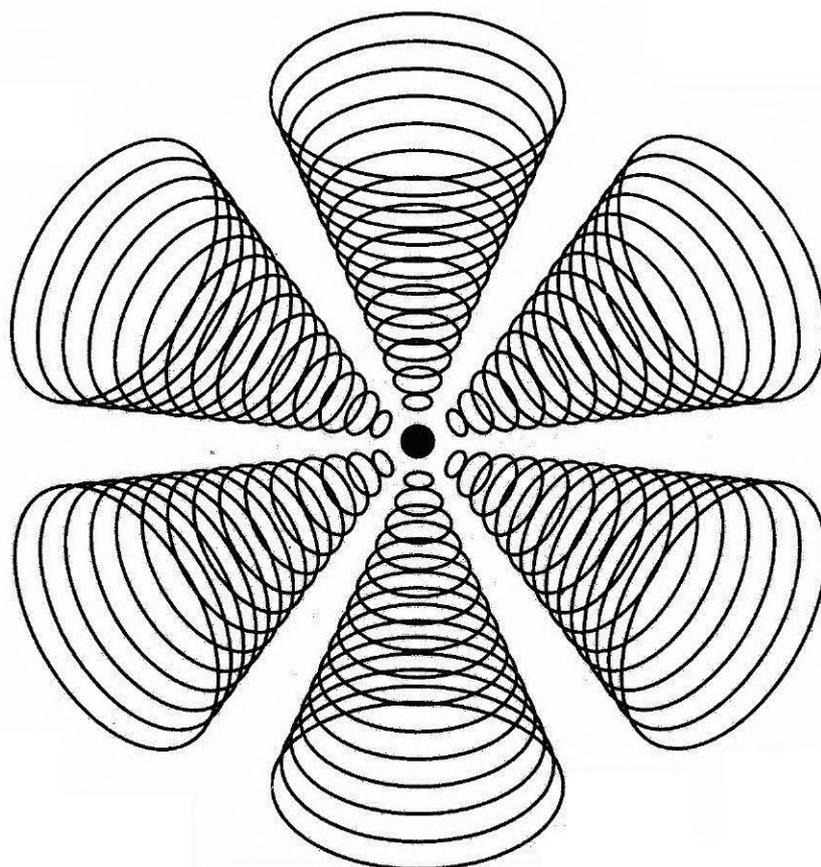


INTRODUÇÃO AO
MATHEMATICA® FOR WINDOWS

Daniel J. R. Nordemann



TRANSTEC EDITORIAL



INTRODUÇÃO AO *MATHEMATICA*®
FOR WINDOWS

DANIEL JEAN ROGER NORDEMANN

TRANSTEC EDITORIAL 

**Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)**

Nordemann, Daniel Jean Roger

Introdução ao mathematica for Windows / Daniel Jean Roger Nordemann. -- São José dos Campos, SP : Transtec, 1994.

Bibliografia.

1. Mathematica for Windows (Programa de computador)
2. Programação matemática I. Título.

94-3608

CDD-005.43

Índices para catálogo sistemático:

1. Mathematica for Windows : Ambiente operacional : Sistemas operacionais : Computadores : Processamento de dados 005.43

ISBN 85-85417-06-4

CRÉDITOS

Este trabalho refere-se ao livro de grande sucesso internacional *Mathematica: A System for Doing Mathematics by Computer*, 2a. edição, 1991, por Stephen Wolfram. Este trabalho original pode ser encontrado nos pontos de venda abaixo. A Wolfram Research Inc deu permissão por escrito à Transtec Editorial para tradução parcial ao Português do apêndice 8 do original (copyright 1988, 1991 Wolfram Research, Inc.)em nosso livro **Introdução ao Mathematica for Windows.**

Endereços:

Anacom Software

Rua Conceição, 627 - São Caetano do Sul

09530-060 S.Paulo.

Telefone:(011)453-5588 - Fax (011) 441-5563

Terasys-Informatica de Alto Desempenho, Ltda

R.Rego Freitas, 245 - Cj 21

01220-010 S.Paulo, SP.

Telefone: (011)451-4798 - Fax: (011)452-3989

Dedalus Sistemas

R. Jardim Ivone, 17, Cj. 12

04105-020 S.Paulo, SP

Telefone:(011)575-8011 - Fax:(011) 575-5813

CRÉDITOS ARTÍSTICOS

Arte de capa:

Carlos Alberto Vieira

Lourdes Beatriz de Araujo

José Dominguez Sanz

Direitos autorais

Copyright © TRANSTEC EDITORIAL LTDA
Av. Dr. João Guilhermino, 429 - 4. andar - sala 47
12210-131 - São José dos Campos, S. Paulo
Telefax -55 (123) 41.61.08
1994

MATHEMATICA® é marca registrada de Wolfram Research Inc., e está sendo usada com permissão por escrito.

O Apêndice A.8 é de direito autoral © 1988, 1991 de Wolfram Research Inc., e a reprodução parcial nesta obra tem seu consentimento por escrito.

Notas:

Wolfram Research Inc. é o proprietário dos direitos autorais do sistema de Software *Mathematica*, incluindo sem limitações aspectos como seu código, estrutura, sequência, organização, aspecto e sensação de tato, linguagem de programação e nomes de comando e compilação. Qualquer uso do sistema sem a outorga específica de uma licença de acordo, ou outro tipo de autorização legal por parte de Wolfram Research Inc., será considerado infração dos direitos autorais.

Wolfram Research Inc., não faz representações, implícitas ou explícitas, com relação ao *Mathematica*, incluindo sem limites, quaisquer garantias implícitas de mercadoria ou adequação para quaisquer propósitos. Todos os usuários devem estar cientes de que nos termos e condições em que Wolfram Research Inc., está disposta a licenciar *Mathematica* fica claro que Wolfram Research Inc., em nenhuma hipótese ficará responsável por qualquer dano indireto, consequente, ou decorrente de acaso, e que sua responsabilidade direta fica limitada ao custo de compra pago por *Mathematica*.

TRANSTEC EDITORIAL 

INTRODUÇÃO AO *MATHEMATICA* FOR WINDOWS

ERRATA

- p. 13 Linha 2 do terceiro parágrafo: substituir " insuficiência" por "à insuficiência".
- p. 15 Linha 1 do terceiro parágrafo: substituir "nas Aula 2 e 3" por "no Capítulo 4".
- p. 18 Linha 4 do quarto parágrafo: substituir " primeira vista" por "à primeira vista".
- p. 18 Linha 3 do quinto parágrafo: substituir " edição/execução" por "à edição/execução".
- p. 22 Última linha do parágrafo 1: substituir " s de" por "às de".
- p. 29 Linha 2 do primeiro parágrafo: substituir "do trabalhos do" por "dos trabalhos desenvolvidos no".
- p. 31 Linha 7 do segundo parágrafo: substituir " s entradas" por "às entradas".
- p. 93 Linha 3 do verbete **MaxBend**: substituir "num" por "numa".
- p. 96 Linha 1 dos verbetes **NDSolve[...]**: substituir "diferencias" por "diferenciais".
- p. 96 Linha 3 do verbete **NDSolve[eq, {y₁, y₂,...}, {x, xmin, xmax}]**: substituir "função" por "funções".
- p. 109 Linha 2 do verbete **Prepend[...]**: substituir "acrescentado a" por "acrescentado".

- p. 118 Linha 2 do verbete **SampleDepth**: substituir "se" por "de".
- p. 126 Linha 2 do verbete **Streams["name"]**: substituir "dos os" por "dos".
- p. 128 Linha 2 do verbete **StringSkeleton[...]**: substituir "na" por "n".
- p. 136 Linha 2 do verbete **Trace[expr]**: substituir "a expressões" por "as expressões".
- p. 136 Linha 2 do verbete **Trace[expr, form]**: substituir "a expressões usada" por "expressões usadas".
- p. 136 Linha 2 do verbete **Trace[expr, s]**: substituir "a expressões" por "as expressões".
- p. 136 Linha 3 do verbete **TraceAbove**: substituir "as" por "das".
- p. 137 Linha 3 do verbete **TraceBackward**: substituir "as" por "das".
- p. 146 Linha 2 do verbete **\$Messages**: substituir "forne" por "fornece".
- p. 149 Linha 2 do verbete **\$Urgent**: substituir "doas" por "dos".
- p. 155 Linha 2 do segundo parágrafo: substituir "um" por "uma".
- p. 158 Linha 5 do segundo parágrafo: substituir "A" por "O".

ÍNDICE

| | |
|--|-----|
| CAPÍTULO 1 - O <i>Mathematica</i> | 9 |
| CAPÍTULO 2 - Operações e funções..... | 15 |
| CAPÍTULO 3 - Gráficos..... | 21 |
| CAPÍTULO 4 - Programação..... | 23 |
| CAPÍTULO 5 - Entradas e Saídas..... | 27 |
| CAPÍTULO 6 - Otimização do tempo de execução..... | 31 |
| CAPÍTULO 7 - Matemática simbólica..... | 35 |
| CAPÍTULO 8 - Orientação, otimização e exercícios..... | 37 |
| CAPÍTULO 9 - Lista dos Objetos do <i>Mathematica 2.21</i> | 41 |
| | |
| APÊNDICE A - LISTA DE NOTEBOOKS..... | 151 |
| APÊNDICE B - MICRODICIONÁRIO..... | 161 |
| | |
| LIVROS E REVISTAS..... | 165 |

APRESENTAÇÃO

Algumas idéias dão muitos frutos. Acredito que a idéia do Dr. Nordemann de escrever uma introdução para um dos programas mais completos de matemática para micros, vai produzir também muitos frutos.

É preciso lembrar que o ambiente de trabalho de vários Institutos de pesquisa e Universidades, não pode mais existir sem a presença do "micro" e seus "softs". Este conjunto, que algumas vezes é chamado de "Informática", torna-se cada vez mais complexo e especializado, e na grande maioria dos casos as instruções de programas e catálogos vem escritas e explicadas em língua estrangeira. É, assim, um transtorno para quem está começando, um transtorno que pode tornar-se um pesadelo quando se trata, por exemplo, de conquistar ou manter um emprego.

O Dr. Nordemann caracteriza-se por sua grande experiência na vida de pesquisador. Apaixonado pelo micro, e com sua paciência habitual, começou a dar "dicas" para os menos experientes, especialmente aqueles envolvidos com o programa de pós-graduação do INPE. O interesse foi crescendo a tal ponto que foi organizada uma disciplina especial para tratar do assunto. Dos rabiscos iniciais de dicas resultou uma apostila, e finalmente, da apostila surgiu o presente livro.

O autor está de parabéns. Soube otimizar um trabalho que vai beneficiar a muitos.

Algumas idéias dão muitos frutos. Esta, é uma delas.

Volker W.J.H.Kirchhoff
Diretor Substituto
INPE

PREFÁCIO

A pesquisa científica precisa de ferramentas. Hoje, ao lado dos instrumentos da pesquisa experimental, os microcomputadores e softwares matemáticos passaram a ser ferramentas imprescindíveis. O uso de uma linguagem de alto nível para a programação matemática passou a ser o complemento indispensável de alguns cursos de pós-graduação em Geofísica Espacial. A falta de manuais em geral e em particular em português me levou a redigir algumas páginas de dicas para ajudar os alunos a começar a usar o *Mathematica*, um dos melhores ambientes matemáticos do momento.

Estas primeiras dicas transformaram-se numa apostila com os oito primeiros capítulos (Apresentação; Operações e funções; Gráficos; Programação; Entradas e Saídas; Otimização do tempo de execução; Matemática simbólica; Orientação, otimização e exercícios). A pedido dos alunos, para permitir a auto-suficiência até um certo nível de proficiência no uso do *Mathematica*, foi acrescentado o Capítulo 9 (Lista dos Objetos do *Mathematica*), redigido a partir de nossa experiência pessoal crescente e do Apêndice A8 do livro de Stephen Wolfram (*Mathematica, A System for Doing Mathematics by Computer*, Addison-Wesley, 1991). Uma bibliografia especializada e uma lista de exemplos de notebooks usados nos cursos foram acrescentados, assim como um microdicionário inglês-português de termos mais usados, para facilitar a iniciação dos alunos menos familiares com a língua de Shakespeare.

São José dos Campos, 1994

Daniel J. R. Nordemann

CAPÍTULO 1

O MATHEMATICA

- O *Mathematica* representa um ambiente de cálculos e operações matemáticas que usa uma poderosa linguagem de alto nível, com códigos muito elaborados. O uso do *Mathematica* é valioso para cálculos complicados, sofisticados, que podem incluir operações sobre símbolos literais e fórmulas, como derivação e integração.

- O *Mathematica* compreende duas partes que trabalham em conjunto, mas que podem, de uma certa maneira ser usadas separadamente: o "front end" ("terminal da frente") que é a interface *Mathematica*-usuário (com o uso do monitor, teclado, arquivos etc.) e o "kernel" que realiza as operações matemáticas nos processadores (com o uso de espaço na memória RAM e no disco rígido).

- Ambiente: O "front end" oferece um ambiente de trabalho com o padrão do Windows, isto é, bem semelhante ao do Word for Windows, por exemplo. Dispõe-se de menus de cortina, ícones de ferramentas, todos acionados pela seta movida pelo mouse. A tela principal, para edição, resultados, gráficos etc. será, ao longo do desenvolvimento do trabalho, dividida em espaços com funções diferentes: as células ("cells"). Um trabalho, com um conjunto de células chama-se "notebook" (ver mais adiante).

- O *Mathematica* for Windows faz uso de um sistema de Notebooks (Tradução literal: Livro de Apontamentos) que são gravados simultaneamente em arquivos ASCII com nomes *.ma e binários nomes *.mb. No caso do *Mathematica* for Windows, uma palavra

curta que corresponde melhor ao que é um notebook poderia ser "sessão de trabalho" ou "pacote". Por sinal, o equivalente (gravado em ASCII com um nome *.m) do notebook no ambiente do *Mathematica* for DOS é chamado de "package".

- Um notebook é constituído de células (Cells). A primeira vista este sistema é muito mais complexo do que o do *Mathematica* for DOS (Edição em ASCII, dentro ou fora do *Mathematica*, no editor ASCII da escolha do usuário e gravação de programas em arquivos ASCII). Esta complexidade é devida ao número relativamente elevado de tipos de células (editáveis ou não, ativas ou não, resultados etc.). Entretanto, devido à integração no ambiente Windows (menus e comandos idênticos ou semelhantes), as vantagens deste sistema aparecem logo, para o melhor aproveitamento pelo usuário.

- Recomenda-se uma abordagem pragmática, nos primeiros trabalhos práticos, sem esforçar-se na memorização dos diferentes colchetes que identificam o tipo das células; é suficiente abrir um notebook e começar a operar com exemplos simples.

Algumas possibilidades entre muitas

- Matrizes. O *Mathematica* oferece bastante facilidade na operação de matrizes, incluindo o fato de poder aplicar uma operação ou uma função a uma matriz e de se obter a aplicação desta a cada elemento da matriz.
- Funções e operação sobre as funções, operações sobre símbolos, derivação (`Derivative[1][f]`), integração (`Integrate[f,x]`) etc.
- Leitura e gravação de arquivos em ASCII para entrada de dados e saída de resultados.
- O *Mathematica* possui um sistema de manipulação e conversão de unidades, prático para cálculos de Física, Química etc.
- O *Mathematica* permite gerar gráficos de duas e de três dimensões, com bastante recursos (cores, pontos de visualização no caso dos gráficos 3D etc.). Deste ponto de vista, o *Mathematica* é um dos melhores softwares de matemática disponíveis no mercado atual.
- Existem ainda muitas outras possibilidades (não descritas aqui) como som, animações etc.

Sobre o tempo de desenvolvimento e de execução

- A grande vantagem do *Mathematica* em geral reside no tempo de desenvolvimento de programas (Exemplo esquematizado na realização de trabalhos equivalentes: Programa em C, com listing ASCII de 8 páginas, desenvolvido em meses de trabalho e rodando em 15 minutos num micro 386 25 MHz; o equivalente no *Mathematica*, com meia página de instruções, desenvolvido em uma tarde porém rodando em algumas horas).
- Por ser uma linguagem interpretada de alto nível, o uso do *Mathematica* para o DOS leva, nos micros de tecnologia mais antiga (386), a tempos de cálculo muito longos quando são tratados

grandes volumes de dados e/ou quando há muitas repetições de loop. Nos microcomputadores de tecnologia mais recente (486 e Pentium) e nas workstations, este inconveniente é minimizado. Porém não se deve esquecer que o uso de uma linguagem compilada (como o C ou o C++) é mais adequado para tratar grandes volumes de dados e/ou quando há muitas repetições de loop.

- A procura da otimização de um programa (notebook) do *Mathematica* compreende a concisão (redigir um programa tão curto quanto possível), a clareza (no uso das instruções e na adição de comentários), a apresentação que facilita sua leitura assim como a redução do tempo de processamento (escolha de instruções e de loops mais rápidos, uso de funções compiladas ou com memória). A melhoria da apresentação é feita pela adição de espaços ou linhas puladas, onde isto é permitido, pela escolha do lugar de corte das linhas quando for necessário, pelos deslocamentos progressivos para direita etc.

- A melhor solução, para certos problemas, pode ser a incorporação e/ou o uso externo de programas compilados (em C ou C++ por exemplo) a programas (notebooks) do *Mathematica*.

- Mas uma das grandes vantagens do *Mathematica* for Windows, pelo fato de ser "for Windows", reside no fato de se poder efetuar cálculos enquanto que o usuário pode continuar a trabalhar no microcomputador, no *Mathematica* (certas ações, sim, outras, não) ou fora do *Mathematica*, em outras aplicações do Windows. Esta possibilidade assimila seu uso ao de uma Workstation (Multitask).

- Outra vantagem do *Mathematica* for Windows, também pelo fato de ser "for Windows", reside no fato de poder transferir para, ou de, outros softwares compatíveis com Windows, textos, gráficos, imagens etc. graças ao uso do Clipboard (Espaço de memória para transferência rápida).

Exigências

- O *Mathematica* existe em versões para micros a partir de AT386: versão para AT386 no ambiente DOS e versão para ambiente Windows em micros 386 ou 486. Existe também uma versão para workstations (SUN).

- A Versão 2.2 para Windows exige pelo menos 6 MB de RAM e no Winchester 7,31 MB ocupados pelo software no diretório WNMATH22 assim como bastante espaço de memória livre para uso temporário durante os cálculos (A título de exemplo, num micro 386 25 MHz, com o *Mathematica* já instalado e com 40 MB livres no Winchester, o próprio Windows recomenda da ordem de 20 MB de memória virtual ("Virtual Memory"), para arquivos de permutação ("Swap Files"). A escolha do valor atribuído memória virtual é feito através do acesso no "Control Panel" do "Main" a "386 Enhanced" e "Change". Estes valores são realmente mínimos absolutos, pois é melhor dispor de muito mais RAM para agilidade nos cálculos (e correlativamente menos uso dinâmico da "Memória Virtual" do Winchester) e mais espaço livre no Winchester, se tiver pouca RAM. Informações a respeito podem ser obtidas no arquivo WINFEAT.MA no subdiretório DOC do Diretório WNMATH22.

- Uma limitação freqüentemente encontrada no uso do *Mathematica* é ligado insuficiência de memória RAM do microcomputador usado. Esta insuficiência aparece no caso de o resultado dos cálculos comportar, por exemplo, gráficos a 3 dimensões ou som. Depois de se ter carregado o kernel (Tradução literal: semente, âmago; sentido aproximado: núcleo para os cálculos, motor) e de os cálculos estarem em andamento por um certo tempo (muitos minutos), o *Mathematica* pode obrigar a parar e voltar ao DOS. Ele permite em geral gravar o Notebook, antes de ser desativado.

CAPÍTULO 2

OPERAÇÕES E FUNÇÕES

1. Operações simples

- Ponto importante: Na edição/entrada de instruções, <Enter> serve apenas para mudar de linha (mesmo no meio de uma instrução ou de uma seqüência de instruções). Para executar uma instrução ou uma seqüência de instruções, é preciso colocar o cursor de edição sendo em qualquer parte da instrução ou da seqüência de instruções e digitar <Ins> (só). Outros comandos são equivalentes para executar uma instrução ou uma seqüência de instruções: <Shift><Enter>; <5> do teclado numérico, que não tem nenhuma outra função quando o <NumLock> não está ativado; Evaluate Selection do menu Action; e, simplesmente, acionar com o mouse o ícone de avaliação, isto é, o icosaedro da barra de ferramentas.

- O cursor/apontador ("pointer") tem muitas faces. Existe simultaneamente o cursor, como lugar de edição (inserir, modificar caracteres), quando e onde isto é permitido, e a seta, ou equivalente de seta, para, por exemplo, apontar um lugar, uma área (como gráfico), um ícone ou um colchete de célula. Nestas últimas funções, o apontador tem várias formas gráficas: seta, traço vertical, traço horizontal, círculo cruzado etc. Só a prática ensinará a distinguir entre as diferentes funções e aparências.

- Instrução por instrução (a programação descrita nas Aula 2 e 3). As instruções já digitadas podem ser modificadas e executadas de novo. Elas podem ser copiadas e colocadas num outro lugar para uso e/ou edição/modificação pelo uso de comandos "naturais" do ambiente do Word for Windows (Cut, Copy and Paste).

- Fim da sessão de *Mathematica*: Sempre que terminar a sessão, sair do *Mathematica* para não deixar no Winchester volumosos arquivos temporários. Para terminar a sessão, usa-se o Exit do Menu de cortina File

- Operações:

a+b adição

a-b subtração

a*b ou a b

 multiplicação (O * da multiplicação pode ser omitido e substituído apenas por um espaço)

a/b divisão

a^b potência

- Existe a escala habitual de prioridade entre as operações elementares (em caso de ambigüidade ou dúvida, parênteses esclarecedoras devem ou podem ser acrescentadas).

- Constantes que já existem:

I = Sqrt[-1] (raiz quadrada de -1)

Pi = 3.141592

Degree = Pi/180 (fator de conversão grau para radianos)

E = 2.71828

- Para os valores numéricos envolvendo potências de 10, jamais usar E; na entrada de dados, usar o "caret" (Exemplo: 3 10^-3); o *Mathematica* devolve os resultados usando sobrescrito depois do 10 (Exemplo: 3 10⁻³).

- Observar o uso generalizado do ponto decimal no lugar da vírgula (no *Mathematica* pelo fato de ser um software americano, e, em consequência disto, neste curso).

- Operações compostas:

$b+=c$ equivalente a $b=b+c$
 $b-=c$ equivalente a $b=b-c$
 $b*=c$ equivalente a $b=b*c$ ou seja $b=b \cdot c$
 $b/=c$ equivalente a $b=b/c$
 $b++$ equivalente a $b=b+1$
 $b--$ equivalente a $b=b-1$

Estas instruções vêm do C (seus símbolos compostos +=, -=, *=, /=, ++ e -- não podem comportar espaço no meio).

2. Funções elementares

- Nomes de funções do *Mathematica* : começam por maiúscula, o resto em minúscula. Exemplo: Sqrt[], Exp[], Sin[], Cos[], Tan[], ArcSin[], ArcCos[], ArcTan[], Abs[], Max[a,b,c,...], Min[a,b,c,...] etc.

- Outras funções úteis: Log[] (logaritmo natural, de base e), Log[b, x] (logaritmo de base b), Round[] (inteiro mais perto do argumento), Mod[n, m] (n módulo m), Random[] (número pseudo-aleatório compreendido entre 0 e 1).

- Caso do ponto de exclamação: no *Mathematica*, o ponto de exclamação é empregado com três funções diferentes: n! (fatorial de n); !expr é a negação lógica da expressão lógica expr; !comando isolado no início da linha executa comando dentro do DOS (o comando tem que ser um comando válido do DOS).

- Nomes de variáveis, tabelas etc. : recomenda-se usar só minúscula. O *Mathematica* recomenda o uso de nomes de variáveis descrevendo por inteiro as grandezas correspondente, mesmo se isto leva a nomes compridos; não abreviar porque se abreviar demais, pode levar a confusão. Evidentemente, não usar para suas variáveis e funções, os nomes que o *Mathematica* já usa. O *Mathematica* sempre distingue entre minúsculas e maiúsculas e não tolera nenhuma confusão entre caixa alta e caixa baixa!

- A função `Table[objeto, {parâmetro, início, fim}, {outroparâmetro, início, fim}]` é muito útil para montar tabelas cujo conteúdo são valores, funções dos parâmetros, etc. A função `Table[]` serve até para encurtar o tempo de processamento, em substituição de loops. Os objetos inseridos na tabelas podem ser outras tabelas, havendo assim uma possibilidade suplementar de montar tabelas multidimensionais. As tabelas inseridas numa mesma tabela não precisam ter todas, ao mesmo nível, o mesmo tamanho, o que permite montar, por exemplo, tabelas triangulares ou trapezoidais. Os objetos inseridos numa tabela podem ser os resultados respectivos da última instrução de seqüências de instruções separadas por ponto-vírgula.

- Um célula pode comportar várias operações separadas por ponto-vírgula (isto já não é programação?). Consultar o módulo sobre a programação.

- Condições: A instrução é semelhante da linguagem C: `If[condição, casoverdadeiro, casofalso, casonenhumdosdois]`; Nesta condição, `casoverdadeiro`, `casofalso` e `casonenhumdosdois` podem ser operações, funções ou procedimentos (combinações, seqüências de operações e funções). Existe outra escritura mais simples: `casoverdadeiro /; condição`

- As diferentes funções das células (entrada de instruções e/ou fórmulas; resultados simbólicos/literais ou numéricos; gráficos etc.) aparecem pelo desenho do topo dos colchetes que as delimitam (na direita da tela). Isto pode parecer primeira vista uma complicação que não existia no *Mathematica* for DOS.

- Uma das vantagens oferecidas pela existência das células é a facilidade para passar da edição/execução de uma instrução isolada edição/execução de uma seqüência de instruções ou programa. As instruções em seqüência são separadas apenas por ";" e/ou mudança de linha (<Enter> simples). Sem mudança de linha, o separador ";" é evidentemente obrigatório. Com mudança de linha, ele é facultativo: com ";" o *Mathematica* executa a instrução e não mostra o resultado na tela; sem ";" o *Mathematica* executa a instrução e mostra o resultado na tela.

- Uma ferramenta do *Mathematica* for Windows pode ajudar a entrar-se mais rápido com os diferentes tipos de células: trata-se da possibilidade de se ter fundos de cores diferentes e estilos diferentes (fontes, tamanho dos caracteres etc.) para os diferentes tipos de células (acesso através de Style, Cell Style, Edit Notebook Styles, Edit Default Styles etc). Recomenda-se a padronização dos estilos e das cores entre todos os usuários dos mesmos microcomputadores.

- Sugestão de cores de fundo e fontes:

[[linha,coluna]]

| | | |
|--------|---------------------|-------------------|
| Título | Arial 16 Bold | Azul médio[[1,6]] |
| Texto | Courier New 10 | Azul claro[[2,5]] |
| Input | Courier New 12 Bold | Amarelo [[3,2]] |
| Output | Courier New 12 | Verde [[1,3]] |

- No *Mathematica* for Windows, os notebooks encontram-se em arquivos com extensão .ma. Por serem misturas de textos, gráficos, convertidos no formato POSTSCRIPT, estes arquivos só podem ser lidos no *Mathematica* for Windows. Já os programas do *Mathematica* for DOS são arquivos em ASCII e com extensão .m e podem ser lidos e editados com qualquer editor ASCII.

CAPÍTULO 3

GRÁFICOS

1. Geração de gráficos a duas dimensões usando as instruções: `Plot[f,{x,xmin,xmax}]`; `ListPlot[{a,b,c}` ou `ListPlot[{{x,y},...}]`. As instruções que contém a palavra `Plot` em seu nome são específicas para gráficos de dados contidos em tabelas. As demais referem a gráficos de funções.

2. Geração de gráficos a três dimensões (padrão ou com controle de parâmetros diversos, incluindo a escolha do ponto de visualização): `Plot3D[f,{x,xmin,xmax}]`; `ListPlot3D[{a,b,c}]`. São funções que consomem grandes volumes de memória na memória RAM e no Winchester e que podem levar (em certos microcomputadores) interrupção forçada da sessão de trabalho.

3. Geração de gráficos ou mapas de contorno que podem ser facilmente obtidos a partir dos mesmos dados.
`ContourPlot[f,{x,xmin,xmax},{y,ymin,ymax}]`; `ContourGraphics[matriz]`

4. Para as funções gráficas, o controle de parâmetros é da maior importância. Entre os mais usados são: `AspectRatio -> 1`; `Axes -> False`; `AxesLabel -> None`; `ColorFunction -> Automatic`; `PlotLabel -> None`; `PlotRange -> Automatic`; `Ticks -> Automatic`; `ColorFunction -> Hue` (para se gerar cores nos gráficos de contorno). Observar que no caso das funções com controle de parâmetros, o efeito não é apenas restrito à instrução dentro da qual o parâmetro é controlado, mas é válido "doravante", isto é, a partir de sua primeira execução dentro do notebook.

5. Para deslocar um gráfico, pressionar a tecla esquerda do mouse, deixando-a pressionada e mover o mouse até colocar o gráfico no lugar desejado (drag). Usar os marcadores (handles) de canto e do meio dos lados para modificar o tamanho dos gráficos, exatamente como no Word for Windows.

6. Para ler coordenadas num gráfico, e inseri-las automaticamente numa tabela, pressionar a tecla Ctrl do teclado, deixando-a pressionada e pressionar a tecla esquerda do mouse acima dos pontos a serem digitalizados ou deixando-a pressionada ao longo da curva a ser digitalizada. O cursor passou a aparecer como uma cruz fina em formato de colimador). Pontos aparecem nos lugares marcados ou ao longo da trajetória. No fim do processo, soltar ambas as teclas pressionadas, mover o cursor no ícone Copy e pressionar a tecla do mouse uma vez. Levar o cursor no lugar de edição (dentro de uma célula conveniente), pressionar a tecla do mouse uma vez para aí colocar o cursor de edição. Levar o cursor no ícone Paste e pressionar a tecla do mouse uma vez. As coordenadas dos pontos gerados dentro do gráfico aparecem entre chaves, numa tabela. Um nome pode ser dado a esta tabela colocando o nome escolhido seguido de um sinal = antes da tabela e executando esta instrução. As coordenadas lida são expressas relativamente ao tamanho da página e podem ser facilmente convertidas em qualquer unidade pela aplicação de uma formula linear. Esta operação pode ser feita tanto num gráfico gerado no *Mathematica* quanto numa imagem qualquer importada (por exemplo no formato *.bmp) no *Mathematica*. As fotografias e as imagens desenhadas ou impressas podem ser transferidas em arquivos de imagens usando um scanner. Assim sendo, o uso de um scanner e do *Mathematica* permite realizar funções equivalentes a de uma mesa digitalizadora.

8. Vários exemplos de opções de gráficos coloridos e preto/cinza/branco são dados no notebook "colortst" (gráficos de duas e três dimensões), com o uso das opções de matiz Hue[matiz, saturação, brilho] e de composição de cor RGBColor[vermelho, verde, azul].

9. Vale salientar que os gráficos coloridos (na tela do monitor) são facilmente impressos em impressoras coloridas (devidamente instaladas no Windows) e que eles podem ser impressos em impressoras não coloridas (impressora matricial ou laser) com a conversão automática das cores em tons de cinza.

CAPÍTULO 4

PROGRAMAÇÃO

Programação no *Mathematica* for DOS:

1. Quem não tiver necessidade de usar o *Mathematica* for DOS, ou de usar e transferir "Packages" (*.m) ou programas (*.m) feitos para o *Mathematica* for DOS, pode passar diretamente na parte Programação no *Mathematica* for Windows.
2. A edição dos pacotes é feita fora do *Mathematica* com qualquer editor ASCII disponível escolhido pelo usuário (o Norton Editor NE por exemplo). Dentro do *Mathematica* for DOS, o usuário pode chamar o mesmo editor uma vez que o *Mathematica* for DOS não comporta um editor embutido.
3. Chamada dos programas (com detecção de erros de sintaxe) por <<nome.m sendo que os listing ASCII de programas do *Mathematica* comportam a extensão .m . O listing ASCII do programa começa pelo "cabeçalho":
nome/: nome := (o programa).
4. Execução digitando: nome <Enter>
5. Regras de programação e convenções de uso de caracteres e símbolos: estas são as mesmas do que no *Mathematica* for Windows, porque elas dependem do "kernel" e não do "front end" e são dadas mais adiante.

Programação no *Mathematica* for Windows:

1. De um certo ponto de vista, quando se trabalha com uma linguagem de alto nível, em um ambiente como o do *Mathematica* for Windows, passa-se sem dificuldade, quase sem notar, do modo instrução por instrução ao modo da programação.
2. Regras de programação e convenções de uso de caracteres e símbolos:

| Carácter: | Uso: |
|-------------------------|---|
| ; | separar as instruções numa seqüência de instruções (diferente do C) |
| , | separar os componentes dentro de uma instrução (diferente do C) |
| [] | variáveis de funções |
| [[]] | índices de vectores (uma dimensão só) |
| [[,]] ou [[]][]] | matrizes, tabelas (mais de uma dimensão) |
| () | agrupar instruções; delimitar procedimentos (procedures) no caso do uso de Goto |
| { } | empacotamento de elementos de tabelas |
| { , } | limites e valores iniciais de índices e variáveis |
| (* *) | comentários (lembra o Pascal e /* */ do C) |

3. Alguns conselhos gerais:

- Se tiver loops e se a execução pode tomar muito tempo, verificar se tudo está correndo bem com visualização de resultados parciais.
- Evitar a armadilha da execução parcial das instruções para as quais os resultados ficam parcialmente como fórmula (quando não é desejado); além disto este efeito "entope" a memória usada pelo *Mathematica* até saturação (em particular no caso de funções de

funções de funções...). A instrução `N[A,d]` dá, se for possível, o valor numérico de A com d dígitos de precisão.

- **Action, Interrupt** (em cinza claro, isto é, desativado/inativo): de maneira ainda não explicada, o comando `Interrupt` destinado a parar uma avaliação no meio da execução de um programa, é desativado. Às vezes, retomar o controle (o que pode ser, esperando o fim dos cálculos em andamento), sair do *Mathematica* e recarregá-lo de novo resolve, isto é, reativa o comando `Interrupt`. Às vezes não.

- **Loops**: O equivalente das principais instruções de loop que existem nas outras linguagens científicas existem no *Mathematica*, com a seguinte sintaxe: `For[início, teste, incrementação, corpo]`; `While[teste, corpo]`; `Do[expressão, {i, imin, imax}]`.

No caso de loops usados apenas para encher tabelas com valores, é recomendado usar a instrução

`Table[conteúdo, {i, imin, imax}, {j, jmin, jmax}]`

muito mais econômica em tempo de computação, que as instruções de loop acima enumeradas,

- **Goto**. Este comando funciona da seguinte maneira:

(... `Goto[name]; ...; Label[name]; ...`). Observar que os parênteses da frase anterior fazem parte do uso do `Goto`, uma vez que há necessidade de "empacotar o `Goto[name]` e o `Label[name]` correspondente dentro de um par de parênteses que delimitam um procedimento (procedure). O uso de `Goto` não é recomendado (O mesmo do que no C: em linguagem estruturada, não se usa `Goto`). O comando `Goto` existe, porém não é conveniente usá-lo, por três razões: pelo motivo anterior; porque não é muito incerto usá-lo (vezes funciona, outras, não; colocar alguns `Goto[]` no mesmo programa é ainda mais difícil). Dentro de um loop (`For`, `While`), o uso de `Break[]` ou de `Continue[]` resolve a maior parte dos problemas relativos a ação a tomar em função de um teste realizado `If[, , ,]`, para sair do loop (`Break[]`), ou incrementar a variável do loop e continuar (`Continue[]`).

CAPÍTULO 5

ENTRADAS E SAÍDAS

1. Saída temporária do Kernel para execução de comandos no DOS (!)

2. Leitura de arquivos de Notebooks (*.ma e *.mb do *Mathematica* for Windows). Os pacotes do *Mathematica* for Windows são gravados no disco rígido ou em disquete **simultaneamente** sob forma de arquivos ASCII (ASCII files) com a extensão .MA e sob forma de arquivos binários (Binary files) com a extensão .MB. Os arquivos ASCII .MA contém a seqüência das instruções em caracteres ASCII e podem ser modificados, caso precisaria, num editor ASCII e posteriormente lidos pelo "front end" do *Mathematica* for Windows. Os arquivos binários .MB contém a seqüência das instruções e dos resultados dos cálculos obtidos pelo "kernel" e apresentados pelo "front end", incluindo os gráficos. Por este motivo, os arquivos binários .MB podem ser muito mais volumosos do que os arquivos ASCII. Além disto, os arquivos .MB não podem ser modificados ou alterados fora do *Mathematica* for Windows uma vez que não poderiam mais ser incorporados pelo *Mathematica* for Windows. Mas se isto acontecer, isto não acarreta prejuízo uma vez que o arquivo ASCII .MA correspondente permite reconstituir (fazendo "trabalhar" de novo o "kernel") a seqüência dos cálculos e resultados.

3. Gravação de arquivos de Notebooks. Colocar a seta do cursor acima do ícone que representa um disquete e pressionar uma vez a tecla esquerda do mouse, ou usar o menu File, Save ou SaveAs (a depender do caso); Estes comandos são os mesmos do que os do Word for Windows.

4. Leitura de arquivos ASCII (de programas *.m do *Mathematica* for DOS e de dados). Usar o menu File, Open/Import ... ou <<nome.m numa linha do notebook.

5. Gravação de arquivos ASCII: Put[expressão, outras, nome] ou expressão >> nome grava no arquivo nome (no diretório default) a sucessão de expressões, separadas por vírgulas (após apagar seu conteúdo anterior, se já existia); PutAppend[expressão, outras, nome] ou expressão >> nome grava a sucessão de expressões, separadas por vírgulas no arquivo nome (do diretório default), depois do seu conteúdo anterior (se já existia).

6. Uso do "clipboard" para transferir (isto é, mover ou copiar) instruções, células, dados etc. dentro da sessão de trabalho (notebook), de dentro para fora ou de fora para dentro. O uso habitual do clipboard é de permitir a transferência de um caráter ou de um grupo de caracteres, no modo da edição, de qualquer lugar para qualquer lugar da sessão de trabalho, exatamente como na edição no Word for Windows. Mas ele permite também isto entre sessões diferentes com vários notebooks abertos simultaneamente, com acesso instantâneo pelo menu de cortina Window. Isto aplica-se também a qualquer célula ou elemento de célula. E isto aplica-se também a transferências do *Mathematica* para outros softwares do ambiente Windows, e reciprocamente, isto é, de dentro para fora e de fora para dentro.

7. Chamada de programas em outras linguagens dentro de um programa do *Mathematica*. Existe a possibilidade de rodar qualquer programa compilado (em qualquer linguagem) e executável no ambiente DOS pelo uso de !. Evidentemente, deve-se evitar incompatibilidades (como, por exemplo, tentar escrever em arquivos já abertos no ambiente Windows).

8. Observações:

- É sugerido criar na raiz do C: um diretório M onde todos os arquivos do trabalhos do *Mathematica* serão arquivados (*.m se tiver, *.ma, *.prn com dados, *.asc com resultados ASCII etc., sempre com extensão padronizada para cada tipo de arquivo para evitar confusão). A justificativa para a existência deste diretório é que é mais simples, ganha tempo e evita problemas indicar, nos programas e nas janelas de entrada e saída, apenas nomes de arquivos sem a indicação do caminho.

- No início, parece difícil no *Mathematica*, através de comandos no Notebook ou num programa, utilizar um diretório de trabalho diferente do que foi indicado na configuração. Por isto, pode-se impor o diretório de trabalho (Working Directory) na configuração permanente, isto é feito através do Program Item Properties do ícone do *Mathematica* acessado pelo File Properties do Program Manager após clicar uma vez no ícone do *Mathematica* (o ícone do icosaedro).

- Se há necessidade de mudar de diretório de trabalho durante uma sessão de *Mathematica* ou na execução de um programa, usar o comando `SetDirectory["\nome"]` sem indicação do drive. O funcionamento deste comando pode ser verificado com a execução de uma célula que contem:

```
SetDirectory["\nome"];
```

```
!dir /p
```

Observar que a instrução que começa por um ponto de exclamação (para execução no DOS) não pode estar na mesma linha do que a instrução anterior.

- Tanto no *Mathematica* for DOS quanto no for Windows, há sempre necessidade de indicar nas janelas de entrada ou equivalentes o nome do arquivo com sua extensão entre aspas. Se o Windows for configurado para os acentos do português (Lembrete: foi feito no International do Control Panel do Main com "Brazil; English (International); US-International..."), acionar sucessivamente as teclas <Aspa> e <Espaço> para evitar vogais acentuadas, cedilha ou falta de aspa.

9. Resumo das principais operações sobre arquivos:

| | |
|-------------------------|---|
| <<nome.m | lê um arquivo ASCII de programa para o <i>Mathematica</i> |
| expr >> file | escreve a expressão expr no arquivo file |
| Put[expr, "file"] | escreve a expressão expr no arquivo file |
| expr >>> file | acrescenta a expressão expr no arquivo file |
| PutAppend[expr, "file"] | acrescenta a expressão expr no arquivo file |
| !!file | mostra o conteúdo de file |
| Save[nome, var] | grava no arquivo nome a(s) variável(is) var |
| !instr | executa a instrução instr do DOS no DOS |
| !prog.exe | executa no DOS os programas prog |
| !prog.com | executável .exe e .com |

CAPÍTULO 6

OTIMIZAÇÃO DO TEMPO DE EXECUÇÃO

Diante da riqueza das possibilidades oferecidas pelo *Mathematica*, impossível é citar todas as funções, as particularidades, os "atalhos" de programação e de linguagem acessíveis ao usuário. A consulta dos manuais, das telas de ajuda ("Help"), dos livros especializados deve permitir que cada um encontre o que lhe é necessário para a resolução do seu problema. Ninguém deve ter a pretensão de conhecer tudo, é impossível, e nem há motivo para isto. Cada um deve tentar conhecer e usar convenientemente as ferramentas mais adequadas para a resolução de seus problemas. O que é possível fazer, para ajudar a ir além do estágio de principiante, é indicar algumas funções (ou como encontrá-las) que podem permitir resolver os problemas do usuário e diminuir o tempo de desenvolvimento dos programas. Simplificá-los e facilitar sua compreensão (através de recursos visuais e de comentários) por outros usuários é também muito importante.

Deve-se também tentar diminuir o tempo de execução a fim de se realizar a execução dos programas dentro de tempos razoáveis, mesmo com microcomputadores de desempenho limitado. Para instruções que são executadas uma ou poucas vezes e que operam sobre poucos dados, o ganho de tempo de execução pela escolha das funções mais rápidas não será importante, em parte por causa do tempo (freqüentemente irreduzível) dedicado às entradas e saídas. O ganho sobre o tempo de execução é maior quando se trata da execução de loops com muitas repetições e/ou do tratamento de muitos dados. Os principais meios oferecidos para diminuir o tempo de execução são: uso de funções compiladas, de função com memória de valores já calculados e de instruções de loop mais velozes.

Um meio eficaz para diminuir o tempo de execução consiste em aumentar o uso da memória RAM disponível para os cálculos e diminuir o tempo de uso do disco rígido para a mesma função. Mas isto depende principalmente do microcomputador usado (memória RAM total, menos a que é dedicada a outros softwares residentes; memória livre no disco rígido) e depende apenas em parte do usuário (ajuste da memória virtual do Windows e da memória disponível para o kernel no *Mathematica*).

Emprego preferencial da função Table[] no lugar das demais funções de laço (loop). Para diminuir drasticamente os tempos de cálculo, usar, sempre que possível, as fórmulas que comportam embutidas as iterações (como Table[]), de preferência aos loops clássicos (Exemplo: For[início, condição, incrementação, {conteúdo}]; While[condição, incrementação, {conteúdo}]). Em certos casos, é possível converter loops clássicos em instruções que usam Table[], mesmo que a tabela gerada não seja útil: uma instrução pode começar por Table[...] e tudo que está dentro é executado, o objeto do enchimento da tabela podendo ser uma instrução ou seqüência de instruções, se for preciso com If[] que dependem dos parâmetros de linha e coluna. Isto funciona também na conversão de laços aninhados, onde foi verificado que Table[Table[]] serve também para encurtar o tempo de processamento.

Função com memória de valores já calculados ou seja Funções que se lembram de valores que elas já acharam (2.4.9 - Functions That Remember Values They Have Found, no livro de Wolfram, segunda edição; Saving, intermediate values). O *Mathematica* apresenta a possibilidade de conservar valores já calculados por uma função para os valores já usados da variável. Ao usar de novo o mesmo valor da variável na mesma função, o valor calculado anteriormente é usado. Isto pode representar uma economia no tempo de cálculo, mas, evidentemente apresenta o inconveniente de ocupar espaço de memória. O usuário deverá escolher o que é mais vantajoso para seus cálculos. A sintaxe deste comando é:

`f[x_] := f[x] = rhs`; O leitor poderá consultar o livro de Wolfram (1991), para maiores esclarecimentos.

Compilação de expressões no *Mathematica* (2.5.14 Compiling *Mathematica* Expressions, no livro de Wolfram, segunda edição; Compiled functions). Assumindo que uma variável numa expressão possa tomar apenas valores numéricos (ou variáveis lógicas), o *Mathematica* pode compilá-la, o que permite encurtar o tempo necessário para sua avaliação. A sintaxe do comando de definição de uma função não compilada é, por exemplo,

`f[x] = x Sin[x];` ou seja,
`f = Function[{x}, x Sin[x]];`

A sintaxe do comando de compilação é:

`fc = Compile[{x}, x Sin[x]];`

O leitor poderá consultar o livro de Wolfram, para maiores esclarecimentos.

Comparação da velocidade de execução de vários tipos de funções. Carregar e executar as diferentes células do notebook "Speeding". Foram testadas as instruções `f=Function[{x},função]`; `f[x_]:=função`; `f[x_]:=f[x]=função`; `f=Compile[{x},função]` com a execução repetida 500 vezes para o mesmo valor de $x=7$ da função $8 x \sin[0.5 x + 3] \exp[-0.5 x] + 2 x^4 + x^7$. Tempos variáveis dentro de um faixa estreita foram encontrados. Os valores médios dos tempos de execução foram respectivamente: 11.2 sec, 10.7 sec, 2.1 sec e 2.7 sec. no exemplo escolhido, o melhor tempo foi, evidentemente para a função com memória (por tratar-se do cálculo da função para o mesmo valor da variável).

CAPÍTULO 7

MATEMÁTICA SIMBÓLICA: Aplicação ao regime transitório de sistemas; Resposta a sinais envolvendo pulsos, degraus e periodicidades.

A possibilidade de o *Mathematica* poder efetuar cálculos de Computação Simbólica (Symbolic Computation), ou seja de Matemática Simbólica (Symbolic Mathematics), é aproveitada no tratamento desta aplicação. No Apêndice A, Lista de Notebooks, o leitor poderá consultar e experimentar, a título de exemplo, o Notebook "Respostas senóide-exponenciais em séries temporais geofísicas" [SINEXP.MA], com a possibilidade de modificá-lo, retrabalhá-lo, acrescentando nele suas próprias observações e conclusões.

`DSolve[y'[t] + K y[t] == B + A Sin[pu t] , y[t] , t]`

->

$y[t] = C[1] \text{Exp}[-K t] + (B/K + A \text{Cos}[pu t] + A \text{Sin}[pu t]) / (K^2 + pu^2)$

C[1] é determinado pela condição limite $y[0] = 0$ (por exemplo):

$y[0] = 0 \rightarrow C[1] = A pu / (K^2 + pu^2) - B / K$

$y[t] = (B / K) (1 - \text{Exp}[-K t]) + (A / (K^2 + pu^2)) (pu \text{Exp}[-K t] - pu \text{Cos}[pu t] + K \text{Sin}[pu t])$

Casos particulares:

$$K = \infty \text{ (ou } t \gg 1 / K \text{)}$$

$$y[t] = B / K + (A / (K^2 + pu^2)) (- pu \text{ Cos}[pu t] + K \text{ Sin}[pu t])$$

$$K = 0$$

$$y''[t] = B + A \text{ Sin}[pu t] \rightarrow y[t] = c[1] + B t - (A/pu) \text{ Cos}[pu t]$$

$$pu = 0 \text{ (ou } A = 0 \text{) (frequência baixa) } \rightarrow y[t] = (B/k) (1 - \text{Exp}[-K t])$$

$$pu = \infty \text{ (frequência alta) } \rightarrow y[t] = (B/k) (1 - \text{Exp}[-K t])$$

Regime permanente periódico:

$$y[t] = B / K + (A / (K^2 + pu^2)) (- pu \text{ Cos}[pu t] + K \text{ Sin}[pu t])$$

equivalente a: $y[t] = B / K + A \text{ Sin}[pu t - C]$

com: $\text{Sin}[C] = pu / (K^2 + pu^2)$

e $\text{Cos}[C] = K / (K^2 + pu^2)$

Observações:

- Não se pode usar $N[t]$ no lugar de $y[t]$ por causa do significado de $N[...]$ no *Mathematica*.

- Em alguns casos mais complicados, o *Mathematica* indica soluções com exponenciais de funções imaginárias para as funções senoidais, que são pouco legíveis e levariam ao uso da fórmula de Euler.

- Para ficar em conformidade com os símbolos do ambiente do *Mathematica*, recomenda-se, (pelo menos) quando trabalha-se (de maneira firme) neste ambiente, usar os símbolos usados nele, por exemplo Exp , Sin , Cos , Tan etc e a mesma "hierarquia" de parênteses, colchetes, chaves etc, $[]$ para funções, $\{ \}$ para domínios de valores, $()$ para agrupamento de termos etc. Observar (isto é conhecer e respeitar, evidentemente) também as diferenças de uso entre $,$ e $;$ entre o *Mathematica* e o C.

CAPÍTULO 8

ORIENTAÇÃO, OTIMIZAÇÃO E EXERCÍCIOS

Padrão de apresentação de notebook do *Mathematica* (para exercícios, provas de conclusão do curso, trabalhos pessoais etc.)

- Para guiar o principiante, é dado a seguir um roteiro de padronização de desenvolvimento e apresentação de notebook. O intuito principal deste roteiro é facilitar o funcionamento do algoritmo e sua compreensão pelos leitores ou usuários que não participaram de sua elaboração, assim como evitar a omissão de elementos importantes. O roteiro a seguir vale tanto para a apresentação impressa quanto para os arquivos nos quais o notebook é salvo (*.ma e *.mb). Um notebook deve comportar:
- O título (usar letras com fonte de tamanho superior ao do resto do notebook e/ou caracteres em negrito).
- Um resumo curtíssimo indicando a finalidade do notebook e a indicação das metodologias de destaque (pela originalidade, pelo desempenho, pela facilidade de desenvolvimento ou de operação etc.).
- Um resumo do roteiro (caso precisaria para esclarecer o resto).
- O corpo do algoritmo, com comentários, entre (* e *), em quantidade certa, nem demais, nem de menos. A finalidade dos comentários é ajudar o autor e os usuários a entender o processo, e facilitar assim seu uso e sua manutenção ou modificação.
- Os resultados, tais como eles aparecem na tela (gráficos, resultados numéricos etc.)

Outras dicas gerais, orientação:

- Reproduzir, digitando-os para treinamento no ambiente *Mathematica*, exemplos disponíveis ou parte deles. Modificá-los e ampliá-los vontade. Logo depois de familiarizar-se com o *Mathematica*, atacar partes de problemas mais complexos de seu interesse, para integrá-las posteriormente num trabalho mais amplo.
- Princípio geral: no caso do tratamento de um problema particular, definir claramente as tarefas a executar (montar um organograma é recomendado nos casos complicados), escolher as funções e informar-se sobre as instruções usadas pela primeira vez, montar seu programa, testá-lo elemento por elemento, ou por inteiro (se não funcionar da primeira vez, haverá necessidade de testá-lo instrução por instrução), fazer rodar o programa, verificar com dados artificiais ou mais simples se os resultados são certos, corrigir os problemas até se obter as condições desejadas de funcionamento.
- Observar a semelhança das etapas recomendadas para o desenvolvimento de um programa (por sinal, em qualquer linguagem) com as que são recomendadas dentro dos conceitos da Qualidade Total, no esquema PDCA (Plan-Do-Check-Action): planejamento (com definição clara da finalidade); execução (incluindo aprendizagem e treinamento); verificação (conformidade com o desejado) e ação corretiva (resolução dos problemas)
- lembrar-se que o importante é "fazer funcionar", isto é, ver o *Mathematica* funcionar num prazo curto, uma vez que o *Mathematica*, por ser uma linguagem de alto nível, permite desenvolver programas em seus notebooks em tempos muito mais curtos do que se fosse em outra linguagem (BASIC, FORTRAN, PASCAL, C, C++ etc.). Como para as demais linguagens de programação, sempre existem várias maneiras de programar para se chegar ao mesmo resultado a partir dos mesmos dados. Os esforços para otimização na duração da execução, concisão do programa, sua clareza e estética vêm

depois, isto é, depois de adquirir experiência e prática. Não parece exagerado dedicar um certo tempo para fazer funcionar (mais ou menos) um programa, e, em seguida, dedicar dez ou mais vezes este tempo para redimir possíveis erros, encurtar o tempo de processamento, tornar a lista mais legível (* acrescentando comentários *), isto é, dar o "polimento" final que ele precisa.

- Recomenda-se também analisar as justificativas para o uso do *Mathematica*. Com certeza existem casos para os quais usar o C, por exemplo, é muito mais viável, para se ter um tempo de processamento menor (casos com muitos dados e/ou muitas iterações, entre outros).

Exemplo de exercício:

- Construir uma série artificial dentre 20 a 50 dados (constante + senoidal; constante + exponencial; constante + dente de serra etc.), seja numa planilha ou num editor qualquer. Gravar a série num arquivo em ASCII (tempo e função em 2 colunas ou duas linhas, separando os dados por um espaço). Verificar em todos os casos para não haver linhas em branco no início do arquivo nem caracteres estranhos (isto é, que não seriam lidos convenientemente pelo *Mathematica*) no lugar dos espaços ou das mudanças de linha (Line feed).
- Montar um notebook no *Mathematica* para ler o arquivo gerado na parte 1., traçar o gráfico da série, fazer o ajuste pela constante e pela função da série visando determinar os parâmetros usados. Gravar os resultados do ajuste num arquivo ASCII.
- Recomeçar o mesmo exercício, acrescentando a cada valor da série um ruído de no máximo 20% do valor. Como complemento, gerar um gráfico onde aparecerão a série original e a série reconstituída a partir dos parâmetros obtidos no ajuste.

CAPÍTULO 9

LISTA DOS OBJETOS DO *MATHEMATICA* 2.21

(obtida a partir da parte "Listing of Built-in *Mathematica* Objects" do livro de Wolfram, 2a Edição, 1991, das páginas 749 até 905; Os nomes do vocabulário do *Mathematica* são apresentados em negrito para facilitar a leitura, e são seguidos da descrição sumária da função em português)

Abort[]

Pára um cálculo.

AbortProtect[]

Avalia seu argumento, sendo os eventuais Abort adiados para depois do fim da avaliação.

Abs[]

Valor absoluto de um número real ou complexo.

AbsoluteDashing[{d₁, d₂, d₃}

Pontilha as linhas que seguem com segmentos do comprimento indicado, em seqüência (expressos em pontos, isto é, em 1/72 de polegada, ou seja, 0.35 mm aproximadamente).

AbsolutePointSize[r]

Faz com que os pontos sejam representados nos gráficos por regiões circulares de raio r (expresso em pontos, isto é, em 1/72 de polegada, ou seja, 0.35 mm aproximadamente).

AbsoluteThickness[d]

Faz com que as linhas sejam representadas nos gráficos com uma espessura d (expressa em pontos, isto é, em $1/72$ de polegada, ou seja, 0.35 mm aproximadamente).

AbsoluteTime[]

Número de segundos desde o início do dia 1^o janeiro de 1900.

AccountingForm[expr]

Mostra os números dentro de $expr$ em notação padrão da contabilidade.

Accuracy[x]

Indica o número de dígitos após o ponto decimal (equivalente à nossa vírgula decimal) no número x .

AccuracyGoal

Opção para indicar quantos dígitos de precisão devem ter os resultados.

AddTo[x, dx]

$x += dx$ equivalente a $x = x + dx$.

AiryAi[x]

Função de Airy $Ai(x)$.

AiryAiPrime

Derivada da função de Airy $Ai(x)$.

AiryBi

Função de Airy $Bi(x)$.

AiryBiPrime

Derivada da função de Airy $Bi(x)$.

AlgebraicRules[eqns, {x₁, x₂,...}]

Gera um conjunto de regras algébricas para substituição das variáveis da lista dos x_j , segundo a ou as equações $eqns$.

All

Todos (É usado em certas opções).

Alternatives

Objeto representado por várias alternativas.

AmbientLight

Opção em Graphics3D para simular iluminação ambiente de uma imagem tridimensional.

AmbientLight->GrayLevel, **AmbientLight->Hue**,
AmbientLight->RGBColor.

AnchoredSearch

Opção de **Find** e **FindList** para especificar se o texto procurado deve estar no início de um segmento.

And

a && b && ..., isto é, função lógica AND. Devolve False se qualquer um dos argumentos a, b, ... é False; devolve True se todos são True.

Apart[expr]

Rescreve a expressão **expr** como soma de termos com denominadores mínimos.

Append[expr, elem]

Devolve uma expressão que contém **expr** acrescentada com **elem**.

AppendTo[s, elem]

Equivalente a **s = Append[s, elem]**, isto é, devolve a expressão **s** constituída pela antiga expressão **s** acrescentada com **elem**.

Apply[f, expr]

f @@ expr substitui o cabeçalho de **expr** por **f**.

ArcCos[z]

arco cosseno, $\arccos(z)$ ou $\cos^{-1}(z)$, do número complexo **z**. Os resultados são dados em radianos.

ArcCosh[z]

arco cosseno hiperbólico, $\operatorname{arccosh}(z)$ ou $\cosh^{-1}(z)$, do número complexo z .

ArcCot[z]

arco co-tangente, $\operatorname{arccotg}(z)$ ou $\cot^{-1}(z)$, do número complexo z . Os resultados são dados em radianos.

ArcCoth[z]

arco co-tangente hiperbólica, $\operatorname{arccotgh}(z)$ ou $\coth^{-1}(z)$, do número complexo z .

ArcCsc[z]

arco co-secante, $\operatorname{arccosec}(z)$ ou $\csc^{-1}(z)$, do número complexo z .

ArcCsch[z]

arco co-secante hiperbólica, $\operatorname{arccosech}(z)$ ou $\operatorname{csch}^{-1}(z)$, do número complexo z .

ArcSec[z]

arco secante, $\operatorname{arcsec}(z)$ ou $\sec^{-1}(z)$, do número complexo z .

ArcSech[z]

arco secante hiperbólica, $\operatorname{arcsech}(z)$ ou $\operatorname{sech}^{-1}(z)$, do número complexo z .

ArcSin[z]

arco seno, $\operatorname{arcsen}(z)$ ou $\sin^{-1}(z)$, do número complexo z .

ArcSinh[z]

arco seno hiperbólico, $\operatorname{arcsenh}(z)$ ou $\sinh^{-1}(z)$, do número complexo z .

ArcTan[z]

arco tangente, $\operatorname{arctg}(z)$ ou $\tan^{-1}(z)$, do número complexo z .

ArcTanh[z]

arco tangente hiperbólica, $\text{arctgh}(z)$ ou $\tanh^{-1}(z)$, do número complexo z .

Arg[z]

Argumento do número complexo z .

ArithmeticGeometricMean[a, b]

Média aritmética-geométrica de a e b .

Array[f, n]

Lista de comprimento n com os elementos $f[i]$...

AspectRatio

Especifica em `Show` a relação altura/largura de um gráfico;

Exemplos: **AspectRatio->0.8**,

AspectRatio->Automatic.

AtomQ[expr]

Devolve **True** se a expressão expr não pode ser subdividida em subexpressões e **False** no caso contrário.

Attributes[symbol]

Lista dos atributos do símbolo symbol .

Automatic

Opção para a determinação automática das opções.

Axes

Opção para as funções gráficas que especifica se os eixos devem aparecer.

Axes->True desenha todos os eixos nos gráficos,

Axes->False não desenha os eixos

Axes->{False, True} desenha o eixo y sem desenhar o eixo x .

AxesEdge

Opção para indicar que limites devem ser desenhados nas caixas dos gráficos tridimensionais.

AxesLabel

Opção para se ter ou não legendas nos eixos dos gráficos.

Exemplos: **AxesLabel->None**

AxesLabel->{labelx, labely, labelz}

AxesOrigin->{x, y}

Opção para gráficos de duas dimensões que indica o ponto de cruzamento dos eixos.

AxesStyle

Opção para gráficos que indica o estilo para os eixos.

Background

Opção para gráficos que indica a cor de fundo a ser usada.

Background->CMYKColor

Background->RGBColor

Background->Hue

Background->GrayLevel

BaseForm[expr, n]

Mostra os números de expr usando a base n.

Begin["context"]

Restabelece o contexto "context" habitual.

BeginPackage["context"]

Ativa o contexto "context".

BernoulliB[n]

Número de Bernoulli B_n .

BernoulliB[n, x]

Polinômio de Bernoulli $B_n(x)$.

BesselI[n, z]

Função de Bessel de primeira ordem $I_n(z)$.

BesselJ[n, z]

Função de Bessel de primeira ordem $J_n(z)$.

BesselK[n, z]

Função de Bessel de segunda ordem modificada $K_n(z)$.

BesselY[n, z]

Função de Bessel de segunda ordem $Y_n(z)$.

Beta[a, b]

Função beta de Euler $B(a, b)$.

Beta[z, a, b]

Função beta incompleta de Euler $B_z(a, b)$.

BetaRegularized[z, a, b]

Função beta regularizada incompleta $I_z(a, b)$.

Binomial[n, m]

Coeficiente do binómio ($\binom{n}{m}$).

Blank[]

`_` ou **Blank[]** é um objeto que pode tomar o lugar de qualquer expressão do *Mathematica*.

`_h` ou **Blank[h]** pode tomar o lugar de qualquer expressão que começa por h.

BlankNullSequence[]

`___` (três espaços sublinhados consecutivos) ou

BlankNullSequence[] é um objeto que pode tomar o lugar de nenhuma, uma ou algumas expressões do *Mathematica*.

BlankSequence[]

`__` (dois espaços sublinhados consecutivos) ou **BlankSequence[]** é um objeto que pode tomar o lugar de uma ou mais expressões do *Mathematica*.

Block[{x, y, ...}, expr]

Especifica que expr deve ser calculada com os valores locais dos símbolos x, y, ...

Boxed

Opção para Graphics3D que especifica se deve ou não aparecer a caixa dentro da qual o gráfico encontra-se: **Boxed->True** desenha a caixa, **Boxed->False** não desenha a caixa.

BoxRatios

Opção para Graphics3D e SurfaceGraphics que especifica os comprimentos dos lados da caixa dentro da qual o gráfico encontra-se: **BoxRatios->**{s_x, s_y, s_z}.

BoxStyle

Opção para Graphics3D que especifica como deve aparecer a caixa dentro da qual o gráfico encontra-se.

Break[]

Sai do mais próximo laço (loop) Do, For ou While

Byte

Representa um único byte de dado na execução de Read.

ByteCount[expr]

Número de bytes usados internamente pelo *Mathematica* para armazenar a expressão expr.

C[i]

Formato padrão da i-ésima constante de integração na resolução de uma equação diferencial por DSolve.

Cancel[expr]

Elimina os fatores comuns no numerador e no denominador da expressão expr.

Cases[{e₁, e₂, ...}, pattern]

Lista dos e_i que seguem o formato pattern.

Catalan

Constante de Catalan aproximadamente igual a 0.915966.

Catch[expr]

Retorna o argumento do primeiro **Throw** gerado na avaliação de `expr`.

Ceiling[x]

Menor inteiro que é maior ou igual a `x`.

CForm[expr]

Mostra a expressão `expr` em linguagem C.

Character

Representa um único caracter na execução de `Read`.

Characters["string"]

Lista dos caracteres da seqüência de caracteres `string`.

ChebyshevT[n, x]

Polinômio de Chebyshev de primeira ordem $T_n(x)$

ChebyshevU[n, x]

Polinômio de Chebyshev de segunda ordem $U_n(x)$.

Check[expr, falha]

Avalia a expressão `expr` e retorna o resultado caso não houver mensagem emitida; se houver, retorna `falha`.

CheckAbort[expr, failexpr]

Avalia a expressão `expr` e retorna `failexpr` caso houver parada (**Abort**) na execução.

Chop[expr]

Substitui na expressão `expr` números reais aproximados vizinhos de zero pelo inteiro exato 0.

Chop[expr, delta]

Substitui na expressão expr os números reais de amplitude menor que delta por 0.

Circle[{x, y}, r]

Primitiva gráfica bidimensional que representa um círculo de raio r centrado no ponto de coordenadas x e y.

Clear[symb₁, symb₂,...]

Apaga valores e definições para os símbolos symb_i. É equivalente a:

symb_i =. (sinal igual seguido de um ponto)

Clear["form₁", "form₂",...]

Apaga valores e definições para todos os símbolos cujos nomes correspondem as seqüências "form_i".

ClearAll[symb₁, symb₂,...]

Apaga todos os valores, definições, atributos, mensagens etc associados aos símbolos symb_i.

ClearAll["form₁", "form₂",...]

Apaga todos os valores, definições, atributos, mensagens etc. para todos os símbolos cujos nomes correspondem a qualquer uma das seqüências "form_i".

ClearAttributes[s, attr]

Remove o atributo attr da lista dos atributos do símbolo s.

ClebschGordan[{j₁, m₁}, {j₂, m₂}, {j, m}]

Fornece o coeficiente de Clebsch-Gordan para a decomposição de $|j, m\rangle$ em termos de $|j_1, m_1\rangle |j_2, m_2\rangle$.

ClipFill

Opção para **SurfaceGraphics** que especifica quais são as partes recortadas da superfície que devem ser desenhadas.

Close[stream]

Fecha o fluxo stream.

CMYKColor[c, m, y, k]

Diretriz gráfica que indica a cor com a qual devem ser representados os objetos que seguem, em termos da mistura com os valores c, m, y, k (cada um entre 0 e 1) para as cores ciã (azul), magenta (carmesim/roxo), amarelo e preto, respectivamente.

Coefficient[expr, form]

Fornece os coeficientes de form na expressão expr.

Coefficient[expr, form, n]

Fornece os coeficientes de form^n (isto é, form^n) na expressão expr.

CoefficientList[poly, var]

Fornece a lista dos coeficientes das potências de var no polinômio poly.

Collect[expr, x]

Junta os termos de expr que têm a mesma potência de x.

ColorFunction

Opção para funções gráficas que indica a cor, em função dos valores de z, que deve ser usada para uma região x, y dada

ColorFunction->CMYKColor

ColorFunction->RGBColor

ColorFunction->Hue

ColorFunction->GrayLevel

ColorOutput

Opção para funções gráficas que permite especificar o tipo de cor produzida na saída.

ColumnForm[{e₁, e₂,...}]

Imprime em uma coluna, com e₁ acima de e₂ etc.

ColumnForm[list, horiz] especifica o alinhamento horizontal de cada elemento.

ColumnForm[list, horiz, vert] especifica também o alinhamento vertical da coluna inteira.

O alinhamento horizontal pode ser:

Center centrado
Left justificado a esquerda (default)
Right justificado a direita

O alinhamento vertical pode ser:

Above o elemento de baixo da coluna é alinhado na base
Below o elemento de cima é alinhado na base (default)
Center a coluna é centrada na base

Compile[{x₁, x₂,...}, expr]

Cria uma função compilada que avalia expr assumindo valores numéricos para os x_i.

Compiled

Opção para várias funções numéricas ou gráficas que especifica que suas expressões devem ser automaticamente compiladas.

Compiled->True cria automaticamente funções compiladas.

Compiled->False deve ser utilizado caso for preciso utilizar números com precisão elevada.

CompiledFunction[args, nregs, instr, func]

Representa um código compilado para avaliar uma função func compilada; args é uma lista que contém os formatos de cada tipo de argumento da função func; nregs é uma lista que indica o número de registros inteiros, reais, complexos e lógicos necessários para avaliar o código compilado; instr é a lista das próprias instruções compiladas.

Complement[list, e₁, e₂, ...]

Elementos da tabela list que não estão em qualquer um dos e_i.

Complex

Cabeçalho usado para números complexos ($x + I y$).

ComplexExpand[expr]

Expanda a expressão *expr* supondo que todas suas variáveis são reais.

ComplexExpand[expr, {x₁, x₂,...}]

Expanda a expressão *expr* supondo que as variáveis de nomes iguais aos x_1, x_2, \dots são complexas.

ComplexInfinity

Representa uma quantidade de amplitude infinita e de fase complexa indefinida.

ComposeList[{f₁, f₂,...},x]

Gera uma lista do tipo $\{x, f_1[x], f_2[f_1[x]], \dots\}$

Composition[f₁, f₂, f₃,...]

Representa a composição das funções f_1, f_2, f_3, \dots

Exemplo:

Composition[a, b, c][x] fornece **a[b[c[x]]]**

CompoundExpression

*expr*₁; *expr*₂; ...; *expr*_j; ...; *expr*_j avalia as expressões *expr*_j e fornece apenas o resultado da última *expr*_j.

Condition

expr /; test executa *expr* apenas se o resultado de *test* é **True**.

nome := regra /; test representa uma regra aplicada apenas se o resultado de *test* é **True**.

nome := expr /; test é uma definição válida apenas se o resultado de *test* é **True**.

Conjugate[z]

Conjugado complexo z^* do número complexo *z*.

Constant

Atributo que indica que um símbolo possui uma derivada nula em relação a todos os parâmetros.

Constants

Opção para **Dt** que indica uma lista de objetos que devem ser tomados como constantes.

ConstrainedMax[f, {inequalities}, {x, y,...}]

Acha o máximo global de f dentro do domínio especificado pelas desigualdades $inequalities$. Todas as variáveis x, y, \dots são supostamente não negativas.

ConstrainedMin[f, {inequalities}, {x, y,...}]

Acha o mínimo global de f dentro do domínio especificado pelas desigualdades $inequalities$. Todas as variáveis x, y, \dots são supostamente não negativas.

Context[]

Contexto atual (estado das opções).

Context[symb]

Contexto dentro do qual o símbolo $symb$ encontra-se.

Contexts[]

Fornece a lista de todos os contextos.

Contexts["string"]

Fornece a lista de todos os contextos que combinam com a seqüência "string".

ContextToFileName["context"]

Fornece a seqüência que especifica o nome de arquivo que é, por convenção, associado a um contexto particular.

Continuation[n]

É apresentado no início da n -ésima linha de uma expressão de várias linhas.

Continue[]

Sai para o mais próximo laço (loop) **Do**, **For** ou **While** (isto é, deixe de executar o resto da procedura do laço, entre a posição de **Continue[]** e o fim do laço).

ContourGraphics[array]

Representação de um mapa de contorno dos dados da tabela retangular (array) que contem os valores de z (linhas x e colunas y). Pode comportar as seguintes opções:

| | | |
|-------------------------|--------------------------|--------------------------------------|
| AspectRatio | 1 | relação altura/largura |
| Axes | False | desenhar eixos |
| AxesLabel | None | legenda nos eixos (nenhuma: none) |
| AxesOrigin | Automatic | ponto de cruzamento dos eixos |
| AxesStyle | { } | estilo para os eixos |
| BackGround | Automatic | cor de fundo |
| ColorFunction | Automatic | cor entre linhas de contorno |
| ColorOutput | Automatic | saída colorida para gerar |
| ContourLines | True | linhas de contorno (sim ou não) |
| Contours | 10 | contornos para utilizar |
| ContourShading | True | sombras entre contornos |
| ContourSmoothing | None | alisamento (linhas de contorno) |
| ContourStyle | Automatic | estilo (linhas de contorno) |
| DefaultColor | Automatic | cor padrão para gráfico |
| DefaultFont | { } | fonte padrão para texto |
| DisplayFunction | \$DisplayFunction | função para gerar saída |
| Epilog | { } | primitivas gráficas após gráfico |
| Frame | True | moldura (sim ou não) |
| FrameLabel | None | legenda na moldura |
| FrameStyle | Automatic | estilo da moldura |
| FrameTicks | Automatic | marcas na moldura |
| MeshRange | Automatic | domínios para x e y |
| PlotLabel | None | legenda para o gráfico |
| PlotRange | Automatic | domínio das valores de z |
| PlotRegion | Automatic | região que o gráfico ocupará |
| Prolog | { } | primitivas gráf. antes do gráfico |
| RotateLabel | True | girar a legenda do eixo y |
| Ticks | Automatic | marcas nos eixos |

ContourLines

Opção de mapa de contorno para desenhar linhas de contorno explícitas (**ContourLines**->**True**) ou não (**ContourLines**->**False**).

ContourPlot[f, {x, xmin, xmax}, {y, ymin, ymax}]

Gera um mapa de contorno de f em função de x e y.

Contours

Opção de **ContourGraphics** que especifica os contornos para usar.

ContourShading

Opção de **ContourGraphics** que especifica se as regiões entre as linhas de contorno devem ser sombreadas.

ContourSmoothing

Opção de **ContourGraphics** que especifica como as linhas de contorno devem ser alisadas.

ContourStyle

Opção de **ContourGraphics** que especifica o estilo com o qual as linhas de contorno devem ser desenhadas.

CopyDirectory["dir₁", "dir₂"]

Copia o diretório dir₁ para o diretório dir₂. Apenas o diretório dir₁ já deve existir.

CopyFile["file₁", "file₂"]

Copia o arquivo file₁ para o arquivo file₂.

Cos[z]

Cosseno, $\cos(z)$, do número complexo z.

Cosh[z]

Cosseno hiperbólico, $\cosh(z)$, do número complexo z.

CosIntegral[z]

Cosseno integral, $\text{Ci}(z)$, do número complexo z.

Cot[z]

Co-tangente, $\cotg(z)$ ou $\cotan(z)$, do número complexo z .

Coth[z]

Co-tangente hiperbólica, $\cotgh(z)$ ou $\cotanh(z)$, do número complexo z .

Count[list, format]

Número de elementos da lista `list` que têm o formato `format`.

CreateDirectory["dir"]

Cria o diretório `dir`.

Csc[z]

Co-secante, $\operatorname{cosec}(z)$, do número complexo z .

Csch[z]

Co-secante hiperbólica, $\operatorname{cosech}(z)$, do número complexo z .

Cubics

Opção para **Roots** e funções afins que especifica se as soluções devem ser geradas para equações cúbicas irredutíveis.

Cuboid[{xmin, ymin, zmin}]

Primitiva gráfica tridimensional que representa um cubóide unitário orientado paralelo aos eixos.

Cuboid[{xmin, ymin, zmin}, {xmax, ymax, zmax}] é um cubóide especificado pelas coordenadas dos cantos opostos.

Cyclotomic[n, x]

Fornece o polinômio ciclotímico de ordem n de x .

D[f, x]

Derivada parcial de f em relação a x .

$$\frac{\partial}{\partial x} f$$

D[f, x₁, x₂, ...]

Derivada múltipla de f em relação a x.

$$\frac{\partial^n}{\partial x_n} f$$

D[f, x]

Derivada múltipla de f em relação a x.

$$\frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots f$$

Dashing[{r₁, r₂, ...}]

Pontilha as linhas que seguem com segmentos com os comprimentos indicados r₁, r₂, ... (expressos em frações da largura total do gráfico) em seqüência.

Date[]

Data e hora no formato {ano, mês, dia, hora, minuto, segundo}.

DeclarePackage["context", {"name₁", name₂", ...}]

Declara que **Needs["context"]** deve automaticamente ser executado se o símbolo de qualquer um dos nomes indicado é usado.

Decompose[poly, x]

Decompõe o polinômio poly numa composição de polinômios mais simples, se for possível.

Decrement

x-- utiliza o valor de x e, em seguida, diminui este valor de x de 1.

Default[f]

Fornece, se definido, o valor padrão dos argumentos da função f obtida com um objeto de formato . (espaço sublinhado seguido de um ponto).

Default[f, i]

Fornece o valor padrão para usar quando _. (espaço sublinhado seguido de um ponto) aparece como o i-ésimo argumento de f.

Default[f, i, n]

Fornece o valor padrão para o i-ésimo argumento de um total de n argumentos.

DefaultColor

Opção para as funções gráficas que especifica a cor padrão a ser usada para linhas, pontos etc.

DefaultFont

Opção para as funções gráficas que especifica a fonte padrão a ser usada para o texto.

Definition[s₁, s₂,...]

Mostra as definições dadas aos símbolos s_i.

Degree

Fração de radiano em um grau, ou seja $\pi/180$.

Delete[expr, n]

Apaga o elemento que ocupa a posição n na expressão expr. Se n for negativo, a posição é contada a partir do fim.

Delete[expr, {i, j, ...}]

Apaga o elemento que ocupa a posição {i, j, ...} na matriz expr.

Delete[expr, {{i₁, j₁, ...}, {i₂, j₂, ...}, ...}]

Apaga as partes que ocupam as posições {i₁, j₁, ...}, {i₂, j₂, ...}, ... na matriz expr.

DeleteCases[expr, pattern]

Apaga todos os elementos de expr que tem o formato pattern.

DeleteCases[expr, pattern, levspec]

Apaga todos as partes de expr que têm o formato pattern nos níveis indicados por levspec.

DeleteDirectory["dir"]

Apaga o diretório especificado dir.

DeleteFile["file"]

Apaga o arquivo file.

DeleteFile[{"file₁", "file₂", ...}]

Apaga os arquivos file_i que constam na lista indicada.

Denominator[expr]

Fornece o denominador da expressão expr.

DensityGraphics[array]

Representa um mapa de densidade da tabela retangular array que contem os valores de z. As opções que seguem podem ser empregadas:

| | | |
|-----------------|-------------------|--|
| AspectRatio | 1 | relação altura/largura |
| Axes | False | desenhar eixos (sim: True / não: False) |
| AxesLabel | None | legenda nos eixos (nenhuma: none) |
| AxesOrigin | Automatic | ponto de cruzamento dos eixos |
| AxesStyle | {} | estilo para os eixos |
| BackGround | Automatic | cor de fundo |
| ColorFunction | Automatic | cor entre linhas de contorno |
| ColorOutput | Automatic | saída colorida para gerar |
| DefaultColor | Automatic | cor padrão para gráfico |
| DefaultFont | \$DefaultFont | fonte padrão para texto |
| DisplayFunction | \$DisplayFunction | função para gerar saída |
| Epilog | {} | primitivas gráficas após gráfico |
| Frame | True | moldura (sim ou não) |
| FrameLabel | None | legenda na moldura |
| FrameStyle | Automatic | estilo da moldura |
| FrameTicks | Automatic | marcas na moldura |
| Mesh | True | desenhar uma malha ou não |

DensityGraphics[array] (Continuação)

| | | |
|-------------|-----------|-----------------------------------|
| MeshRange | Automatic | domínios para x e y |
| MeshStyle | Automatic | estilo para as linhas de malha |
| PlotLabel | None | legenda para o gráfico |
| PlotRange | Automatic | domínio das valores de z |
| PlotRegion | Automatic | região que o gráfico ocupará |
| Prolog | { } | primitivas gráf. antes do gráfico |
| RotateLabel | True | girar a legenda do eixo y |
| Ticks | Automatic | marcas nos eixos |

DensityPlot[f, {x, xmin, xmax}, {y, ymin, ymax}]

Cria um mapa de densidade de f em função de x e y dentro dos limites indicados.

Depth[expr]

Número máximo de índices necessários para especificar qualquer elemento de expr, mais um. Um objeto simples tem uma profundidade (depth) de 1.

Derivative[1][f]

f' é a derivada da função f de uma variável.

Derivative[2][f]

f'' é a derivada segunda da função f de uma variável.

Derivative[n₁, n₂,...][f]

Função obtida a partir de f, derivando-a, n₁ vezes em relação à primeira variável, n₂ vezes em relação à segunda variável etc.

Det[m]

Determinante da matriz quadrada m.

DiagonalMatrix[list]

Matriz cujos elementos na diagonal são os elementos da lista list e os demais valem 0.

Dialog[]

Começa um diálogo (janelinha de comunicação).

Dialog[expr]

Começa um diálogo (janelinha de comunicação) com a expressão expr como valor corrente de %.

DialogIndent[d]

Imprime um deslocamento para direita de largura d para as linhas de entrada e saída do diálogo.

DialogProlog

Opção para **Dialog** que pode indicar uma expressão para ser avaliada antes do início do diálogo.

DialogSymbols

Opção para **Dialog** que indica uma lista de símbolos cujos valores devem ser localizados no diálogo.

DigitBlock

Opção para **NumberForm** e as funções afins que especifica o comprimento máximo dos blocos de dígitos entre as rupturas (breaks).

Exemplos: **DigitBlock->Infinity**, **DigitBlock->n**.

DigitQ[string]

Fornece **True** se todos os caracteres da corrente string são dígitos entre 0 e 9 e fornece **False** no caso contrário.

Dimensions[expr]

Fornece a lista das dimensões de expr.

Dimensions[expr, n]

Fornece a lista das dimensões de expr até o nível n.

DirectInfinity[]

Quantidade numérica infinita de direção desconhecida no plano complexo.

DirectInfinity[z]

Quantidade numérica infinita que é um múltiplo (multiplicando por um número real positivo) do número complexo z.

Directory[]

Indica o diretório ativo.

DirectoryStack

Fornece a pilha da seqüência dos directórios ativos.

Disk[{x, y},r]

Primitiva gráfica a 2 dimensões que representa um disco cheio de raio r, centrado no ponto x, y.

Dispatch[{lhs₁->rhs₁, lhs₂->rhs₂, ...}]

Gera o equivalente de uma tabela ou lista de regras (substituição de regras, símbolos etc...) visando a otimização do tempo de execução.

Display[channel, graphics]

Escreve um gráfico ou manda um som no canal de saída indicado.

DisplayFunction

Opção para funções gráficas e de som que especifica a função a ser aplicada a objetos gráficos e de som antes de mostrá-los.

Exemplos:

DisplayFunction->Display[channel, #]&,

DisplayFunction->Identity

Distribute[f(x₁, x₂,...)]

Distribui f em relação aos Plus que aparecem em cada um dos x_i.

Exemplo: Distribute[f[a+b, c+d]] = f[a, c] + f[a, d] + f[b, c] + f[b, d]

Divide[x, y]

x/y ou **Divide[x, y]** é equivalente a $x y^{-1}$, isto é, $x y^{-1}$.

DivideBy

x /= c é equivalente a **x = x/c**.

Divisors[n]

Fornecer a lista dos inteiros que dividem n .

DivisorSigma[k, n]

Função divisor $\sigma_k(n)$.

Do[expr, {imax}]

Avalia $imax$ vezes a expressão $expr$.

Dot[a, b, c]

$a \cdot b \cdot c$ é o produto (interno) de vectores, matrizes ou tensores.

DownValues[f]

Fornecer a lista das regras de transformação que corresponde a todos os valores definidos pelo símbolo f .

Drop[list, n]

Lista $list$ da qual foram retirados os n primeiros elementos.

Drop[list, -n]

Lista $list$ da qual foram retirados os n últimos elementos.

Drop[list, {n}]

Lista $list$ da qual foi retirado o n -ésimo elemento.

Drop[list, {m,n}]

Lista $list$ da qual foram retirados os elementos do m -ésimo ao n -ésimo.

DSolve[eqn, y, x]

Resolve a equação diferencial eqn para a função y da variável independente x .

DSolve[{eqn1, eqn2,...}, {y1, y2,...}, x]

Resolve um sistema de equações diferenciais eq_1, eq_2, \dots para as funções y_1, y_2, \dots da variável independente x .

Dt[f]

Diferencial total df.

Dt[f, x]

Derivada total

$$\frac{d}{dx} f$$

Dt[f, {x, n}]

Derivada múltipla de ordem n de f em função de x

$$\frac{d^n}{dx^n} f$$

Dump["filename"]

Grava a imagem completa da sessão de *Mathematica* em andamento num arquivo "filename".

E

Base dos logaritmos naturais $e = 2.71828$.

EdgeForm[g]

Primitiva gráfica tridimensional que especifica as beiras dos polígonos a serem desenhados usando a diretriz gráfica ou a lista de diretrizes gráficas g.

Eigensystem[m]

Fornece a lista {valores, vectores} dos autovalores e dos autovectores da matriz quadrada m.

Eigenvalues[m]

Lista {valores} dos autovalores da matriz quadrada m.

Eigenvectors[m]

Lista {vectores} dos autovectores da matriz quadrada m.

Eliminate[eqns, vars]

Elimina as variáveis vars (uma ou uma lista de variáveis) do conjunto (lista) de equações simultâneas eqns.

EllipticE[m]

Integral elíptica completa $E(m)$.

EllipticE[phi, m]

Integral elíptica de segunda espécie $E(\phi|m)$.

EllipticExp[u, {a, b}]

Exponencial generalizada associada à curva elíptica $y^2 = x^3 + a x^2 + b x$.

EllipticF[phi, m]

Integral elíptica de primeira espécie $F(\phi|m)$.

EllipticK[m]

Integral elíptica completa de primeira espécie $K(m)$.

EllipticLog[{x, y}, {a, b}]

Logaritmo generalizado associado à curva elíptica $y^2 = x^3 + a x^2 + b x$.

EllipticPi[n, m]

Integral elíptica completa de terceira espécie $\Pi(n|m)$.

EllipticPi[n, phi, m]

Integral elíptica incompleta $\Pi(n, \phi|m)$.

EllipticTheta[a, u, q]

Função elíptica theta $\theta_a(u|q)$ ($a=1, \dots, 4$).

Encode["source", "dest"]

Escreve uma versão codificada do arquivo source para o arquivo dest.

<<dest decodifica o arquivo antes de ler seu conteúdo.

End[]

Devolve o contexto atual e volta ao anterior.

EndOfFile

Símbolo devolvido por Read ao atingir o fim de um arquivo.

EndPackage[]

Restora **\$Context** e **\$ContextPath** aos seus valores iniciais antes do **BeginPackage** anterior e acrescenta o contexto atual à lista **\$ContextPath**.

EngineeringForm[expr]

Mostra todos os números reais de expr usando a notação de engenharia.

Environment["var"]

Fornece o valor de uma variável de ambiente do sistema operacional.

Epilog

Opções para as funções gráficas que fornece a lista das primitivas gráficas para montar depois de a parte principal do gráfico ter sido montada.

Equal

lhs == rhs (com dois sinais = consecutivos) retorna True se lhs e rhs são iguais

Erf[z]

Função de erro erf(z).

Erf[z₀, z₁]

Função de erro generalizada erf(z₁) - erf(z₀).

Erfc[z]

Função de erro complementar erfc(z).

EulerE[n]

Número de Euler E_n.

EulerE[n, x]

Polinômio de Euler $E_n(x)$.

EulerGamma

Constante γ de Euler, $\gamma = 0.577216$.

EulerPhi[n]

Função de Euler $\phi(n)$, isto é, o número de inteiros positivos inferiores a n e primos em relação a n .

Evaluate[expr]

Avalia *expr*, mesmo que aparecendo como argumento de uma função cujo atributo único que não deve ser avaliada.

EvenQ[expr]

Fornece True se *expr* é um número inteiro par e False no caso contrário.

Exit[]

Termina uma sessão do *Mathematica*.

Exp[z]

Função exponencial $\exp(z)$ ou e^z . **Exp[z]** é convertida em E^z .

Expand[expr]

Expande os produtos e as potências positivas de *expr*.

ExpandAll[expr]

Expande os produtos e as potências positivas de todas os elementos de *expr*.

ExpandDenominator[expr]

Expande os produtos e as potências positivas que aparecem como denominadores de *expr*.

ExpandNumerator[expr]

Expande os produtos e as potências positivas que aparecem no numerador de *expr*.

ExpIntegralE[n, z]

Função exponencial integral $E_n(z)$.

ExpIntegralEi[z]

Função exponencial integral $Ei(z)$.

Exponent[expr, f]

Potência máxima com a qual f aparece em $expr$.

ExponentFunction

Opção para **NumberForm** e funções afins que determina o expoente a ser usado na impressão dos números reais aproximados.

Expression

Símbolo que representa uma expressão comum em **Read** e funções afins.

ExtendedGCD[n, m]

Maior divisor comum estendido dos números inteiros n e m .

FaceForm[gf, gb]

Diretriz gráfica tridimensional que especifica que as faces da frente dos polígonos devem ser desenhadas com a primitiva gráfica gf , e as de trás, com a primitiva gráfica gb . As especificações gf e gb devem ser diretrizes **CMYKColor**, **GrayLevel**, **Hue** ou **RGBColor**.

FaceGrid

Opção para gráficos tridimensionais que especifica as linhas de grade para desenhar nas faces da caixa na qual fica o gráfico.

Exemplos:

FaceGrid->**None**,

FaceGrid->**All**,

FaceGrid->**{face₁, face₂}**

Factor[poly]

Fatoriza um polinômio em relação a seus inteiros.

FactorComplete

Opção para **FactorInteger** que indica se uma fatoração completa deve ser realizada.

Exemplos:

FactorComplete->**True**

FactorComplete->**False**

Factorial[n]

Fornece o fatorial de n ou seja $n!$.

Factorial2[n]

Fornece o fatorial duplo de n ou seja $n!!$

FactorInteger[n]

Lista de fatores primos do inteiro n juntos com seus expoentes.

FactorList[poly]

Fornece a lista dos fatores de um polinômio e de seus expoentes.

FactorSquareFree[poly]

Omite qualquer fator quadrado no polinômio poly.

FactorSquareFreeList

Fornece a lista dos fatores não quadrados de um polinômio e de seus expoentes.

FactorTerms[poly]

Elimina do polinômio poly os fatores numéricos.

FactorTerms[poly, x]

Elimina do polinômio poly os fatores numérico que não dependem de x.

FactorTerms[poly, {x₁, x₂,...}]

Elimina do polinômio poly os fatores numérico que não dependem de cada um dos x_i .

FactorTermsList[poly, {x₁, x₂,...}]

Fornece uma lista ordenada dos fatores do polinômio poly.

False

Símbolo do valor booleano false (falso).

FileByteCount["file"]

Fornece o número de bytes do arquivo file.

FileDate["file"]

Fornece a data e hora da última modificação do arquivo file.

FileNames[]

Lista de todos os arquivos dentro do diretório ativo

FileType["file"]

Indica o tipo de objeto (do DOS) que tem o nome "file", ou seja, **None** (caso de não existir), **File** ou **Directory**.

Find[stream, "text"]

Acha a primeira linha de um fluxo de entrada que contém uma seqüência de caracteres especificada.

FindList["file", "text"]

Fornece a lista das linhas no arquivo file que contém a seqüência de caracteres especificada.

FindMinimum[f,{x, x₀}

Procura um mínimo local de f, iniciando a partir de $x = x_0$.

FindRoot[expr₁==expr₂,{x, x₀}

Procura uma solução numérica da equação $expr_1 == expr_2$, iniciando a partir de $x = x_0$.

First[expr]

Primeiro elemento da expressão expr.

Fit[data, funs, vars]

Ajuste linear pelos mínimos quadrados dos dados data como uma combinação linear das funções funs das variáveis vars.

FixedPoint[f, expr]

Seqüência de iterações que começam por expr e, em seguida, aplicam f, de maneira repetida, até o resultado não variar mais.

FixedPoint[f, expr, n]

Seqüência de iterações que começam pela expressão expr e, em seguida, aplicam f, de maneira repetida n vezes.

FixedPointList[f, expr]

Lista da seqüência dos resultados das iterações que começam por expr e, em seguida, aplicam f, de maneira repetida até o resultado não variar mais.

Flat

Atributo que pode ser associado ao símbolo f para indicar que todas as expressões que envolvem funções aninhadas devem ser reduzidas a um só nível.

Flatten[list]

Reduz as listas aninhadas contidas na lista list a um só nível.

Flatten[list, n]

Reduz as listas aninhadas contidas na lista list até o nível n.

Flatten[list, n, h]

Reduz de nível as expressões do n-ésimo elemento da lista list que começam pelo cabeçalho h.

FlattenAt[list, n]

Reduz de nível a sub-lista que aparece como n-ésimo elemento da lista list.

Floor[x]

Maior inteiro que é inferior ou igual a x.

Fold[f, x, list]

Fornece o último elemento de **FoldList[f, x, list]**.

FoldList[f, x, {a, b, c,...}]

Fornece {x, f[x, a], f[f[x, a], b], f[f[f[x, a], b], c]}.

FontForm[expr, {"font", size}]

Especifica que a expressão **expr** deve ser imprimida com a fonte **font** usando o tamanho **size** (expressos em pontos, isto é, em 1/72 de polegada, ou seja, 0.35 mm aproximadamente).

For[start, test, incr, body]

Laço que executa a ou as condições iniciais **start** e repete a incrementação **incr** e o corpo do procedimento **body** enquanto que o teste **test** dá **True** como resultado.

Format[expr]

Imprime a versão formatada de **expr**.

Format[expr, type]

Imprime a versão formatada de **expr** com a indicação do tipo **type** de formato que pode ser:

CForm, FortranForm, InputForm, OutputForm, TeXForm, TextForm.

FormatType

Opção para fluxos de saída que especifica o tipo de formato padrão usado para imprimir expressões.

FortranForm[expr]

Imprime a versão da expressão **expr** em linguagem Fortran.

Fourier[list]

Transformação de Fourier discreta de uma lista **list** de números complexos.

Frame

Opção para as funções gráficas bidimensionais que especifica se uma moldura deve ser desenhada arredor do gráfico.

FrameLabel

Opção para as funções gráficas bidimensionais que especifica as legendas para colocar nos lados da moldura arredor do gráfico.

FrameStyle

Opção para as funções gráficas bidimensionais que especifica como devem ser desenhados os lados da moldura arredor do gráfico.

FrameTicks

Opção para as funções gráficas bidimensionais que especifica as marcas de referência nos lados da moldura arredor do gráfico.

Exemplos:

FrameTicks->None

FrameTicks->Automatic

FrameTicks->{x, y}

FreeQ[expr, form]

Fornece **False** se nenhuma das subexpressões de expr é do formato form.

FromCharacterCode[n]

Fornece uma seqüência com o caracter cujo código é o inteiro n.

FromCharacterCode[{n₁, n₂,...}]

Fornece uma seqüência com os caracteres de código inteiro n₁, n₂ etc.

FromDate[date]

Converte a data date do formato {ano, mês, dia, hora, minuto, segundo} para o número absoluto de segundos desde o início do 1^o de janeiro de 1900.

FullDefinition[s₁, s₂,...]

Imprime as definições dados aos símbolos s₁, s₂, ... e de todos os símbolos dos quais estes dependem.

FullForm[expr]

Imprime a expressão expr completa, sem sintaxe especial.

FullGraphics[g]

Gera, a partir de um objeto gráfico, um novo no qual os objetos especificados por opções gráficas são apresentados como listas explícitas de primitivas gráficas.

FullOptions[expr]

Fornece as especificações completas das opções explicitamente determinadas numa expressão como um objeto gráfico.

Function[body]

body é uma função pura. Os parâmetros formais (variáveis) são # (ou #1), #2 etc.

Function[x, body]

Função pura de um parâmetro formal (variável) x.

Function[{x₁, x₂,...}, body]

Função pura com uma lista de parâmetros formais (variáveis).

Gamma[z]

Função gamma de Euler $\Gamma(z)$.

Gamma[a, z]

Função gamma incompleta de Euler $\Gamma(a, z)$.

Gamma[a, z₀, z₁]

Função gamma incompleta de Euler $\Gamma(a, z_0) - \Gamma(a, z_1)$.

GammaRegularized[a, z]

Função gamma incompleta regularizada $Q(a, z)$.

GaussianIntegers

Opção para **FactorInteger**, **PrimeQ** e funções afins que especifica se a fatoração deve ser feita com os inteiros gaussianos.

GCD[n₁, n₂,...]

Fornece o maior divisor comum dos inteiros n₁, n₂, ...

GegenbauerC[n, m, c]

Fornece o polinômio de Gegenbauer C_n^m(x).

General

Símbolo ao qual as mensagens do sistema geral são ligadas.

Get

<<name ou <<"name" lê um arquivo, avalia cada uma de suas expressões e retorna a última.

GoldenRatio

$\phi = (1 + \text{Sqrt}[5])/2 = 1.61803$.

Goto[tag]

Procura a marca **Label[tag]** e transfere o controle para este ponto.

Graphics[primitives, options]

Representa uma imagem gráfica bidimensional. A imagem é mostrada através de **Show**. As seguintes primitivas gráficas podem ser usadas:

| | |
|--|--|
| Circle {x, y}, r] | círculo |
| Disk {x, y}, r] | disco cheio |
| Line {x ₁ , y ₁ }, ...}] | linha |
| Point {x, y}] | ponto |
| Polygon {x ₁ , y ₁ }, ...}] | polígono cheio |
| PostScript "string"] | código PostScript para incluir legenda |
| Raster [array] | Tabela de níveis de cinza |
| Rectangle {xmin, ymin},{xmax, ymax}] | retângulo cheio |
| Text [expr, {x, y}] | texto |

Graphics[primitives, options] (Continuação)

- As primitivas de som **SampleSoundList** e **SampleSoundFunction** podem ser incluídas também.

- As seguintes diretrizes gráficas podem ser usadas:

| | |
|---|--|
| AbsoluteDashing [[d ₁ , ...]] | Especificação absoluta para linha pontilhada |
| AbsolutePointSize [d] | Especificação absoluta de tamanho de ponto |
| AbsoluteThickness [d] | Especificação absoluta de espessura de linha |
| CMYKColor [c, m, y, k] | Especificação de cor |
| Dashing [[r ₁ , ...]] | Especificação para linha pontilhada |
| GrayLevel [i] | Especificação de intensidade |
| Hue [h] | Especificação de matiz |
| RGBColor [r, g, b] | Especificação de cor |
| PointSize [r] | Especificação de tamanho de ponto |
| Thickness [r] | Especificação de espessura de linha |

- As seguintes opções podem ser dadas:

| | | |
|------------------------|--------------------------|--|
| AspectRatio | 1/GoldenRatio | relação altura/largura |
| Axes | False | desenhar eixos (sim: True / não: False) |
| AxesLabel | None | legenda nos eixos (nenhuma: none) |
| AxesOrigin | Automatic | ponto de cruzamento dos eixos |
| AxesStyle | Automatic | estilo para os eixos |
| BackGround | Automatic | cor de fundo |
| ColorOutput | Automatic | saída colorida para gerar |
| DefaultColor | Automatic | cor padrão para gráfico |
| DefaultFont | \$DefaultFont | fonte padrão para texto |
| DisplayFunction | \$DisplayFunction | função para gerar saída |
| Epilog | { } | primitivas gráficas após gráfico |
| Frame | True | moldura (sim ou não) |
| FrameLabel | None | legenda na moldura |
| FrameTicks | Automatic | marcas na moldura |

Graphics[primitives, options] (Continuação)

| | | |
|--------------------|------------------|--|
| FrameStyle | Automatic | diretrizes gráficas do estilo da moldura |
| FrameTicks | Automatic | marcas na moldura |
| GridLines | None | linhas de grade |
| PlotLabel | None | legenda para o gráfico |
| PlotRange | Automatic | domínio das valores de z |
| PlotRegion | Automatic | região que o gráfico ocupará |
| Prolog | { } | primitivas gráf. antes do gráfico |
| RotateLabel | True | girar a legenda do eixo y |
| Ticks | Automatic | marcas nos eixos |

- Listas aninhadas de gráficos podem ser produzidas. Especificações como **Graylevel** permanecem ativas apenas até o fim da lista que as contem.

- O formato padrão de saída para **Graphics[]** é **-Graphics-**. A instrução **InputForm** imprime a lista explícita das primitivas.

Graphics3D[primitive, options]

Representa uma imagem gráfica tridimensional. A imagem é mostrada através de **Show**. As seguintes primitivas gráficas podem ser usadas:

| | |
|--|----------|
| Cuboid[{xmin, ymin, zmin}, ...] | cubóide |
| Line[{{x₁, y₁, z₁}, ...}] | linha |
| Point[{x, y, z}] | ponto |
| Polygon[{{x₁, y₁, z₁}, ...}] | polígono |
| Text[expr, {x, y, z}] | texto |

- As primitivas de som **SampleSoundList** e **SampleSoundFunction** podem ser incluídas também.

- As seguintes diretrizes gráficas podem ser usadas:

| | |
|--|--|
| AbsoluteDashing[{d₁, ...}] | Especificação absoluta para linha pontilhada |
| AbsolutePointSize[d] | Especificação absoluta de tamanho de ponto |
| AbsoluteThickness[d] | Especificação absoluta de espessura de linha |
| CMYKColor[c, m, y, k] | Especificação de cor |
| Dashing[{r₁, ...}] | Especificação para linha pontilhada |

Graphics3D[primitive, options] (*Continuação*)

| | | |
|---------------------------|--|--|
| EdgeForm[spec] | | especificação de beiras de polígono |
| FaceForm[spec] | | especificação de faces de polígono |
| GrayLevel[i] | | Especificação de intensidade |
| Hue[h] | | Especificação de matiz |
| RGBColor[r, g, b] | | Especificação de cor |
| SurfaceColor[spec] | | Especificação de propriedades de superfícies |
| Thickness[r] | | Especificação de espessura de linha |

- As seguintes opções podem ser dadas:

| | | |
|------------------------|--------------------------|--|
| AmbientLight | Graylevel[0.] | nível de iluminação ambiente |
| AspectRatio | Automatic | relação altura/largura |
| Axes | False | desenhar eixos (sim: True / não: False) |
| AxesLabel | None | legenda nos eixos (nenhuma: none) |
| AxesStyle | Automatic | estilo para os eixos |
| BackGround | Automatic | cor de fundo |
| Boxed | True | com ou sem caixa limite |
| BoxRatios | Automatic | relação entre os lados da caixa limite |
| BoxStyle | Automatic | estilo da caixa limite |
| ColorOutput | Automatic | saída colorida para gerar |
| DefaultColor | Automatic | cor padrão para gráfico |
| DefaultFont | \$DefaultFont | fonte padrão para texto |
| DisplayFunction | \$DisplayFunction | função para gerar saída |
| Epilog | { } | primitivas gráficas após gráfico |
| FacesGrid | None | linhas de grade na caixa limite |
| Lighting | True | uso ou não de iluminação simulada |
| LightSources | | posições e cores das fontes de luz |
| PlotLabel | None | legenda para o gráfico |
| PlotRange | Automatic | domínio das valores de z |
| PlotRegion | Automatic | região que o gráfico ocupará |

Graphics3D[primitive, options] (*Continuação*)

| | | |
|------------------------|------------------------|---|
| RenderAll | True | representar ou não todos os polígonos |
| Shading | True | sombrear ou não os polígonos |
| SphericalRegion | False | caber ou não esferas na região do gráfico |
| Ticks | Automatic | marcas nos eixos |
| Viewcenter | Automatic | ponto a ser colocado no centro da região do gráfico |
| ViewPoint | {1.3, -2.4, 2.} | posição do observador |
| ViewVertical | {0, 0, 1} | direção que deve ser vertical |

- Listas aninhadas de gráficos podem ser produzidas. Especificações como **Graylevel** permanecem ativas apenas até o fim da lista que as contem.

- O formato padrão de saída para **Graphics3D[]** é **-Graphics3D-**. A instrução **InputForm** imprime a lista explícita das primitivas.

GraphicsArray[g₁, g₂,...]

Representa uma linha de objetos gráficos.

GraphicsArray[{{g₁, g₂,...},...}]

Tabela bidimensional de objetos gráficos.

GraphicsSpacing

Opção para **GraphicsArray** que especifica o espaçamento entre seus elementos.

GrayLevel[level]

Diretriz gráfica que especifica a intensidade em tons de cinza com a qual o objeto que segue deve ser visualizado. Deve ser um número entre 0 e 1.

Greater

x > y fornece **True** se **x** é maior do que **y** e **False** no caso contrário.

GreaterEqual

x >= y fornece **True** se **x** é maior ou igual a **y** e **False** no caso contrário.

GridLines

Opção para funções gráficas bidimensionais que especifica as linhas de grade.

GroebnerBasis[{poly₁, poly₂,...},{x₁, x₂,...}]

Fornecer a lista dos polinômios que formam uma base de Gröbner para o ideal gerado pelos polinômios poly_i.

Head

Cabeçalho de expr.

Exemplos:

Head[f[x]]→f

Head[a+b]→Plus

Head[4]→Integer

Head[x]→Symbol

Heads

Opção para as funções que usam especificações de nível para inclusão ou não dos cabeçalhos das expressões.

HoldPart[expr, pos]

Extraí parte ou partes de **expr** especificadas por **pos** e as empacota, cada uma, em **Hold**.

HermiteH[n, x]

Polinômio de Hermite $H_n(x)$.

HiddenSurface

Opção para **SurfaceGraphics** que especifica se as superfícies escondidas devem ser eliminadas (com **HiddenSurface**→**True**) ou não.

Hold[expr]

Mantem **expr** numa forma não avaliada.

HoldAll

Atributo que especifica que todos os argumentos de uma função devem ser mantidos numa forma não avaliada.

HoldFirst

Atributo que especifica que o primeiro argumento de uma função deve ser mantido numa forma não avaliada.

HoldForm[expr]

Imprime a expressão expr mantida de uma forma não avaliada.

HoldRest

Atributo que especifica que todos menos o primeiro argumento de uma função devem ser mantidos numa forma não avaliada.

HomeDirectory[]

Fornece seu diretório em uso. Nos sistemas operacionais multi-usuários, **HomeDirectory[]** fornece o diretório principal do usuário ativo.

Hue[h]

Diretriz gráfica que especifica que o objeto concernido deve ser mostrado, se possível, em cores correspondentes à matiz h.

Hue[h, s, b]

Diretriz gráfica que especifica as cores em termos de matiz h, saturação s e brilho b.

HyperGeometric0F1[a, z]

Função hypergeométrica ${}_0F_1(a;z)$.

HyperGeometric1F1[a, b, z]

Função hypergeométrica confluyente de Kummer ${}_1F_1(a;b;z)$.

HyperGeometric2F1[a, b, c, z]

Função hypergeométrica ${}_2F_1(a;b;c;z)$.

HyperGeometricU[a, b, z]

Função hypergeométrica confluyente $U(a;b;z)$.

I

Símbolo da unidade imaginária, raiz quadrada de -1.

Identity[expr]

Fornece *expr* (operação identidade).

IdentityMatrix[n]

Matriz identidade $n \times n$.

If[condition, t, f]

Devolve *t* se a condição *condition* é avaliada como True e *f* se ela é avaliada como False.

IgnoreCase

Opção para manipulação de seqüências de caracteres e funções de procura que especifica que letras minúsculas ou maiúsculas devem ser tratadas como equivalentes.

Im[z]

Parte imaginária do número complexo *z*.

Implies[p, q]

Implicação lógica $p \Rightarrow q$. **Implies[p, q]** é equivalente a $\neg p \vee q$.

In[n]

Objeto global que foi indicado para tomar o valor esperado da *n*-ésima linha.

Increment

x++ utiliza o valor de *x* e, em seguida, aumenta este valor de *x* de 1.

Indent[d]

Imprime um deslocamento para direita de largura *d* antes da subexpressão que segue.

Indeterminate

Símbolo que representa uma quantidade numérica cujo valor não pode ser determinado.

Infinity

Símbolo que representa uma quantidade positiva infinita.

Infix[f[e1, e2,...]]

Imprime $f[e_1, e_2, \dots]$ na forma prefixada padrão

$e_1 \sim f \sim e_2 \sim f \sim e_3 \dots$

Infix[expr, h]

Imprime $f[e_1, e_2, \dots]$ com os argumentos separados por h :

$e_1 h e_2 h e_3 \dots$

Information[symb]

Imprime as informações sobre o símbolo *symb*.

Inner[f, list1, list2, g]

Generalização de Dot (produto interno) no qual f representa a multiplicação e g , a adição.

Exemplo:

$\text{Inner}[f, \{a, b\}, \{x, y\}, g] \rightarrow g[f[a, x], f[b, y]]$

$\text{Inner}[f, \{a, b\}, \{c, d\}, \{x, y\}, g] \rightarrow \{g[f[a, x], f[b, y]], g[f[c, x], f[d, y]]\}$.

Input[]

Lê de maneira interativa uma expressão.

Input["prompt"]

Solicita uma entrada, através da mensagem contida na sequência de caracteres "prompt".

InputForm[expr]

Imprime uma versão de *expr* adequada para entrada no *Mathematica*.

InputStream["name", n]

Objeto que representa um fluxo de entrada para funções como **Read** e **Find**.

InputString[]

Lê de maneira interativa uma seqüência de caracteres.

InputString["prompt"]

Solicita a entrada de uma seqüência de caracteres, através da mensagem contida na seqüência de caracteres "prompt".

Insert[list, elem, n]

Insere o elemento elem na posição {i, j,...} na lista list.

Exemplo:

Insert[{a,b,c},x,2]-->{a,x,b,c}

Insert[list, elem,

{i, j,...}]

Insere o elemento elem na posição n na lista list. Se n é negativo, a posição é contada a partir do fim.

Exemplo:

Insert[{a,b,c},x,{1},{-1}]-->{x,a,b,c,x}

Install["command"]

Executa um programa externo e instala definições do *Mathematica* para chamar funções nele.

InString[n]

Objeto global que foi indicado para ser o texto da n-ésima linha.

Integer

Cabeçalho usado para inteiros.

IntegerDigits[n]

Fornece a lista dos dígitos decimais do inteiro n.

IntegerDigits[n, b]

Fornece a lista dos dígitos decimais com base b do inteiro n.

IntegerQ[expr]

Fornece **True** e **expr** é um inteiro e **False** no caso contrário.

Integrate[f, x]

Integral indefinida $\int f dx$

InterpolatingFunction[range, table]

Função aproximada cujos valores são determinados por interpolação.

InterpolatingPolynomial[data, var]

Fornece um polinômio da variável **var** que é o ajuste exato à lista de dados **data**.

Interpolation[data]

Constrói um objeto **InterpolatingFunction** que representa uma função aproximada de interpolação dos dados.

Interrupt[]

Gera uma interrupção.

Intersection[list₁, list₂,...]

Fornece uma lista ordenada dos elementos comuns a todas as listas **list_i**.

Inverse[m]

Matriz inversa da matriz quadrada **m**.

InverseFourier[list]

Fornece a transformada inversa discreta de Fourier de lista **list** de números complexos.

InverseFunction[f]

representa a função inversa da função **f**, tal que **InverseFunction[f][y]** fornece o valor de **x** para o qual **f[x]** é igual a **y**.

InverseFunctions

Opção de **Solve** e funções afins que especifica se as funções inversa devem ser usadas.

InverseJacobiSN[v, m]

Função elíptica inversa de Jacobi $\text{sn}^{-1}(v|m)$.

Existem 12 funções, de nome formado a partir de **InverseJacobiPQ** onde **P** e **Q** são todo os pares distintos das letras S, C, D e N.

InverseSeries[s, x]

Fornece, a partir da série *s* gerada por **Series**, a série inversa da função representada por *s*.

JacobiAmplitude[u, m]

Amplitude $\text{am}(u|m)$ das funções elípticas de Jacobi.

JacobiP[n, a, b, x]

Polinômio de Jacobi $P_n^{(a,b)}(x)$.

JacobiSN[u, m]

Função elíptica de Jacobi $\text{sn}\left(\frac{n}{m}\right)$.

Existem 12 funções, de nome formado a partir de **JacobiPQ** onde **P** e **Q** são todo os pares distintos das letras S, C, D e N.

JacobiSymbol[n, m]

Fornece o símbolo de Jacobi $\left(\frac{n}{m}\right)$.

JacobiZeta[phi, m]

Função zeta de Jacobi $Z\left(\frac{\phi}{m}\right)$.

Join[list₁, list₂,...]

Concatena as listas. **Join** pode ser usado com qualquer conjunto de expressões que têm o mesmo cabeçalho.

Label[tag]

Lugar numa expressão composta para o qual o controle é transferido por **Goto[tag]**

LaguerreL[n, a, x]

Polinómio de Laguerre $L_n^a(x)$.

Last[expr]

Último elemento da expressão expr.

LatticeReduce[{v₁, v₂, ...}]

Fornece a base reduzida para o conjunto de vetores v_i .

LCM[n₁, n₂,...]

Menor comum múltiplo dos inteiros n_i .

LeafCount[expr]

Número total de subexpressões indivisíveis de expr.

LegendreP[n, x]

Polinómio de Legendre $P_n(x)$.

LegendreP[n, m, x]

Polinómio associado de Legendre $P_n^m(x)$.

LegendreQ[n, z]

Função de Legendre de segunda espécie $Q_n(z)$.

LegendreQ[n, m, z]

Função associada de Legendre de segunda espécie $Q_n^m(x)$.

LegendreType

Opção para **LegendreP** e **LegendreQ** que especifica os cortes para as funções de Legendre no plano complexo.

Length[expr]

Número de elementos da expressão expr.

LerchPhi[z, s, a]

Função transcendental de Lerch $\phi(z, s, a)$.

Less

$x < y$ fornece True se x é menor do que y e False no caso contrário.

LessEqual

$x \leq y$ fornece True se x é menor ou igual a y e False no caso contrário.

LetterQ[string]

Fornece True se todos os caracteres da seqüência string são letras e False no caso contrário.

Level[expr, levelsp]

Fornece uma lista de subexpressões de $expr$ nos níveis especificados por $levelsp$.

Lighting[True]

Opção para **Graphics3D** e funções afins que especifica o uso da iluminação simulada nas imagens tridimensionais.

Lighting[False]

Opção para **Graphics3D** e funções afins que especifica o fato de não usar a iluminação simulada nas imagens tridimensionais.

LigthSources

Opção para **Graphics3D** e funções afins que especifica as propriedades dos pontos de iluminação simulada nas imagens tridimensionais.

Limit[expr, x->x0]

Acha o valor limite da expressão $expr$ quando x se aproxima de x_0 .

Line[{pt1}, {pt2},...]

Primitiva gráfica que representa uma linha que passa pelos pontos pt_i indicados.

LinearProgramming[c, m, b]

Acha o vector x que minimiza a quantidade $c \cdot x$ obrigada a satisfazer $m \cdot x \geq b$ e $x \geq 0$. Todos os elementos dos vectores c e b e da matriz m devem ser números.

LinearSolve[m, b]

Fornece o vector x solução da equação matricial $m \cdot x == b$.

LineBreak[n]

Saída entre a n -ésima e a $(n+1)$ -ésima linha numa expressão impressa de várias linhas.

List

$\{e_1, e_2, \dots\}$ é a lista dos elementos e_i .

Listable

Atributo que pode ser dado a um símbolo f para indicar que a função f deve automaticamente ser distribuída em relação às listas que aparecem entre seus elementos.

ListContourPlot[array]

Gera um mapa de contorno a partir de uma tabela array de valores de altura.

ListDensityPlot[array]

Gera um mapa de densidade a partir de uma tabela array de valores de altura.

ListPlay[{a₁, a₂,...}]

Toca um som cujas amplitudes sucessivas são os valores a_i da lista (tabela) indicada.

ListPlot[{y₁, y₂,...}]

Gráfico de uma lista de valores y_i , as coordenadas x sendo tomadas iguais a 1, 2,....

ListPlot[{{x₁,y₁}, {x₂,y₂},...}]

Gráfico de uma lista de valores de coordenadas y_i e x_i .

ListPlot3D[array]

Gráfico tridimensional de uma superfície representando uma tabela array de alturas.

ListPlot3D[array, shades]

Gráfico tridimensional de uma superfície representando uma tabela array de alturas, com sombras especificadas por shade.

Literal[expr]

Equivalente a expr do ponto de vista da formatação, porém mantém expr sem avaliá-la.

Locked

Atributo que impede a modificação de qualquer atributo de um símbolo.

Log[z]

Logaritmo natural de z (logaritmo com base e).

Log[b, z]

Logaritmo de z com base b.

LogGamma[z]

Logaritmo da função Gama, ou seja, $\log \Gamma(z)$.

LogicalExpand[expr]

Expansão das expressões que contêm relações lógicas como **&&** e **||**.

LogIntegral[z]

Função logaritmo integral $li(z)$.

LowerCaseQ[string]

Fornece **True** se todos os caracteres da seqüência de caracteres string são letras minúsculas e **False** no caso contrário.

MachineNumberQ[expr]

Fornece **True** se *expr* é um número real com precisão da máquina e **False** no caso contrário.

MainSolve[Eqns]

Função implícita de transformação das equações *eqns*. Ela é chamada por **Solve** e **Eliminate**.

MantissaExponent[x]

Fornece uma lista que contem a mantissa e o expoente de um número real aproximado *x*.

Map[f, expr]

Aplica *f* a cada elemento do primeiro nível da expressão *expr*. É equivalente a *f* /@ *expr*.

MapAll[f, expr]

Aplica *f* a cada subexpressão da expressão *expr*. É equivalente a *f* //@ *expr*.

MapAt[f, expr, n]

Aplica *f* ao elemento que ocupa a posição *n* em *expr*. Se *n* for negativo, a posição é contada a partir do fim.

MapIndexed[f, expr]

Aplica *f* aos elementos de *expr*, apresentando a especificação de cada elemento como segundo argumento de *f*.

MapThread[f, {a1,a2,...}, {b1,b2,...}]

Fornece {*f*[*a1,a2,...*], *f*[*b1,b2,...*],...}.

MatchLocalNames

Opção para **Trace** e funções afins.

MatchQ[expr, form]

Fornece **True** se *expr* apresenta-se no formato *form* e **False** no caso contrário.

MatrixExp[mat]

Matriz exponencial da matriz mat.

Matrixform[list]

Imprime os elementos da lista list arrumados numa tabela.

MatrixPower[mat, n]

Fornece a n-ésima potência da matriz mat.

MatrixQ[expr]

Fornece **True** se a lista das listas de expr pode representar uma matriz e **False** no caso contrário.

Max[x₁, x₂,...]

Fornece o maior valor de x_i.

MaxBend

Opção para **Plot** que mede o maior ângulo de curvatura entre dois segmentos sucessivos num curva.

MaxMemoryUsed[]

Número máximo de bytes usados para armazenar todos os dados da sessão do *Mathematica* em andamento.

MemberQ[list, form]

Fornece **True** se um elemento da lista list apresenta-se no formato form e **False** no caso contrário.

Memoryconstrained[expr, b]

Avalia a expressão expr e pára se mais de b bytes de memória são necessários.

MemoryInUse[]

Número de bytes usados para armazenar todos os dados da sessão do *Mathematica* em andamento.

Mesh

Opção para **SurfaceGraphics** e **DensityGraphics** que especifica se uma malha explícita x-y deve ser desenhada.

MeshRange

Opção para **ListPlot3D**, **SurfaceGraphics** e **ListContourPlot**, **ListDensityPlot** e funções afins que especifica os domínios das coordenadas x e y que correspondem à tabela dos valores de z dados.

MeshStyle

Opção para **Plot3D**, **ContourPlot**, **DensityPlot** e funções afins que especifica como as linhas de malha devem ser representadas.

Message[symb::tag]

Imprime a mensagem `symb::tag` a menos que esta foi desligada.

MessageList[n]

Objeto global indicado para ser uma lista de nomes de mensagens gerada durante o processamento da n-ésima linha de entrada.

MessageName[symb, "tag"]

`symbol::tag` é uma mensagem que é convertida em **MessageName[symbol, "tag"]**. O comando `?f` mostra a mensagem `f::usage`

Exemplos:

`f::usage`

`f::example`

`f::notes`

Messages[symb]

Fornecer todas as mensagens associadas ao símbolo `symb`.

Min[x₁, x₂,...]

Fornecer o valor mínimo de x_i .

Minors[m, k]

Matriz constituída dos determinantes de todas as submatrizes $k \times k$ de m .

Minus

$-x$ é o valor negativo de x (obtido pela conversão de x por **Times[-1, x]**)

Mod[m, n]

Resto da divisão de m por n (módulo). O sinal de **Mod[m, n]** é sempre o de n .

Modular

Definição para a opção **Mode** de **Solve** ou outras funções afins que especifica que as equações devem ser satisfeitas apenas em termos de módulo de um inteiro.

Module[{x, y,...}, expr]

Especifica que a ocorrência dos símbolos x, y, \dots em **expr** devem ser tratadas como locais.

Modulus->n

opção de **Solve** e funções algébricas afins para especificar que os inteiros devem ser tratados módulo n .

MoebiusMu[n]

Função de Möbius $\mu(n)$.

Multinomial[n₁, n₂,...]

Coefficiente multinomial $(n_1+n_2+\dots)!/n_1!n_2!\dots$

N[expr]

Valor numérico de **expr**.

N[expr, n]

Valor numérico de **expr** com precisão de n dígitos.

NameQ["string"]

Fornece **True** se tem elementos que correspondem à seqüência de caracteres string, e **False** no caso contrário.

Names["string"]

Lista dos nomes dos símbolos que correspondem à seqüência de caracteres string.

Names["string", SpellingCorrection-True]

Lista dos nomes dos símbolos que correspondem à seqüência de caracteres string após correção ortográfica.

NBernoulliB[n]

Valor numérico do número de Bernoulli B_n .

NBernoulliB[n, d]

Valor numérico do número de Bernoulli B_n com precisão de d dígitos.

NDSolve[eq, y, {x, xmin, xmax}]

Acha a solução numérica das equações diferenciais eq para a função y da variável independente x, dentro do intervalo de xmin a xmax.

NDSolve[eq, {y1, y2,...}, {x, xmin, xmax}]

Acha a solução numérica das equações diferenciais eq para as funções y_1, y_2, \dots da variável independente x, dentro do intervalo de xmin a xmax.

Needs[context`"]

Carrega um arquivo apropriado se o contexto já especificado não se encontra ainda em **\$Packages**.

Negative[x]

Fornece **True** se x é um número negativo.

Nest[f, expr, n]

Fornece uma expressão com f aplicado n vezes à expressão expr.

NestList[f, expr, n]

Fornecer a lista dos resultados da aplicação de f à expressão $expr$ de 0 até n vezes.

NIntegrate[f, {x, xmin, xmax}]

Aproximação numérica da integral $\int_{xmin}^{xmax} f dx$.

NonCommutativeMultiply

$a ** b ** c$ é uma forma geral associativa porém não comutativa.

NonConstants

Opção para D que indica a lista dos objetos para serem tomados e que dependem de forma implícita das variáveis de diferenciação.

None

Parâmetro que significa nenhum e é usado para certas opções.

NonNegative[x]

Fornecer **True** se x é um número não negativo.

Normal[expr]

Converte $expr$ em uma expressão normal a partir de uma variedade de formas especiais.

Not

$!expr$ é a função lógica **NOT**. Ela fornece **False** se $expr$ é **True**, e **True** se $expr$ é **False**.

NProduct[f, {i, imin, imax}]

Aproximação numérica do produto $\prod_{imin}^{imax} f$.

NSolve[lhs==rhs, var]

Fornecer a lista das aproximações numéricas das raízes da equação polinomial.

NSum[f, {i, imin, imax}]

Aproximação numérica do somatório $\sum_{i_{\min}}^{i_{\max}} f$.

Null

Símbolo usado para indicar a ausência de uma expressão ou de um resultado.

NullRecords

Opção para **Read** e funções afins que especifica se registros nulos devem ser considerados entre separadores (de registros) repetidos.

NullSpace[m]

Lista dos vectores que formam uma base para o espaço nulo da matriz m.

NullWords

Opção para **Read** e funções afins que especifica se palavras nulas devem ser consideradas entre separadores (de palavras) repetidos.

Number

Número inteiro exato ou real aproximado em **Read**.

NumberForm[expr, n]

Imprime expr com números reais aproximados, com precisão de n dígitos.

NumberFormat

Opção de NumberForm e funções afins que especifica como a mantissa, a base e o expoente devem ser assemblados na forma final de impressão.

NumberPadding

Opção de NumberForm e funções afins que indica as seqüências de caracteres a serem usados para colocar de cada lado dos números.

NumberPoint

Opção de NumberForm e funções afins que indica qual deve ser a seqüência de caracteres a ser usada no lugar do ponto decimal.

NumberQ[expr]

Fornece **True** se expr é um número e **False** no caso contrário.

NumberSeparator

Opção de NumberForm e funções afins que indica qual deve ser a seqüência de caracteres a ser inserida como separador entre dígitos.

Configuração padrão:

NumberSeparator->","

NumberSigns

Opção de NumberForm e funções afins que indica qual devem ser as seqüências de caracteres a ser usada como sinal para números positivos e negativos.

Configuração padrão:

NumberSigns->{"-", ""}

Numerator[expr]

Numerador de expr.

O[x]^n

Representa um termo de ordem x^n .

OddQ[expr]

Fornece **True** se expr é um inteiro ímpar e **False** no caso contrário.

Off[symb::tag]

Desliga a mensagem de tal maneira que não será mais mostrada/impressa.

On[symb::tag]

Liga a mensagem de tal maneira que ela possa ser mostrada/impressa.

OnIdentity

Atributo que pode ser dado a um símbolo f para indicar que f[x], f[f[x]] etc. São todos equivalentes a x para fins de identificação de formato.

OpenAppend["file"]

Abre o arquivo file para acrescentar nele dados de saída e devolve um objeto **OutputStream**.

OpenRead["file"]

Abre o arquivo file para ler dados nele e devolve um objeto **InputStream**.

OpenTemporary["file"]

Abre de maneira temporária o arquivo file para escrever nele dados de saída e devolve um objeto **OutputStream**.

OpenWrite["file"]

Abre o arquivo file para escrever nele dados de saída e devolve um objeto **OutputStream**.

Operate[p, f[x, y]]

Fornece **p[f][x, y]**

Operate[p, expr, n]

Aplica p ao nível n no cabeçalho de expr.

Optional

p::v é um objeto de formato que representa uma expressão na forma de p, e que, se omitida, deve ser substituída por v.

Options[symb]

Fornece a lista das opções padrão atribuída ao símbolo symb.

Or

e₁ || e₂ || ... é a função lógica OR (ou). Ela avalia seus argumentos na ordem, e fornece **True** logo, se qualquer um deles é **True**, e fornece **False** se todos são **False**.

Order[expr₁, expr₂]

Fornece 1 se expr₁ é antes de expr₂ segundo a ordem canônica e -1 se expr₁ é depois de expr₂ segundo a ordem canônica, e 0 se a expressão expr₁ é idêntica a expr₂. **Order** utiliza a ordem canônica usada na função **Sort**.

Exemplos:

Order[a, b] ---> 1

Order[b, a] ---> -1

OrderedQ[h[[e₁, e₂,...]]]

Fornece **True** se os e_i estão em ordem canônica, e **False** no caso contrário. **Order** utiliza a ordem canônica usada na função **Sort**.

OrderedQ[e, e] fornece **True**.

Orderless

Atributo que pode ser atribuído ao símbolo f para indicar que os elementos e_i nas expressões do tipo f[e₁, e₂,...] devem automaticamente serem ordenadas segundo a ordem canônica.

Out[n]

%n é um objeto global que toma o valor gerado pela n-ésima linha de saída. % fornece o último resultado obtido. %% fornece o penúltimo resultado obtido. **Out[k]** ou seja %%...% (k vezes) fornece o k-ésimo resultado anterior.

Outer[f, list₁, list₂,...]

Fornece o produto externo generalizado das listas list_i.

Exemplo:

Outer[f, {a,b}, {x,y}] ---> {{f[a,x], f[a,y]}, {f[b,x], f[b,y]}}

OutputForm[expr]

Imprime expr segundo o formato padrão do *Mathematica*.

OutputStream["name", n]

Objeto que representa um fluxo de saída para funções como **Write**.

PaddedForm[expr, n]

Imprime todos os número de expr acolchoados para deixar espaço para um total de n dígitos.

PaddedForm[expr, {n, f}]

Imprime todos os número de expr como números aproximados de exatamente f dígitos à direita do ponto decimal (equivalente à vírgula) e acolchoados para deixar espaço para um total de n dígitos.

PageHeight

Opção para fluxos de saída que especifica quantas linhas de texto devem ser impressas entre as mudanças de página.

PageWidth

Opção para fluxos de saída que especifica a largura das linhas de texto para a impressão.

ParametricPlot[{f_x, f_y}, {t, tmin, tmax}]

Gráfico paramétrico de coordenadas f_x e f_y que são funções da variável t que varia entre $tmin$ e $tmax$.

ParametricPlot3D[{f_x, f_y, f_z}, {t, tmin, tmax}]

Curva paramétrica tridimensional de coordenadas f_x , f_y e f_z que são funções da variável t que varia entre $tmin$ e $tmax$. **ParametricPlot3D** gera um objeto **Graphics3D**.

Exemplo de opção:

Compiled->True

Opções padrão:

Axes->True

PlotPoints->Automatic (isto é: **PlotPoints->75** para curvas e

PlotPoints->{15, 15} para superfícies)

ParentDirectory[]

Fornece o diretório pai do diretório ativo.

ParentDirectory["dir"]

Fornece o diretório pai do diretório dir.

Part[expr, i]

expr[[i]] fornece o i-ésimo elemento de expr.

expr[[-i]] fornece o i-ésimo elemento de expr, contado a partir do último.

expr[[0]] fornece o cabeçalho de expr.

Part[expr, i, j,...]

expr[[i, j,...]] é equivalente a **expr[[i]][j]....**

Partition[list, n]

Fornecer uma partição da lista list em sublistas adjacentes de comprimento n.

Partition[list, n, d]

Fornecer uma partição da lista list em sublistas de comprimento n, com defasagem de d.

PartitionP[n]

Fornecer o número p(n) de partições do inteiro n.

PartitionsQ

Fornecer o número q(n) de partições em elementos distintos do inteiro n.

Pattern

s:obj representa o objeto de formato obj e de nome s.

PatternTest

p?test é o objeto de formato que toma o lugar de qualquer expressão que tem o formato p e para a qual a aplicação de test fornece **True**.

Pause[n]

Pára durante pelo menos n segundos.

Permutations[list]

Gera a lista de todas as permutações possíveis dos elementos de list.

Exemplo:

Permutations[{a, b, c}] →

{a, b, c}, {a, c, b}, {b, a, c}, {b, c, a}, {c, a, b}, {c, b, a}

Pi

π com o valor numérico ≈ 3.14159 .

Play[f, {t, tmin, tmax}]

Toca um som cuja amplitude é dada por f que é função de t (em segundos) que varia entre tmin e tmax. **Play** gera um objeto de **Sound** (som)

Possíveis opções:

Compiled→True

DisplayFunction→\$SoundDisplayFunction

Epilog→{}

PlayRange→Automatic

Prolog→{}

SampleDepth→8

SampleRate→8192

PlayRange

Opção de **Play** e funções afins que especifica qual deve ser o domínio de amplitudes de som.

Plot[f, {x, xmin, xmax}]

Gera um gráfico de f como função de x que varia de xmin até xmax.

Plot[{f1, f2,...},{x, xmin, xmax}]

Gera o gráfico de várias funções f1, f2, ... de x.

Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}]

Gera um gráfico tridimensional de f como função de x e y.

PlotDivision

Opção de **Plot** que especifica o número máximo de subdivisões que devem ser criadas na tentativa de gerar um curva suave.

PlotJoined

Opção de **ListPlot** que especifica se os pontos do gráfico devem unidos por uma linha.

PlotLabel

Opção das funções gráficas que indica a legenda geral de um gráfico.

PlotPoints

Opção das funções gráficas que especifica quantos pontos utilizar.

PlotRange

Opção para as funções gráficas que especifica os pontos a incluir num gráfico.

PlotRange->All

Opção para as funções gráficas que inclui todos os pontos.

PlotRange->Automatic

Opção para as funções gráficas que elimina os pontos isolados fora da região de interesse.

PlotRange->{min, max}

Opção para as funções gráficas que explicita os limites para y (gráficos 2D) ou z (gráficos 3D).

PlotRange->{{xmin, xmax}, ...}

Opção para as funções gráficas que explicita todos limites.

PlotRegion

Opção para as funções gráficas que especifica a região que deve ocupar um gráfico na exibição final.

PlotStyle

Opção para **List** e **ListPlot** que especifica o estilo das linhas ou dos pontos desenhados.

Plus

$x + y + z$ fornece a soma dos valores x , y e z .

Pochhammer[a, n]

Símbolo de Pochhammer $(a)_n$.

Point[x, y, z]

Primitiva gráfica que representa um ponto de coordenadas x , y e z .

PointSize[r]

Diretriz gráfica que especifica o raio r da região circular com a qual os pontos devem ser desenhados. O raio r é expresso como fração da largura total do gráfico.

PolyGamma[z]

Função digama $\psi(z)$.

PolyGamma[n, z]

n -ésima derivada $\psi^{(n)}(z)$ da função digama $\psi(z)$.

Polygon[{pt₁, pt₂,...}]

Primitiva gráfica que representa um polígono cheio.

PolygonIntersections

Opção para **Graphics3D** que especifica se os polígonos que se encontram devem ser mantidos sem modificação.

Opções:

PolygonIntersections->True

PolygonIntersections->False

PolyLog[n, z]

Função polilogarítmica $Li_n(z)$

PolynomialGCD[poly₁, poly₂,...]

Maior divisor comum dos polinômios poly₁, poly₂,...

PolynomialLCM[poly₁, poly₂,...]

Menor comum múltiplo dos polinômios poly₁, poly₂,...

PolynomialMod[poly, m]

Polinômio poly reduzido módulo m.

PolynomialMode[poly, {m₁, m₂, ...}]

Polinômio poly reduzido módulo todos os m_i.

PolynomialQ[expr, var]

Fornece **True** se expr é um polinômio da variável var, e **False** no caso contrário.

PolynomialQuotient[p, q, x]

Fornece o resultado da divisão do polinômio p pelo polinômio q, ambos considerados como polinômios de x; o resto é descartado.

PolynomialRemainder[p, q, x]

Fornece o resto da divisão do polinômio p pelo polinômio q, ambos considerados como polinômios de x.

Position[expr, pattern]

Fornece a lista das posições onde objetos de formato pattern aparecem na expressão expr.

Position[expr, pattern, levspec]

Fornece a lista das posições onde objetos de formato pattern aparecem na expressão expr nos níveis especificados por levspec.

Positive[x]

Fornece **True** se x é um número positivo.

PostFix[f[expr]]

Imprime f[expr] apresentado da forma pós-fixada: **expr // f**.

PostFix[f[expr], h] imprime como **exprh**.

PostScript[string₁", string₂",...]

Primitiva gráfica que especifica o código PostScript para a inclusão de texto nas saídas gráficas.

Power

x^y fornece x elevado à potência y .

PowerExpand[expr]

Expande todas as potências e todas as potências dos produtos de $expr$.

PowerMod[a, b, n]

$a^b \bmod n$.

PrecedenceFormPrecision[expr, prec]

Imprime a expressão $expr$ com parênteses como ela estaria se fosse contida num operador de precedência $prec$.

Precision[x]

Fornece o número de dígitos de precisão do número x .

PrecisionGoal

Opção para várias operações numéricas que especifica quantos dígitos de precisão deve ter o resultado final.

PreDecrement

$--x$ diminui o valor de x de 1 e fornece o novo valor de x .

Prefix[f[expr]]

Imprime $f[expr]$ apresentado da forma pré-fixada padrão: **f @ expr**.

PreFix[f[expr], h] imprime como **hexpr**.

PreIncrement

$++x$ incrementa o valor de x de 1 e fornece o novo valor de x .

Prepend[expr, elem]

Fornece expr com o elemento elem acrescentado a na primeira posição.

Exemplos:

Prepend[{a, b}, x] → {x, a, b}

Prepend[f[a], x+y] → f[x+y, a]

PrependTo[s, elem]

Acrescenta elem a s na primeira posição, sem avaliar o novo s.

PrependTo[s, elem] é equivalente a **s = Prepend[s, elem]**

Prime[n]

Fornece o n-ésimo número primo.

PrimePi[x]

Número de números primos $\pi(x)$ inferiores ou igual a x.

PrimeQ[expr]

Fornece **True** se expr é um número primo e **False** no caso contrário.

Print[expr₁, expr₂,...]

Imprime as expressões expr_i, a última sendo seguida de um avanço de linha (line feed)

PrintForm[expr]

Imprime a representação interna para impressão de expr.

Product[f, {i, imin, imax}]

Avalia o produto dos valores de f para i variando de imin até imax.

Prolog

Opção para funções gráficas que fornece a lista das primitivas gráficas a serem montadas antes que a parte principal do gráfico seja montada.

Protect[s₁, s₂,...]

Coloca o atributo **Protected** (protegido, isto é é inalterável) nos símbolos s_i.

Protected

Atributo que impede qualquer valor associado a um símbolo de ser modificado.

Pseudoinverse[m]

Matriz pseudo-inversa da matriz retangular m .

Put[expr, "filename"]

expr >> filename grava a expressão $expr$ no arquivo $filename$.

Put[expr₁, expr₂, ..., "filename"]

Grava a seqüência das expressões $expr_i$ no arquivo $filename$.

PutAppend[expr, "filename"]

expr >>> filename grava a expressão $expr$ após o antigo conteúdo do arquivo $filename$ (sem apagá-lo).

PutAppend[expr₁, expr₂, ..., "filename"]

Grava a seqüência das expressões $expr_i$ após o antigo conteúdo do arquivo $filename$ (sem apagá-lo).

QRDecomposition[m]

Fornece a decomposição QR de uma matriz numérica m . O resultado é uma lista $\{q, r\}$ onde q é uma matriz ortogonal e r , uma matriz triangular superior.

Quartics

Opção para **Roots** e funções afins que especifica se as soluções explícitas devem ser geradas para as equações quárticas irredutíveis.

Quit[]

Termina uma sessão do *Mathematica*.

Quotient[n, m]

Fornece o inteiro quociente de n por m .

Random[]

Fornece um número pseudo-aleatório, uniformemente distribuído, entre 0 e 1.

Random[type, {min, max}, n]

Fornece um número pseudo-aleatório do tipo `type` especificado, uniformemente distribuído, entre `min` e `max`. O número de dígitos de precisão pode ser indicado pelo valor de `n`.

Exemplos:

Random[Real, {4.2, 9.3}, 7]

Random[Complex, {zmin, zmax}]

Range[imax]

Gera a lista {1, 2, ..., imax}

Raster[{a₁₁, a₁₂,...},...]

Primitiva gráfica bidimensional que representa uma tabela retangular de células cinzas.

RasterArray[{g₁₁, g₁₂,...},...]

Primitiva gráfica bidimensional que representa uma tabela retangular de células coloridas segundo as diretrizes gráficas g_{ij} .

Rational

Cabeçalho usado para números racionais.

Rationalize[x]

Converte o número real x , próximo de um número racional, em um número racional.

Rationalize[x, dx]

Converte o número real x em um número racional quando o erro feito nesta conversão é menor do que dx .

Raw[h, "hexstring"]

Constrói um objeto de dados brutos de cabeçalho h e cujo conteúdo é a seqüência `hexstring` de formato binário, interpretada como número hexadecimal.

Re[z]

Parte real do número complexo z

Read[stream]

Lê uma expressão do fluxo de entrada e fornece uma expressão.

Read[stream, type]

Lê um objeto do tipo type.

Tipos para leitura:

Byte

Character

Expression

Number

Real

Record

String

Word

Possíveis opções:

NullRecords->False

NullWords->False

RecordSeparators->{"\"}

TokenWords->{ }

WordSeparators->{" ", "\"}

Read[stream, {type₁, type₂,...}]

Lê uma seqüência de objetos dos tipos type_i especificados.

ReadList["file"]

Lê num arquivo "file" as expressões que não foram lidas ainda e as devolve como uma lista.

ReadList["file", type]

Lê num arquivo "file" objetos do tipo especificado até atingir o fim do arquivo. Obte-se a lista dos objetos lidos.

ReadList["file", {type₁, type₂,...}]

Lê num arquivo "file" objetos dos tipos especificados até atingir o fim do arquivo. Obte-se a lista dos objetos lidos.

ReadList["file", {type₁, type₂,...}, n]

Lê num arquivo "file" apenas os n primeiros objetos dos tipos especificados. Obte-se a lista dos objetos lidos.

ReadProtected

Atributo que impede que os valores associados a um símbolo sejam vistos.

Real

Cabeçalho utilizado para os números reais (ponto flutuante, isto é, vírgula flutuante)

RealDigits[x]

Lista dos dígitos do número real aproximado x, junta com o número de dígitos que são a esquerda do ponto decimal.

RealDigits[x, b]

Lista dos dígitos a base b do número real aproximado x.

Record

Representa um registro em **Read**, **Find** e funções afins.

RecordLists

Opção para **ReadList** que especifica se os objetos de registros separados devem ser retornados em sublistas separadas.

Exemplos:

RecordLists->True

RecordLists->False

RecordSeparators

Opção de **Read**, **Find** e funções afins que especifica a lista das seqüências dos caracteres que devem ser usados como delimitadores de registros.

RecordSeparators (Continuação)

Opção padrão:

RecordSeparators->{"\n"} (cada linha é considerada como um registro)

Rectangle[{xmin, ymin}, {xmax, ymax}]

Primitiva gráfica bidimensional que representa um retângulo cheio orientado paralelamente aos eixos.

Rectangle[{xmin, ymin}, {xmax, ymax}, graphics]

Primitiva gráfica bidimensional que representa um retângulo preenchido com o gráfico graphics e orientado paralelamente aos eixos.

Reduce[eqns, vars]

Simplifica as equações eqns, tentando resolver para as variáveis vars. As equações geradas por **Reduce** são equivalentes às equações eqns e contém todas as soluções possíveis.

Reduce[eqns, vars, elims]

Simplifica as equações, tentando eliminar as variáveis elims.

ReleaseHold[expr]

Retira **Hold** e **HoldForm** da expressão expr.

Remove[symb₁, ...]

Cancela os símbolos indicados completamente, de tal maneira que seus nomes não são mais reconhecidos pelo *Mathematica*.

RenameDirectory["dir₁", "dir₂"]

Renomea o diretório dir₁ com o nome dir₂.

RenameFile["file₁", "file₂"]

Renomea o arquivo file₁ com o nome file₂.

RenderAll

Opção de **Graphics3D** que especifica se PostScript deve ser gerado para todos os polígonos.

Repeated

`p..` é um objeto de formato que representa uma seqüência de uma ou mais expressões, cada uma análoga a `p`.

RepeatedNull

`p...` é um objeto de formato que representa uma seqüência de nenhuma ou mais expressões, cada uma análoga a `p`.

Replace[expr, rules]

Aplica a regra ou seqüência de regras `rules` na tentativa de transformar a expressão `expr` inteira.

ReplaceAll

`expr /. rules` aplica a regra ou seqüência de regras `rules` na tentativa de transformar cada subelemento da expressão `expr`.

ReplaceHeldPart[expr, Hold[new], n]

Fornecer uma expressão na qual o `n`-ésimo elemento de `expr` é substituído por `new`.

ReplaceHeldPart[expr, Hold[new], {i, j, ...}]

Fornecer uma expressão na qual os elementos de `expr` nas posições `{i, j, ...}` são substituídos por `new`.

ReplacePart[expr, new, n]

Fornecer uma expressão na qual o `n`-ésimo elemento de `expr` é substituído por `new`.

ReplaceRepeated

`expr //. rules` realiza substituições até a expressão `expr` não mudar mais.

ResetDirectory[]

Restabelece como diretório ativo o anterior.

Residue[expr, {x, x₀}

Acha o resíduo da expressão `expr` no ponto $x=x_0$.

Rest[expr]

Fornece a expressão expr amputada do seu primeiro elemento.

Resultant[poly₁, poly₂, var]

Calcula a resultante dos polinômios poly₁ e poly₂ em relação a variável var.

Return[expr]

Fornece o valor expr de uma função.

Return[]

Fornece o valor Null.

Reverse[expr]

Inversa a ordem dos elementos de expr.

RGBColor[red, green, blue]

Diretriz gráfica que especifica que os objetos gráficos serão mostrados, se for possível, na cor indicada. A cor é a resultante da "mistura" das três cores vermelho (red), verde (green) e azul (blue) nas proporções indicadas pelas variáveis red, green e blue (valores entre 0 e 1)

RiemannSiegelTheta[t]

Função de RiemannSiegel $\theta(t)$.

RiemannSiegelZ[t]

Função de Riemann-Siegel $Z(t)$.

Roots[lhs==, var]

Fornece a disjunção de equações que representam as raízes de uma equação polinomial.

RotateLabel

Opção para gráficos de duas dimensões que especifica se as legendas nos eixos verticais devem ser virados para tornarem-se verticais.

RotateLabel (*Continuação*)

Opção padrão:

RotateLabel->True

RotateLeft[expr, n]

Efetua a rotação dos elementos da expressão m de n elementos para a esquerda.

RotateLeft[expr]

Efetua a rotação dos elementos da expressão m de uma posição para a esquerda.

RotateRight[expr, n]

Efetua a rotação dos elementos da expressão m de n elementos para a direita.

RotateRight[expr]

Efetua a rotação dos elementos da expressão m de uma posição para a direita.

Round[x]

Inteiro mais próximo de x.

RowReduce[m]

Forma reduzida por linha da matriz m.

Rule

lhs -> rhs representa a regra de transformação de lhs em rhs.

RuleDelayed

lhs -> rhs representa a regra de transformação de lhs em rhs, avaliando rhs apenas quando a regra for utilizada.

Run[expr₁, expr₂,...]

Gera a forma impressa das expressões expr_i, separadas por espaços e a executa como um comando externo no sistema operacional.

RunThrough["command", expr]

Executa um comando externo gerando a forma impressa da expressão *expr* como entrada; Sua saída é lida como entrada do *Mathematica* e fornecendo assim o resultado final.

SameQ

lhs === rhs fornece **True** se *lhs* é idêntico a *rhs* e **False** no caso contrário..

SameTest

Opção para funções como **FixedPoint** que especifica a função de comparação a ser aplicada a pares de resultados para determinar se eles podem ser considerados como idênticos.

Opção padrão:

SameTest -> **Equal**

SampleDepth

Opção para primitivas de som que especifica quantos bits devem ser usados para codificar níveis de amplitudes de som.

Opção padrão:

SampleDepth -> **8** (o que permite 256 amplitudes diferentes)

SampleSoundFunction[f, n, r]

Primitiva de som que representa um som cuja amplitude, amostrada *r* vezes por segundo, é gerada pela aplicação da função *f* a inteiros sucessivos entre 1 e *n*.

SampleSoundList[{a1, a2, ...}, r]

Primitiva de som que representa um som cuja amplitude, amostrada *r* vezes por segundo, é igual à sucessão dos valores *a_j*.

SampleRate

Opção para primitivas de som que especifica o número de amostras por segundo usado para gerar sons.

Opção padrão:

SampleRate -> **8192** (a frequência maior, em Hertz, que pode existir num som é igual à metade do valor indicado para **SampleRate**).

Save["filename", symb₁, symb₂, ...]

Acrescenta as definições dos symb_i a um arquivo filename.

Scaled[{x, y,...}]

Indica a posição de um objeto gráfico em termos de coordenadas variando entre 0 e 1 nas respectivas dimensões da totalidade do gráfico.

Scan[f, expr]

Aplica f a cada elemento de expr.

Scan[f, expr, levelspec]

Aplica f às partes de expr especificadas por levelspec.

SchurDecomposition[m]

Fornece a decomposição de Schur da matriz m. O resultado é uma lista {q, t} onde q é uma matriz ortogonal e t, a matriz triangular do bloco superior.

ScientificForm[expr]

Imprime expr com todos seus números reais apresentados com notação científica.

Sec[z]

Secante, $\sec(z)$, do número complexo z.

Sech[z]

Secante hiperbólica, $\operatorname{sech}(z)$, do número complexo z.

SeedRandom[]

Reinicia o gerador de números pseudo-aleatórios usando o tempo horário como semente.

SeedRandom[n]

Reinicia o gerador de números pseudo-aleatórios usando o inteiro n como semente.

Select[list, crit]

Extrai todos os elementos e_i da lista list para os quais $\text{crit}[e_i]$ é True.

Select[list, crit, n]

Extrai os n primeiros elementos e_i da lista list para os quais $\text{crit}[e_i]$ é True.

SequenceForm[expr₁, expr₂,...]

Imprime como concatenação de texto os elementos expr_i para impressão.

Series[f, {x, x₀, n}]

Gera uma expansão em série de potências de f na vizinhança do ponto $x=x_0$ até a ordem $(x-x_0)^n$.

SeriesData[x, x₀, {a₀, a₁,...}, nmin, nmax, den]

Série de potências da variável x na vizinhança do ponto $x=x_0$. Os valores dos a_i são os coeficientes na série de potências. As potências dos $(x-x_0)$ que aparecem são $nmin/den$, $(nmin+1)/den$, ..., $nmax/den$.

SessionTime[]

Fornece o número total de segundos desde o início da sessão do *Mathematica*.

Set

lhs = rhs avalia rhs e faz com que este valor torne-se o valor de lhs. Daqui em diante, lhs é substituído por rhs a cada vez que aparece.

SetAccuracy[expr, n]

Fornece uma versão de expr na qual todos os números apresentam uma precisão de n dígitos.

SetAttributes[s, attr]

Acrescente o atributo attr ao símbolo s.

SetDelayed

lhs := rhs impõe rhs para ser o valor diferido de lhs. Então rhs é mantido de uma forma não avaliada. Quando lhs aparece, é substituído por rhs avaliado de novo a cada vez.

SetDirectory["dir"]

Determina o diretório ativo.

SetFileDate["file"]

Determina a data de modificação do arquivo "file" como sendo a data atual.

SetOptions[s, name1->value₁, name₁->value, ...]

Determina as opções padrão do símbolo s.

SetPrecision[expr, n]

Fornece uma versão de expr na qual todos os números apresentam uma precisão de n dígitos.

SetStreamPosition[stream, n]

Determina o ponto atual num fluxo aberto.

Shading

Opção para **SurfaceGraphics** que especifica se as superfícies devem ser sombreadas.

Shallow[expr]

Imprime a forma de menor profundidade de expr.

Share[expr]

muda a maneira da expressão expr ser armazenada internamente visando minimizar o espaço de memória usado.

Short[expr]

Imprime uma forma curta de expr, menor do que aproximadamente uma linha.

Show[Graphics, options]

Mostra gráficos bi- ou tridimensionais, usando as opções especificada.

Sign[x]

Fornece -1, 0 ou 1 a depender do valor de x ser igual a -1, 0 ou 1

Signature[list]

Fornece a assinatura (número) de permutações necessárias para por os elementos da lista list na ordem canónica.

SignPadding

Opção para **NumberForm** e funções afins que especifica se bordas devem ser colocadas após os sinais.

Simplify[expr]

Realiza uma seqüência de transformações algébricas sobre a expressão expr e fornece a mais simples forma achada.

Sin[z]

Seno, $\text{sen}(z)$ ou $\sin(z)$, do número complexo z.

SingularValues[m]

Decomposição em valores singulares de uma matriz numérica m. O resultado é uma lista {u, w, v} onde w é a lista dos valores singulares diferentes de zero; m pode ser escrita como **Transpose[u].DiagonalMatrix[w].v**.

Sinh[z]

Seno hiperbólico, $\text{sinh}(z)$, do número complexo z.

SinIntegral[z]

Função seno integral Si(z).

SixJSymbol[{j1, j2, j3}, {j4, j5, j6}]

Fornece os valores do símbolo de Racah 6-j.

Skeleton[n]

Representa a seqüência dos n elementos omitidos na impressão de uma expressão por **Short**. A forma padrão de impressão de **Skeleton** é <<n>>.

Skip[stream, type]

Omite um objeto do tipo especificado por type num fluxo de entrada stream.

Skip[stream, type, n]

Omite n objetos do tipo especificado por type num fluxo de entrada stream.

Slot[] ou **Slot[1]**

representa o primeiro argumento fornecido a uma função pura.

Slot[n]

#n representa o n-ésimo argumento fornecido a uma função pura.

#0 representa o cabeçalho da função pura.

SlotSequence[] ou**SlotSequence[1]**

representa a seqüência dos argumentos fornecidos a uma função pura.

SlotSequence[n]

##n representa a seqüência dos argumentos, começando com o n-ésimo, fornecidos a uma função pura.

Solve[eqns, vars]

Tenta resolver uma equação, ou um conjunto de equações eqns, para as variáveis vars.

Solve[eqns, vars, elims]

Tenta resolver uma equação, ou um conjunto de equações eqns, para as variáveis vars, eliminando as variáveis elims.

SolveAlways[eqns, vars]

Fornecer os valores dos parâmetros que tornam as equações eqns válidas para todos os valores das variáveis vars.

Sort[list]

Ordena os elementos da lista list em ordem canônica. A ordem canônica para as seqüências é determinada pela ordem dos caracteres em **\$StringOrder**. Os símbolos são ordenados segundo seu texto literal. Os inteiros, racionais e números reais aproximados são ordenados pelos seus valores numéricos. As expressões são ordenadas pela comparação de suas partes pelos primeiros níveis de profundidade encontrados. As expressões mais curtas vêm em primeiro lugar.

Exemplo:

Sort[{b, c, a}] ---> {a, b, c}

Sort[list, p]

Aplica a função p a pares de elementos da lista list para determinar se eles estão em ordem.

Exemplo:

Sort[{4, 1, 3}, Greater] ---> {4, 3, 1}

Sound[primitives]

Representa o som que corresponde à ou às primitivas primitivas. As seguintes primitivas podem ser usadas:

SampleSoundFunction[f, n, r] para níveis de amplitudes gerados por uma função.

SampleSoundFunction[{a1, a2,...}, r] para níveis de amplitudes gerados por uma lista de valores a_i.

SpellingCorrection

Opção para **StringMatchQ**, **Names** e funções afins que especifica se as seqüências de caracteres devem ser consideradas como equivalentes. A opção padrão **SpellingCorrection** -> **False** exige a equiparação exata. **?name** utiliza efetivamente **SpellingCorrection** -> **True** quando não consegue achar a equiparação exata para o nome name.

SphericalHarmonicY[l, m, theta, phi]

Harmônica esférica $Y_l^m(\theta, \phi)$.

SphericalRegion

Opção para funções gráficas tridimensionais que especifica que a imagem final deve ser ajustada para a esfera desenhada arredor da caixa de enquadramento do gráfico caber na área especificada para o gráfico (**SphericalRegion -> True**).

SphericalRegion -> False força as imagens tridimensionais a ser tão grandes quanto possível dentro da área especificada.

Splice["file"]

Emenda a saída do *Mathematica* num arquivo externo. Toma textos inseridos entre < * e * > no arquivo file, os avalia como entrada do *Mathematica* e substitui os textos originais pela saída correspondente do *Mathematica*. Lê arquivo com nome name.mx e escreve a saída num arquivo de nome name.x. O formato padrão para a saída do *Mathematica* é indicada pela extensão do nome do arquivo de entrada:

name.mc **CForm**
name.mf **FortranForm**

Sqrt[z]

Raiz quadrada de z

Stack[]

Mostra a pilha de registros das operações em andamento, com a lista das etiquetas associadas às avaliações em andamento.

StackBegin[expr]

Avalia a expressão expr, recomeçando com uma nova pilha da avaliação.

StackComplete[expr]

Avalia a expressão expr, com a inclusão na pilha da avaliação das expressões intermediárias.

StackInhibit[expr]

Avalia a expressão expr, sem modificar a pilha da avaliação.

StirlingS1[n, m]

Número de Stirling de primeira ordem $S_n^{(m)}$.

StirlingS2[n, m]

Número de Stirling de segunda ordem $S_n^{(m)}$.

StreamPosition[stream]

Fornece um inteiro que especifica a posição do ponto corrido num fluxo aberto stream.

Streams[]

Fornece a lista de todos os fluxos abertos.

Streams["name"]

Fornece apenas a lista dos os fluxos abertos com o nome especificado.

Strings

Cabeçalho de uma seqüência de caracteres.

StringBreak[n]

Saída no fim da n-ésima linha quando uma seqüência de caracteres é quebrada.

StringBiteCount["string"]

Número total de bytes usado para armazenar os caracteres na seqüência de caracteres string.

StringConversion

Opção para funções de saída que especifica como devem sair as seqüências de caracteres que contêm caracteres especiais.

StringDrop["string", n]

Fornece a seqüência de caracteres "string" amputada dos n primeiros caracteres.

StringDrop["string", -n]

Fornece a seqüência de caracteres "string" amputada dos n últimos caracteres.

StringDrop["string", {n}]

Fornece a seqüência de caracteres "string" amputada do n-ésimo caracter.

StringDrop["string", {m, n}]

Fornece a seqüência de caracteres "string" amputada dos caracteres da m-ésima até a n-ésima posição.

StringForm["string₁`1`string₂`2`string₃...", expr₁, expr₂, ...]

Imprime a seqüência das seqüências string_i; inserindo as expressões expr_i nas posições marcadas por `i`.

StringInsert["string", "snew", n]

Gera uma seqüência string na qual é inserida a seqüência **snew** na n-ésima posição.

StringInsert["string", "snew", -n]

Gera uma seqüência string na qual é inserida a seqüência **snew** na n-ésima posição a partir do fim.

StringJoin["s₁", "s₂", ...]

ou

StringJoin[{"s₁", "s₂", ...}]

"s₁"<>"s₂"<>... Gera uma seqüência obtida pela concatenação das seqüências s_i.

StringLength["string"]

Fornece o número de caracteres da seqüência string.

StringMatchQ["string", "pattern"]

Fornece **True** se a seqüência string é equivalente à seqüência pattern.

StringPosition["string", "sub"]

Fornece a lista das posições dos caracteres de início e de fim da seqüência sub quando esta aparece na seqüência string.

StringReplace["string", "s"->"sp"]

Substitui, na seqüência string, as seqüências s por seqüências sp, em todos os lugares onde s aparece como subseqüência de string.

StringReplace["string", {"s₁"->"sp₁", "s₂"->"sp₂", ...}]

Substitui, na seqüência string, as seqüências s_i por seqüências sp_i, em todos os lugares onde as seqüências s_i aparecem como subseqüências de string.

StringReverse["string"]

Inverte a ordem dos caracteres da seqüência string.

StringSkeleton[n]

Representa a seqüência de n caracteres omitidos na impressão de uma seqüência impressa por **Short**.

StringTake["string", n]

Fornece a seqüência que contem os n primeiros caracteres da seqüência string.

StringTake["string", -n]

Fornece a seqüência que contem os n últimos caracteres da seqüência string.

StringTake["string", {n}]

Fornece a seqüência que contem o n-ésimo caracter da seqüência string.

StringTake["string", {m, n}]

Fornece a seqüência que contem os caracteres da seqüência string da posição m a posição n.

StringToStream["string"]

Abre um fluxo de entrada para a leitura da seqüência string.

Stub

Atributo que especifica se um símbolo já foi usado.

Subscript[expr]

Imprime a expressão expr em subscrito.

Subscripted[f[arg₁, arg₂,...]]

Imprime os argumentos arg_i como subscritos de f.

Subtract

$x - y$ é equivalente a $x + (-1 * y)$

SubtractForm

$x -= dx$ subtrai dx de x e devolve o novo valor de x.

Sum[f, {i, imin, imax}]

Avalia o somatório de f quando i varia de imin até imax.

Sum[f, {i, imin, imax}, {j, jmin, jmax}]

Avalia o somatório de f quando i varia de imin até imax e j de jmin até jmax

Superscript[expr]

Imprime a expressão expr em superscrito.

SurfaceColor[dcol]

Diretriz gráfica tridimensional que especifica que os polígonos que seguem devem agir como refletores de luz com uma cor especificada por dcol.

SurfaceGraphics[array]

Representação do gráfico tridimensional de uma superfície, com as alturas de cada ponto numa grade especificadas pelos valores contidos na tabela array.

SurfaceGraphics[array, shades]

Representação do gráfico tridimensional de uma superfície, com as alturas de cada ponto numa grade especificadas pelos valores contidos na tabela **array**; as partes sombreadas da superfície são especificadas na tabela **shades**.

Switch[expr, form₁, value₁, form₂, value₂,...]

Avalia a expressão **expr**; em seguida, compare o resultado achado com cada uma das expressões **form_i** sucessivamente; fornece o valor de **value_i** que corresponde à primeira equivalência entre **form_i** o resultado de **expr₁**.

Symbol

Cabeçalho associado a um símbolo.

SyntaxLength["string"]

Número de caracteres, começando no primeiro, de uma seqüência que corresponde a uma entrada sintaticamente correta para uma expressão única do *Mathematica*.

SyntaxQ["string"]

Fornece **True** se a seqüência corresponde a uma entrada sintaticamente correta para uma expressão única do *Mathematica*, e **False** no caso contrário.

Table[expr, {imax}]

Gera uma tabela (lista) de **imax** cópias de **expr**

Table[expr, {i, imin, imax, di}]

Gera uma tabela da expressão **expr**, função de **i** que varia de **imin** até **imax** variando por passos de **di**.

Table[expr, {i, imin, imax}, {j, jmin, jmax}]

Gera uma tabela da expressão **expr**, função de **i** e de **j**, que variam respectivamente entre os limites indicados.

TableAlignments

Opção para **TableForm** e **MatrixForm** que especifica como as entradas em cada dimensão devem ser alinhadas.

TableDepth

Opção para **TableForm** e **MatrixForm** que especifica o número máximo de níveis a serem impressos com o formato de tabela ou de matriz.

TableDirections

Opção para **TableForm** e **MatrixForm** que especifica se as dimensões sucessivas devem ser colocadas como linhas e colunas.

TableForm[list]

Imprime os elementos de uma tabela list como uma combinação de células retangulares

TableHeadings

Opção para **TableForm** e **MatrixForm** que especifica os textos a serem impressos como legendas em cada dimensão de uma tabela ou matriz.

TableSpacing

Opção para **TableForm** e **MatrixForm** que especifica quantos espaços devem ser deixados entre as linhas ou as colunas. Para as colunas, os espaços são enchidos por caracteres de espaço; para as linhas, os espaços são enchidos por linhas brancas.

TableSpacing -> {s₁, s₂,...} especifica que s_i espaços devem ser deixados na dimensão i.

TableSpacing -> Automatic gera espaçamento de {1, 3, 0, 1, 0, 1,...}.

TagSet

f/: lhs = rhs atribui rhs como valor de lhs e associa esta atribuição ao símbolo f.

TagSetDelayed

f/: lhs := rhs atribui rhs como valor adiado de lhs e associa esta atribuição ao símbolo f.

TagUnset

f/: lhs =. cancela qualquer regra definida anteriormente para lhs e associada ao símbolo f.

Take[list, n]

Fornece os n primeiros elementos da lista list.

Take[list, -n]

Fornece os n últimos elementos da lista list.

Take[list, {m, n}]

Fornece os elementos de m até n da lista list.

Tan[z]

Tangente, $\text{tg}(z)$ ou $\tan(z)$, do número complexo z.

Tanh[z]

Tangente hiperbólica, $\text{tanh}(z)$, do número complexo z.

Temporary

Atributo dos símbolos criados como variáveis locais por **Module**.

TensorRank[expr]

Indica a profundidade até a qual expr é uma tabela completa, com todos seus elementos num dado nível sendo listas do mesmo comprimento.

TeXFormText[expr]

Imprime a expressão expr como uma versão da linguagem $\text{T}_{\text{E}}\text{X}$.

Text[expr, coords]

Primitiva gráfica que representa um texto que corresponde à forma impressa de `expr`, centrada no ponto especificado pelas coordenadas `coords` (expressas seja por `{x, y, ...}` seja por `Scaled[{x, y, ...}]`).

TextForm[expr]

Imprime a forma textual da expressão `expr`.

Thickness[r]

Diretriz gráfica que indica a espessura `r` das linhas (expressas em frações da largura total do gráfico).

Thread[f[args]]

"Enfileira" `f` sobre qualquer lista que aparece nos argumentos `args`.

Exemplo:

`Thread[f[{a, b}, c, {d, e}]]` → `{f[a, c, b], f[b, c, e]}`

ThreeJSymbol[{j₁, m₁}, {j₂, m₂}, {j₃, m₃}]

Fornece os valores do símbolo 3-j de Wigner.

Through[p[f₁, f₂][x]]

Fornece `p[f1[x], f2[x]]`. **Through** distribui os operadores que aparecem dentro dos cabeçalhos das expressões.

Exemplo:

`Through[(f + g)[x, y]]` → `f[x, y] + g[x, y]`

Through[expr, h]

Realiza a transformação a cada vez que `h` aparece no cabeçalho de `expr`.

Throw[expr]

Sai das estruturas de controle de aninhamento, fornecendo o valor da expressão `expr` para o **Catch** de enquadramento mais próximo. **Throw** e **Catch** podem ser usados para implementar retornos não locais.

Ticks

Opção para funções gráficas que especifica as marcas ao longo dos eixos dos gráficos:

Ticks -> **None** (nenhuma marca)

Ticks -> **Automatic** (colocação automática)

Ticks -> {xticks, yticks,...}

TimeConstrained[expr, t]

Avalia expr e pára depois de t segundos.

Times

$x * y * z$ ou $x y z$ representa o produto dos termos x, y e z

TimesBy

$x *= c$ multiplica x por c e fornece o novo valor de x.

TimeUsed[]

Fornece o número total de segundos de tempo de CPU utilizado na sessão do *Mathematica* em andamento.

TimeZone[]

Fornece a zona horária fixada para o computador em uso.

Timing[expr]

Avalia expr e fornece a lista dos tempos usados assim com o resultado obtido.

ToCharacterCode["string"]

Fornece a lista dos códigos inteiros que correspondem aos caracteres contidos na seqüência "string".

ToDate[tine]

Converte uma data absoluta em segundos desde o 1º de janeiro de 1900 numa data no formato {ano, mês, dia, hora, minutos, segundos}.

ToExpression["string"]

Fornece a expressão obtida quando a seqüência string é tomada como entrada do *Mathematica*.

Together[expr]

Coloca os termos de uma soma acima de um denominador comum e cancela fatores no resultado.

Exemplo:

Together[1/x + 1/(1-x)] → 1/(1-x)x

ToHeldExpression["string"]

Fornece dentro de **Hold** a expressão obtida quando a seqüência string é tomada como entrada do *Mathematica*.

Exemplo:

ToHeldExpression["1 + 1"] → Hold[1 + 1]

TokenWords

Opção para **Read** e funções afins que fornece uma lista de palavras para usar como delimitador de palavras.

ToLowerCase[string]

Fornece a partir da seqüência string uma seqüência na qual todas as letras foram convertidas para minúsculas.

ToRules[eqns]

Toma as combinações lógicas das equações eqns, no formato gerado por **Roots** e **Reduce**, e as converte em listas de regras, no formato produzido por **Solve**.

Exemplo:

{ToRules[x==1 || x==2]} → {{x -> 1}, {x -> 2}}

ToString[expr]

Fornece uma seqüência que corresponde à forma impressa de expr.

TotalHeight

Opção para fluxos de saída que especifica o número máximo de linhas de textos que devem ser impressas para cada expressão de saída. Formas curtas das expressões são usadas se o número de caracteres necessários para imprimir a expressão inteira é grande demais.

TotalWidth

Opção para fluxos de saída que especifica o número máximo de caracteres de textos que devem ser impressas para cada expressão de saída. Formas curtas das expressões são usadas se o número de linhas necessárias para imprimir a expressão inteira é grande demais.

ToUpperCase

Fornece a partir da seqüência string uma seqüência na qual todas as letras foram convertidas para maiúsculas.

Trace[expr]

Gera a lista de todas a expressões usadas na avaliação de expr.

Trace[expr, form]

Gera a lista das a expressões usada na avaliação de expr que correspondem à forma form.

Trace[expr, s]

Gera a lista de todas a expressões que usam as regras de transformação associadas ao símbolo s.

TraceAbove

Opção para **Trace** e as funções associadas que especifica a inclusão ou não as cadeias de avaliação que contém a feição form procurada.

Exemplos: **TraceAbove->True**, **TraceAbove->All**, ...

TraceBackward

Opção para **Trace** e as funções associadas que especifica a inclusão ou não as expressões anteriores que contém a feição form procurada.

Exemplos: **TraceAbove->True**, **TraceAbove->All**,

TraceDepth

Opção para **Trace** e as funções associadas que especifica o nível de aninhamento das cadeias das expressões que devem ser incluídas.

TraceDialog[expr]

Começa um diálogo para cada expressão usada na avaliação de expr.

TraceDialog[expr, form]

Começa um diálogo somente para as expressões usadas na avaliação de expr e que tem a forma form.

TraceDialog[expr, s]

Começa um diálogo somente para as expressões usadas na avaliação de expr e que usam as regras de transformação associadas ao símbolo s.

TraceForward

Opção para **Trace** e as funções associadas que especifica se deve incluir ou não as expressões posteriores na corrente de avaliação que contem a forma form achada.

Exemplos: **TraceForward->True**, **TraceForward->All**.

TraceOff

Opção para **Trace** e as funções associadas que especifica as formas dentro das quais o detalhamento deve ser desativado.

TraceOn

Opção para **Trace** e as funções associadas que especifica quando o detalhamento deve ser ativado.

TraceOriginal

Opção para Trace e as funções associadas que especifica se deve-se testar a forma de cada expressão antes de avaliar seus cabeçalho e argumentos.

TracePrint[expr]

Mostra todas as expressões usadas na avaliação de expr.

TracePrint[expr, form]

Mostra apenas as expressões usadas na avaliação de expr e que combinam com form.

TraceScan[f, expr,...]

Aplica f a todas ou partes das expressões contidas em expr, a depender do que for especificado.

Transpose[list]

Realiza a transposição dos dois primeiros níveis em list

TreeForm[expr]

Imprime os níveis diferentes de expr mostrados a profundidades diferentes.

Trig

Opção para funções algébricas que especifica que as funções trigonométricas devem ser tratadas como funções racionais de exponenciais.

Trig->False

Opção padrão para todas as funções salvo **Simplify**. Com esta opção, as funções trigonométricas são tratadas como objetos indivisíveis nas operações algébricas.

Trig->True

As funções trigonométricas são tratadas como funções racionais de exponenciais nas operações algébricas.

True

Símbolo do valor booleano true (verdadeiro)

TrueQ[expr]

Fornece **True** se a expressão *expr* é **True**, e **False** no caso contrário.

Unequal

a **!=** *b* gera **False** se *a* e *b* são idênticos e **True** no caso contrário

Unevaluated[expr]

Forma não avaliada da expressão *expr* quando ela aparece como argumento de uma função

Uninstall[link]

Termina um programa externo começado por **Install** e cancela as definições do *Mathematica* que ele introduziu.

Union[list₁, list₂,...]

Gera uma lista ordenada que contém todos os elementos distintos presentes em qualquer uma das *list_i*

Unique[]

Gera um novo símbolo com o nome do tipo *\$nnn*.

Unique[x]

Gera um novo símbolo com o nome do tipo *x\$nnn*.

Unique[{x, y,...}]

Gera uma lista de novos símbolos.

Unique["xxx"]

Gera um novo símbolo com o nome do tipo *xxxnnn*.

Unprotect[s₁, s₂,...]

Cancela o atributo **Protected** dos símbolos *s_i*

UnsameQ

expr₁ **!=** *expr₂* fornece **True** se a expressão *expr₁* não é idêntica à expressão *expr₂*, e **False** no caso contrário

Unset

symb =. desativa qualquer regra definida para o símbolo **symb**

Update[symb]

Informa o *Mathematica* que mudanças escondidas já feitas podem afetar os valores associados ao símbolo **symb**

UpperCaseQ[str]

Fornece **True** se todos os caracteres da corrente **str** são letras maiúsculas (upper case)

UpSet

lhs ^= rhs faz **lhs** tomar o valor de **rhs** e associa esta atribuição aos símbolos que aparecem no nível 0 de **lhs**.

UpSetDelayed

lhs ^= rhs faz **lhs** tomar o valor adiado de **rhs** e associa esta atribuição aos símbolos que aparecem no nível 0 de **lhs**.

UpValues

Fornece a lista de regras de transformações que correspondem a todos os valores anteriores definidos para o símbolo **f**.

ValueQ[expr]

Vale **True** se um valor já foi definido para **exp**, e **False** no caso contrário.

Variables[pol]

Lista de todas as variáveis independentes no polinômio **pol**.

VectorQ[expr]

Vale **True** se **expr** é uma lista, nenhum de seus elementos sendo lista, e **False** no caso contrário.

ViewCenter

Opção para Graphics3D e SurfaceGraphics que indica as coordenadas x , y e z do ponto do espaço que deve aparecer no centro da área útil na imagem final (valores entre 0 e 1 que representam cada um o comprimento de um lado da caixa que limita a imagem).

ViewCenter->Automatic

Opção com a qual a caixa que limita a imagem é centrada na área da imagem final.

ViewCenter->>{1/2, 1/2, 1/2}

Opção com a qual o centro da caixa que limita a imagem é no centro da área da imagem final.

ViewPoint->{x, y, z}

Opção para Graphics3D e SurfaceGraphics que indica as coordenadas x , y e z do ponto do espaço a partir do qual os objetos desenhados são vistos.

ViewVertical

Opção para Graphics3D e SurfaceGraphics que indica qual é a direção das coordenadas que deve ser vertical na imagem final.

ViewVertical->{0, 0, 1}

Opção padrão para ViewVertical

WeierstrassP

Função elíptica de Weierstrass.

WeierstrassPPrime

Derivada da função elíptica de Weierstrass.

Which[test₁, valor₁, test₂, valor₂,...]

Avalia cada um dos test_{*i*} e devolve o objeto valor_{*i*} que segue o primeiro test_{*i*} cujo resultado é True.

While[test, body]

Avalia test e em seguida body enquanto que o resultado de test é True.

With[{x=x0, y=y0,...}, expr]

Especifica que os símbolos x, y, ... devem ser substituídos por x0, y0, ...

Word

Representa uma palavra nas funções Read, Find ...

WordSearch

Opção de Find e FindList que especifica se o texto procurado deve aparecer como uma palavra.

WordSearch->True

Com esta configuração da opção WordSearch, o texto deve aparecer como uma palavra limitada por separadores de palavra ou de registro.

WordSeparators

Opção de Read, Find e outras funções afins que indica a lista de dos caracteres ou correntes de caracteres que são usados como delimitadores de palavras

WordSeparators->{"", "\t"}

Opção padrão para WordSeparators

Workingprecision->n

Especifica que a precisão dos cálculos deve ser de n dígitos

**Write[ch, expr1,
expr2,...]**

Escreve as expressões expr_i (seguidas por um newline) em seqüência no canal de saída ch.

WriteString[ch, expr₁, expr₂,...]

Converte as expressões $expr_i$ em correntes (strings) e as escreve em seqüência no canal de saída ch.

Xor[e₁, e₂,...]

Função lógica XOR ou XOU (OR ou OU exclusivo)

ZeroTest

Opção para **LinearSolve** e outras funções lineares algébricas, aplicando-a a elementos de matrizes para determinar se estes devem ou não ser considerados como nulos.

Zeta[s]

Função zeta de Riemann $\zeta(s)$

Zeta[z,a]

Função zeta generalizada de Riemann $\zeta(s,a)$

\$Aborted

Símbolo especial que aparece como resultado de um cálculo que foi interrompido.

\$BatchInput

É **True** se a entrada da sessão corrida é feita diretamente para o kernel em modo de lote (batch).

\$BatchOutput

É **True** se a saída da sessão corrida efetua-se em modo de lote (batch), de maneira conveniente para a leitura por outros programas.

\$CommandLine

Lista de seqüências que indica os elementos da linha de comando do sistema operacional pela qual o *Mathematica* foi chamado.

\$Context

Variável global que indica o contexto atual.

\$ContextPath

Variável global que indica a lista dos contextos, depois de **\$Context**, para tentar achar um símbolo que já foi usado. cada contexto é especificado por uma seqüência do tipo "name".

\$CreationDate

Fornece a data e hora de criação do kernel do *Mathematica* em uso.

\$DefaultFont

Indica a fonte padrão usada para os textos nos gráficos.

\$Display

Fornece a lista dos arquivos e canal (pipe) usados com **\$DisplayFunction** padrão.

\$DisplayFunction

Fornece a configuração padrão para a opção **DisplayFunction** das funções gráficas.

\$DumpDate

Fornece a lista de todas as datas e horas nas quais **Dump** foi utilizado para criar o sistema de *Mathematica* em uso.

\$DumpSupported

É True se Dump pode ser usado com a versão do *Mathematica* em uso.

\$Echo

Fornece a lista dos arquivos e canal (pipe) usados com **\$DisplayFunction** padrão.

\$Epilog

Símbolo, cujo valor, se existe, é avaliado quando uma sessão de *Mathematica* ou um diálogo é encerrado.

\$Failed

Símbolo especial retornado por certas funções quando elas não podem fazer o que elas deveriam.

\$IgnoreEOF

Especifica se o *Mathematica* deve terminar quando ele recebe um caracter fim-de-arquivo (end-of-file - EOF) na entrada.

\$Input

Variável global cujo valor é o nome do fluxo cuja entrada no *Mathematica* é procurada.

\$Inspector

Variável global que fornece a função a ser aplicada quando o inspetor é chamado a partir de um menu de interrupção.

\$IterationLimit

Fornece o comprimento maior da corrente de avaliação usada na tentativa de avaliar uma expressão.

\$Language

Lista das seqüências que dão os nomes das línguas usadas para as mensagens. Os nomes de todas as línguas são dados em inglês e a primeira letra deve ser maiúscula como, por exemplo, em "**French**".

\$Letters

Lista dos caracteres que devem ser tratados como letras. As formas minúsculas e maiúsculas de uma letra particular podem ser dadas em uma sub-lista.

Valor padrão para **\$Letters**:

```
{{"a", "A"}, {"b", "B"}, ..., {"z", "Z"}}
```

\$Line

Variável global que especifica o número da linha atual de entrada.

\$Linked

É **True** se o kernel do *Mathematica* é rodado através do MathLink.

\$LinkSupported

É **True** se MathLink pode ser usado na versão ativa do *Mathematica*, e **False** no caso contrário.

\$MachineEpsilon

Fornece o menor número com a precisão da máquina que pode ser adicionado a 1.0 para dar um resultado que não é igual a 1.0.

\$MachineID

Seqüência que fornece, se possível, um código único de identificação para o computador em uso.

\$MachineName

Seqüência que fornece, se possível, o nome atribuído ao o computador em uso, se o tal nome foi definido.

\$MachinePrecision

Fornece o número de casas decimais de precisão usadas para os números com precisão da máquina.

\$MachineType

Seqüência que fornece o tipo geral do computador no qual o *Mathematica* é carregado.

\$MaxMachineNumber

Fornece o maior número com precisão da máquina que pode ser usado num sistema dado.

\$MessageList

Variável global que fornece a lista dos nomes das mensagens geradas durante a avaliação da linha de entrada corrida.

\$MessagePrePrint

Variável global cujo valor, se determinado, é aplicado às expressões antes que elas sejam incluídas no texto das mensagens.

\$Messages

Forne a lista dos arquivos e canal (pipe) aos quais a mensagem de saída é mandada.

\$MinMachineNumber

Fornece o menor número positivo com precisão da máquina que pode ser usado num sistema dado.

\$ModuleNumber

Fornece o número de série atual que é usado para as variáveis locais que estão sendo criadas.

\$NewMessage

Variável global que, se determinada, é aplicado ao nome e etiqueta das mensagens solicitadas e que ainda não foram ainda definidas.

\$NewSymbol

Variável global que, se determinada, é aplicado ao nome e contexto de cada símbolo novo que cria o *Mathematica*.

\$Notebooks

É *True* se o *Mathematica* é usado com um terminal da frente (front end) específico para microcomputador portátil (notebook).

\$OperatingSystem

Seqüência de caracteres que indica o tipo de sistema operacional no qual o *Mathematica* é rodado.

\$Output

Lista de arquivos e canais para os quais a saída padrão do *Mathematica* é mandada.

\$Packages

Fornece a lista dos contextos que correspondem a todos os pacotes foram carregados na sessão ativa do *Mathematica*.

\$Path

Lista dos diretórios a serem investigados na tentativa de achar um arquivo externo.

\$PipeSupported

É **True** se canais devem ser usados na versão ativa do *Mathematica*.

\$Post

Variável global cujo valor, se determinado, é aplicado à cada expressão de saída.

\$Pre

Variável global cujo valor, se determinado, é aplicado à cada expressão de entrada.

\$Preprint

Variável global cujo valor, se determinado, é aplicado à cada expressão antes de sua impressão.

\$PreRead

Variável global cujo valor, se determinado, é aplicado ao valor de cada entrada antes que seja injetada no *Mathematica*.

\$RecursionLimit

Limite habitual do número de níveis de recursos que o *Mathematica* pode usar.

\$ReleaseNumber

Inteiro que indica o número da versão do kernel do *Mathematica* em uso.

\$SessionID

Número determinado para ser único para cada sessão do *Mathematica*.

\$SoundDisplayFunction

Configuração padrão para a opção **DisplayFunction** nas funções de som.

\$StringConversion

Configuração padrão para a opção **StringConversion** nas funções de saída.

\$StringOrder

Fornece a ordem dos caracteres para ser usada ao ordenar seqüências e nomes de símbolos.

Valor padrão para **\$StringOrder**:

`{{"a","A"}, {"b","B"}, ..., {"z","Z"}, "0", "1", "2", ..., "9"}`

\$SyntaxHandler

Variável global que, se determinada, é aplicada a cada seqüência de entrada na qual foi achado um erro de sintaxe.

\$System

Seqüência que especifica o tipo de sistema de computador usado.

\$TimeUnit

Intervalo de tempo mínimo em segundos registrado pelo sistema de computador usado.

\$Urgent

Fornece a lista dos arquivos e canais nos quais saídas urgentes do *Mathematica* são mandadas.

\$Version

Seqüência que representa a versão do *Mathematica* em uso.

\$VersionNumber

Número real que indica o número da versão do kernel do *Mathematica* em uso e que cresce a cada nova versão.

APÊNDICE A

LISTA DE NOTEBOOKS

| | |
|-------------|---|
| CIRCLE.MA | Plano inclinado limitado por círculo |
| SPHERE.MA | Esfera |
| COLOR.MA | Gráficos coloridos |
| FITSIN.MA | Ajuste senoidal |
| FITEXP.MA | Ajuste exponencial |
| FOURIER.MA | Fourier (Transformação) |
| PERIODOG.MA | Periodograma |
| SPEEDING.MA | Testes de otimização do tempo de execução |

CÍRCULO [CIRCLE.MA]

```
(* Plano inclinado *)  
ParametricPlot3D[{u, t, u},  
{t,-1,1},{u,0,1}, PlotPoints->33]
```

```
(* Região semicircular num plano inclinado *)  
ParametricPlot3D[{If[t^2+u^2<1, u,False],  
If[t^2+u^2<1, t,False],  
If[t^2+u^2<1, u,False]},  
{t,-1,1},{u,0,1}, PlotPoints->20]
```

```
(* Leque *)  
ParametricPlot3D[{  
If[t^2+u^2<1, u + 0.1 Sin[30 t],False],  
If[t^2+u^2<1, t,False],  
If[t^2+u^2<1, u ,False]},  
{t,-1,1},{u,0,1}, PlotPoints->20]
```

```
(* ***** *)
```

ESFERA [SPHERE.MA]

```
ParametricPlot3D[{  
Cos[t] Cos[u], Cos[t] Sin[u], Sin[t]},  
{t,-1.57,1.57},{u,0,6.28},  
Boxed->False, Axes->False]
```

```
(* ***** *)
```

GRÁFICOS COLORIDOS [COLOR.MA]

(* A seguir são dados exemplos de gráficos coloridos apresentados em células separadas. *)

(* Curvas de cores diferentes *)

```
Plot[{2-x,x^2-2,2 Sin[x]},{x,-5,5},  
PlotStyle->{{RGBColor[1,0,0]},{RGBColor[0,1,0]},  
{RGBColor[0,0,1]}}
```

(* Curvas de cores diferentes usando cores "compostas" com RGBColor[vermelho, verde, azul] cada entre 0 e 1 *)

```
Plot[{2-x,x^2-2,2 Sin[x]},{x,-5,5},  
PlotStyle->{{RGBColor[0.2,0.4,0.6]},  
{RGBColor[0.1,0.7,0.3]},{RGBColor[0.7,0.1,0.4]}}
```

(* Curvas de espessuras e cores diferentes Hue[matiz, saturação, brilho] cada entre 0 e 1 *)

```
Plot[{2-x,x^2-2,2 Sin[x]},{x,-5,5},  
PlotStyle->{{Hue[0.5,0.5,0.5],Thickness[0.03]},  
{Hue[0.1,0.5,0.8],Thickness[0.02]},  
{Hue[0.8,0.5,0.1],Thickness[0.01]}}
```

(* Gráfico de 3 dimensões *)

```
t3=Table[Mod[x,y],{x,1,20},{y,1,20}];  
ListPlot3D[t3]
```

(* Gráfico/mapa de contorno com texto nos eixos *)

```
ap=Table[j,{j,1,10}];  
t3={ap,ap,ap};  
ListContourPlot[t3,Frame->True,  
FrameLabel->{"Tempo(seg)","Período(seg)"},  
ColorFunction->Hue]
```

(* Gráfico/mapa de contorno na versão branco/cinza/preto *)

```
ContourPlot[x+y,{x,0,1},{y,0,1}]
```

```
(* Gráfico/mapa de contorno branco/cinza/preto *)
ap=Table[0.1 j,{j,1,10}];
t3={ap,ap,ap};
Show[Graphics[Raster[t3] ] ]
```

```
(* Gráfico/mapa de contorno com matiz padrão
simples Hue *)
ContourPlot[x y,
{x,0,1}, {y,0,1},ColorFunction->Hue ]
```

```
(* Gráfico/mapa de contorno com matiz (Hue[#]&) onde & indica que
trata-se de uma função "pura" do argumento "puro" (chamado "slot")
#, este último símbolo ocupando o papel habitual de x; por exemplo
3 #^2 + 4 # & ou seja (3 #^2 + 4 #)& representaria a função 3 x^2 +
4 x; No exemplo que segue, a variável pura # representa a variável
z = x y . *)
ContourPlot[x y, {x,0,1}, {y,0,1},
ColorFunction->(Hue[#]&)]
```

```
(* Gráfico/mapa de contorno com função de matiz Hue[matiz,
saturação, brilho] cada entre 0 e 1, com efeitos compostos gerando
um gráfico que passa estranhamente do preto, cinza e marrom a
colorido *)
ContourPlot[x y, {x,0,1}, {y,0,1},
ColorFunction->(Hue[#,0.3,1-#]&)]
```

```
(* Gráfico/mapa de contorno com função de cor
(ColorFunction[vermelho, verde, azul]&) cada entre 0 e 1 com
efeitos compostos podendo gerar uma escala de cores escolhida
pelo usuário. Observar no exemplo abaixo que a função do
vermelho cresce linearmente de 0 a 1; a do verde é parabólica,
cresce de 0 a 1 obtido para z=0.5 e decresce até 0 para z-1; a do
azul decresce de 1 a 0. Esta escolha foi feita para se ter o domínio
sucessivo das cores azul, verde e vermelho quando z passa de 0 a
1. *)
ContourPlot[x y, {x,0,1}, {y,0,1},
ColorFunction->(RGBColor[#,1- 4 (#-0.5)^2,1-#]&)]
```

(* Gráfico de contorno que utiliza ListContourPlot[] no lugar de ListPlot[]. Neste caso, a variável "slot" # representa o conteúdo de cada casa da tabela que é o argumento de ListContourPlot[] *)

```
t=Table[C+L,{L,0,9},{C,0,9}]
ListContourPlot[t,Frame->True,
FrameLabel->{"Tempo(seg)","Periodo(seg)"},
ColorFunction->(RGBColor[#,1-4(#-0.5)^2,1-#]&)]
```

(* Gráfico de contorno que utiliza também ListContourPlot[]. Neste caso, a tabela t contém um lista dos números de 0 a 99 distribuídos em 10 linhas de 10 números. Levando em conta que a função (ColorFunction[vermelho, verde, azul]&) constroi uma série de 11 cores entre os valores extremos da variável #, entende-se porque as cores são as mesmas do que no exemplo anterior e, então, apenas sua distribuição "geográfica" muda. *)

```
t=Table[C+10 L,{L,0,9},{C,0,9}]
ListContourPlot[t,Frame->True,
ColorFunction->(RGBColor[#,1-4(#-0.5)^2,1-#]&)]
```

(* ***** *)

AJUSTE DE DADOS A UMA FUNÇÃO SENOIDAL [FITSIN.MA]

Para analisar melhor o funcionamento deste programa, e permitir sua adaptação a problemas específicos, segue uma versão "seqüenciável", isto é uma versão que pode ser executada de uma vez só e que pode também ser fracionada em elementos para a execução elemento por elemento. A fracionamento pode ser efetuado seja dividindo a célula em várias células (Cell, Divide Cell ou <Ctrl><D>) seja tornando inativos segmentos da célula enquadrando-os por "(" e ")" (sem as aspas).

```
oname="FS01.ASC"
s=:s= Table[ N[Sin[2 Pi t/10 + 1] +
4 Sin[2 Pi t/30 + 3] ], {t,1,120}];
len=Length[s]; omegamax= N[Pi];
mean=Sum[s[[i]],{i,1,len}]/len; s-=mean;
Put["FITSIN02=TSIRA in MATH4WIN",oname];

For[omega=0.2; domega=N[2 Pi/len,3], omega<omegamax,
omega+=domega,
{ {a1,a2,a3}= {1, omega, 0 };
f= Compile[{x,a1,a2,a3},- a1 Sin[ a2 x +a3 ] ];
g= Compile[{x,a1,a2,a3},- Sin[ a2 x +a3 ] ];
h= Compile[{x,a1,a2,a3},- a1 x Cos[ a2 x +a3 ] ];
m= Compile[{x,a1,a2,a3},- a1 Cos[ a2 x +a3 ] ];

For[ite=1; d=1, ite<51 && 10^-9<d<10^9 , ite++,
{r=0; c=Table[0,{3},{3}]; v=Table[0,{3}];

For [ t=1, t<=len, t++,
{f0=s[[t]]+f[t,a1,a2,a3];
g1=g[t,a1,a2,a3];h1=h[t,a1,a2,a3];m1=m[t,a1,a2,a3];
gh=g1 h1; gm=g1 m1; hm=h1 m1;
r+= f0 f0;
c+= {{g1 g1, gh, gm},{gh, h1 h1,hm},{gm,hm,m1 m1}};
v+= {f0 g1, f0 h1, f0 m1}; } ];

e= Inverse[c]; b=e.v; {a1,a2,a3}=b;
If[a1<0, a1*=-1 ; a3+=N[Pi] ];
```

```

If[Abs[a2- omega]>0.5 domega, a2=omega];
a3=N[Mod[a3, 2 Pi] ];

d=N[b.b, 3]; aa=N[{a1,a2,a3},3];
da= N[Table[Sqrt[r*e[[i,i]]/(len-2)], {i,1,3}], 3];

Print[omega, " ",aa, " ", ite];
If[d<10^-9, Print[aa, da," ", d];
PutAppend[aa, da, d, oname] ]

}]
}]
!!FS01.ASC

```

(* ***** *)

AJUSTE DE DADOS A UMA FUNÇÃO EXPONENCIAL SOMADA A UMA CONSTANTE [FITEXP.MA]

```

iname=Input["Input file name (between " "):"];
oname=Input["Output file name (between " "):"];
a=Input["Input initial values {a1, a2, a3}:"];
s=ReadList[iname,{Number,Number}]; length=Length[s];
u=Transpose[s];
e=da=a;
f[x_] := -a[[1]] - a[[2]] Exp[a[[3]] x];
df[x_] :={-1,- Exp[a[[3]] x], -a[[2]] x Exp[a[[3]] x]};
d=1.;
For[ite=1, ite<51 && 10^-6<d, ite++,
{c=Table[0.,{i,1,3},{j,1,3}]; v={0.,0.,0.}; r=0.;
For [ k=1, k<=length, k++,
{t=u[[1,k]]; f0=u[[2,k]]+f[t]; r+=f0*f0;
c+=Table[df[t][[i]]*df[t][[j]], {i,1,3},{j,1,3}];
v+=Table[df[t][[i]]*f0, {i,1,3} ] };
e=Inverse[c]; b=e.v; a=b; d=N[Sum[b[[i]]^2, {i,1,3}],4];
Do[da[[i]]=N[Sqrt[r*e[[i,i]]/(length-2)],4], {i,1,3}];
a=N[a,4];
Print[h," ", ite," ", N[a,4]," ", da," ", d] };
Save[oname,iname,oname,h,p,dp,a,da,d,ite] /; d<10^-6 ]

```

O programa acima, como a anterior, é baseado no método de Wolberg (J.R. Wolberg, Prediction Analysis, Van Nostrand, 1967) e é fácil de se entender. Ele foi concebido como um programa ("package" wexp.m) dentro do *Mathematica* for DOS. A conversão para o *Mathematica* for Windows não apresenta nenhuma dificuldade.

Para analisar melhor o funcionamento deste programa, e permitir sua adaptação a problemas específicos, segue uma versão "seqüenciável" do mesmo, isto é, uma versão que pode ser executada de uma vez só e que pode também ser fracionada em elementos para a execução elemento por elemento. A fracionamento pode ser efetuado seja dividindo a célula em várias células (Cell, Divide Cell ou <Ctrl><D>) seja tornando inativos segmentos da célula enquadrando-os por (* e *).

```

a={-3.6, -7.3, -0.7};
s={ {0,10},{1,7},{2,5.5},{3,4.5},{4,4},{5,3.8},{6,3.7},
{7,3.71},{8,3.69},{9,3.7} };
u=Transpose[s]; length=Length[s];
e=da=a;
f[x_] := -a[[1]] - a[[2]] Exp[a[[3]] x];
df[x_] := {-1,- Exp[a[[3]] x], -a[[2]] x Exp[a[[3]] x]};
d=1.;

For[ite=1, ite<51 && 10^-6<d , ite++,
{c=Table[0.,{i,1,3},{j,1,3}]; v={0.,0.,0.}; r=0.;
For [ k=1, k<=length, k++,
{t=u[[1,k]]; f0=u[[2,k]]+f[t]; r+=f0*f0;
c+=Table[df[t][[i]]*df[t][[j]], {i,1,3},{j,1,3}];
v+=Table[df[t][[i]]*f0, {i,1,3} ] };
e=Inverse[c]; b=e.v; a=b; d=N[Sum[b[[i]]^2, {i,1,3}],3];
Do[da[[i]]=N[Sqrt[r*e[[i,i]]/(length-2)],2],{i,1,3}];
a=N[a,3];
Print[ite, a, da, d,"\\n\\r"} ]

(* ***** *)

```

PERIODOGRAMA [PERIDOG.MA]

```
hmin=1; hmax=5; d=1; a=Table[0.,{hmax}];
s=ReadList["sunspot.prn",{Number,Number}]; le=Length[s];
g[x_]=Fit[s, {1,x, x^2},x];
s = Transpose[s]; s[[2]]-=g[s[[1]]];
s = Transpose[s];

For [ h = 1, h <= hmax, h++,
{ pu = N[2 Pi/(h d)];
b[x_]=Fit[s, {Sin[pu x], Cos[pu x]}, x];
a[[h]]=N[Sqrt[b[0.]^2 + b[0.5 Pi/pu]^2], 4 ]; } ];
ca=ColumnForm[a]; Save["sunspot.per",title,hmin,hmax,d,ca];
p=ListPlot[a];
```

```
(* ***** *)
```

AUMENTO DA VELOCIDADE DE CÁLCULO: TESTES DE TIPOS DE FUNÇÕES [SPEEDING.MA]

```
(* Função f[x_] := ... *)
ClearAll[x,f,g];
f[x_] := 8 x Sin[0.5 x + 3] Exp[-0.5 x] + 2 x^-4 + x^-7;
g=Table[N[f[7],3], {1000}];
g[[1]]
(* Tempo = 3.41 até 3.52 seg. para 1000 valores
num 486DX4-100-16 *)
0.365
```

```
(*      Função f = Function[{x}, ... ] *)
ClearAll[x,f,g];
f = Function[{x}, 8 x Sin[0.5 x + 3] Exp[-0.5 x] + 2 x^-4 + x^-7];
g=Table[N[f[7],3], {1000}];
g[[1]]
(* Tempo = 3.52 até 3.63 seg. para 1000 valores
num 486DX4-100-16 *)
0.365
```

```
(*      Função f = Compile[{x}, ... ] *)
ClearAll[x,f,g];
f = Compile[{x}, 8 x Sin[0.5 x + 3] Exp[-0.5 x] + 2 x^-4 + x^-7];
g=Table[N[f[7],3], {1000}];
g[[1]]
(* Tempo = 0.71 até 1.04 seg. para 1000 valores
num 486DX4-100-16 *)
0.365
```

```
(*      Função com memória f[x_] := f[x] = ... *)
ClearAll[x,f,g];
f[x_] := f[x] = 8 x Sin[0.5 x + 3] Exp[-0.5 x] + 2 x^-4 + x^-7;
g=Table[N[f[7],3], {1000}];
g[[1]]
(* Tempo = 0.55 até 0.77 seg. para 1000 valores
num 486DX4-100-16 *)
0.365
```

APÊNDICE B

MICRODICIONÁRIO

| | |
|----------------------|--|
| Algebraic expression | Expressão algébrica |
| Argument | Argumento |
| Array | Tabela |
| Atom | Átomo |
| Attribute | Atributo |
| Batch | Lote |
| Bounding box | Caixa na qual fica um gráfico de três dimensões |
| Cell | Célula |
| Channel | Canal (por exemplo: de saída) |
| Clipboard | Espaço de memória para transferência rápida |
| Column | Coluna (de uma matriz) |
| Compiled language | Linguagem compilada |
| Complex | Complexo |
| Concatenation | Concatenação (união de seqüências em uma seqüência única) |
| Context | Contexto, ambiente |
| Contour | Contorno |
| Contour plot | Mapa de contorno |
| Copy | Copiar (uso do clipboard) |
| Current | Atual |
| Customized | Configurado para o (ou pelo) usuário |
| Cut | Cortar, Retirar, Deslocar, Mover (uso do clipboard) |
| Default | Configuração padrão (antes da configuração feita pelo usuário) |
| Depth | Profundidade |
| Decimal digit | Casa decimal, dígito decimal |
| Delayed | Adiado, retardado |
| Digit | Dígitos |
| Display | Mostrador |
| Display (to) | Mostrar |

| | |
|----------------------|---|
| Directory | Directório |
| Element | Elemento |
| End-of-file (EOF) | Fim-de-arquivo (EOF) |
| Enter | Entrar (dados), equivalente a Retorno de linha (Return) |
| Expression | Expressão |
| False | Falso |
| File | Arquivo |
| File name | Nome de arquivo |
| Fit | Ajuste |
| Floating point | Ponto flutuante, isto é, vírgula (decimal) flutuante |
| Font | Fonte, conjunto de caracteres de um tipo |
| Fractional | Fracionário |
| Frame | Moldura, quadro |
| Front end | Terminal da frente, interface <i>Mathematica</i> -usuário (com o uso do monitor, do teclado, dos arquivos etc.) |
| Function | Função (uso irrestrito) |
| Grid lines | Linhas de grade |
| Handle | Alça, puxadeira (nos lados e cantos das imagens, para alterar suas dimensões) |
| Head | Cabeçalho, início |
| Height | Altura |
| High level language | Linguagem de alto nível |
| Hue | Matiz |
| Imaginary | Imaginário |
| Integer | Inteiro |
| Interpreted language | Linguagem interpretada |
| Interrupt | Interrupção |
| Kernel | Tradução literal: Semente, Âmago; Sentido aproximado: Núcleo para os cálculos; Motor (palavra usada para certos sistemas operacionais); kernel (mais usado) |
| Label | Marca, nome, etiqueta, endereço, legenda (de um gráfico) |

| | |
|--------------------------|---|
| Level | Nível |
| Line feed | Avanço de linha |
| Load (to) | Carregar (um arquivo a partir do disco rígido, por exemplo) |
| Loop | Laço (A palavra "loop" é muito usada também em português) |
| Lower case | Minúsculas, caixa baixa (letras) |
| Match (to) | Ser equivalente, equiparar, combinar |
| Matrix | Matriz |
| Mesh | Malha |
| Name | Nome |
| Newline | Retorno de linha |
| Notebook | Livro de apontamentos; Pacote |
| Object | Objeto |
| Object oriented language | Linguagem orientada para objeto |
| Package | Pacote |
| Pad (to) | Colocar de cada lado (de um símbolo...), acolchoar |
| Padding | Borda(s) |
| Parser | Analisador de sintaxe |
| Paste | Colocar (com o cursor no lugar de destino, no uso do clipboard) |
| Pattern | Formato |
| Pipe | Tubo, canal (de comunicação) |
| Plot | Desenho, gráfico |
| Pointer | Apontador ou cursor |
| Primitive | Primitiva (pode ter o sentido de objeto já preexistente no <i>Mathematica</i>) |
| Procedure | Procedimento |
| Prompt | Mensagem na tela |
| Protected | Protegido (isto é, inalterável) |
| Random | Aleatório |
| Range | Domínio |
| Raster | Imagem de varredura de linhas (trama) |
| Rational | Racional |
| Real | Real |
| Record | Registro, relação |
| Release | Versão |
| Render (to) | Montar, preparar |

| | |
|-------------|---|
| Return | Retorno de linha (equivalente a Enter); Retorno (de loop, de sub-rotina) |
| Return (to) | Retornar, devolver |
| Round | Arredondado |
| Row | Linha (de uma matriz) |
| Run (to) | Rodar, executar |
| Scanner | Dispositivo de captura de imagens por varredura |
| Set (to) | Determinar |
| Setting | Reglagem; configuração, definição ou parâmetro (de opção) |
| Shade | Sombra |
| Skip (to) | Pular, omitir |
| Sorted | Ordenada (lista) |
| Sound | Som |
| Source | 1. Primeira fase de um programa antes de interpretado ou compilado 2. Fabricante/Fornecedor |
| Space | Espaço |
| Splice (to) | Emendar |
| Stack | Pilha (registros de memória) |
| Store | Armazenar |
| Stream | Corrente, fluxo. |
| String | Seqüência, fila (de caracteres) |
| Syntax | Sintaxe |
| Table | Tabela |
| Tag | Etiqueta, marca, nome |
| Thread (to) | Enfileirar |
| Ticks | Marcas de referência ao longo dos eixos nos gráficos |
| True | Verdadeiro |
| Upper case | Maiúsculas, caixa alta (letras) |
| Upvalue | Valor acima (anterior) |
| Value | Valor |
| View point | Ponto de vista (isto é, coordenadas do olho que observa um gráfico de três dimensões) |
| Word | Palavra |
| Wrap (to) | Empacotar |

LIVROS

Abell M. L. and Braselton J. P. *Mathematica by Example*, Revised Edition. Academic Press, 1994.

Abell M. L. and Braselton J. P. *The Mathematica Handbook*. Academic Press, 1992.

Abell M. L. and Braselton J. P. *Differential Equations with Mathematica*. Academic Press, 1993.

Allen A. O. *Introduction to Computer Performance Analysis with Mathematica*. Academic Press, 1994.

Blachman N. *Mathematica: A Practical Approach*. Prentice-Hall, 1991.

Blachman N. *Mathematica Quick Reference, Version 2*. Addison-Wesley, 1992.

Braden B., Krug D. K., McCartney P. W. and Wilkinson S. *Discovering Calculus with Mathematica*. J. Wiley.

Brown D. T., Davis B., Porta H. and Uhl J. J. *Calculus & Mathematica, Part I (Courseware)*. Addison-Wesley, 1991.

Burbulla D. C. M. and Dodson C. t. J. *Self-tutor for Computer Calculus Using Mathematica*. Prentice-Hall.

Crandall R. E. *Mathematica for the Sciences*. Addison-Wesley, 1991.

Crandall R. E. *Projects in Scientific Computation*. Telos/Springer-Verlag, 1994.

Crooke P. and Ratcliffe J. *A Guidebook to Calculus with Mathematica*. Wadsworth Publishing, 1991.

Davis W., Manton J., Porta H. and Uhl J. *Calculus & Mathematica*. Addison-Wesley, 1994.

Ellis W. and Lodi E. *A Tutorial Introduction to Mathematica*. Brooks/Cole, 1990.

Emert J. and Nelson R. *Calculus & Mathematica*. Saunders College Publishing.

Feagin J. M. *Quantum Methods with Mathematica*. Telos/Springer-Verlag, 1994.

Finch J. K. and Lehmann M. *Exploring Calculus with Mathematica*. Addison-Wesley, 1992.

Freeman J. A. *Simulating Neural Networks with Mathematica*. Addison-Wesley, 1994.

Gaylord R. J., Kamin S. N. and Wellin P. R. *Introduction to Programming with Mathematica*. Telos/Springer-Verlag, 1993.

Gray A. *Modern Differential Geometry of Curves and surfaces with Mathematica*. CRC Press.

Gray J. W. *Mastering Mathematica: Programming Methods and Applications*. Academic Press, 1994.

Gray T. and Glynn J. *The Beginner's Guide to Mathematica Version 2*. Addison-Wesley, 1992.

Gray T. and Glynn J. *Exploring Mathematics with Mathematica*. Addison-Wesley, 1991.

Höft M. H. *Laboratory for Calculus I Using Mathematica*. Addison-Wesley, 1992.

Maeder R. E. *Programming in Mathematica, Second Edition*. Addison-Wesley, 1991.

Maeder R. E. Programming in *Mathematica*. Second Edition, Addison-Wesley, 1994.

Maeder R. E. The *Mathematica* Programmer. Academic Press, 1994.

Packel E. and Wagon S. Animating Calculus: *Mathematica* Notebooks for the Laboratory. W. H. Freeman, 1994.

Parker L. and Christensen S. M. *MathTensor*. A System for Doing Tensor Analysis by Computer. Addison-Wesley, 1994.

Riddle A. and Dick S. Applied Electronic Engineering with *Mathematica*. Addison-Wesley, 1994.

Seggern D. von. CRC Standard Curves (including *Mathematica* notebooks). CRC Press.

Shaw W. T. and Tig J. Applied *Mathematica*: Getting Started, Getting It Done. Addison-Wesley, 1994.

Shuchat A. and Shultz F. The Joy of *Mathematica*: A Point-and-Click Way to Use and Learn *Mathematica*. Addison-Wesley, 1994.

Skeel R. D. and Keiper J. B. Elementary Numerical Calculus with *Mathematica*. McGraw-Hill, 1993.

Skiena S. Implementing Discrete Mathematics: Combinatorics and Graph Theory with *Mathematica*. Addison-Wesley, 1990.

Smith C. The *Mathematica* Graphics Guidebook. Addison-Wesley, 1991.

Sparks A. G., Davenport J. W. and Braselton J. P. Calculus Labs using *Mathematica*. Harpers Collins College Publishers.

Stroyan K. D. Calculus Using *Mathematica*. Academic Press, 1993.

Vardi I. Computational Recreations in *Mathematica*. Addison-Wesley, 1991.

Variable Symbols, Inc., *Mathematica*: Quick Reference with usage statements from *Mathematica* Version 1.2. Variable Symbols, Inc., 1990.

Varian H. R. Economic and Financial Modeling with *Mathematica*. Telos/Springer-Verlag, 1993.

Vvedensky D. Partial Differential Equations with *Mathematica*. Addison-Wesley, 1993.

Wagon S. *Mathematica* in Action. W. H. Freeman, 1991.

Wickham-Jones T. *Mathematica* Graphics: Techniques & Applications. Telos/Springer-Verlag, 1994.

Wolff R. S. and Yaeger L. Visualization of Natural Phenomena. Telos/Springer-Verlag, 1993.

Wolfram Research. Elementary Tutorial Notes. Wolfram Research, Inc. *, 1991.

Wolfram Research. Intermediate Tutorial Notes. Wolfram Research, Inc. *, 1991.

Wolfram Research. Advanced Tutorial Notes. Wolfram Research, Inc. *, 1991.

Wolfram Research (P. Boyland, A. Chandra, J. Keiper, E. Martin, J. Novak, M. Petkovsek, S. Skiena, I. Vardi, A. Wenzlow, T. Wickham-Jones, D. Withoff, and others). Guide to Standard *Mathematica* Packages, Version 2.2. *Mathematica* Technical Report. Wolfram Research, Inc. *, 3rd Ed, 1993.

Wolfram Research. *Mathematica*, A System for Doing Mathematics by Computers. Version 2.2. User's Guide for Microsoft Windows. Wolfram Research, Inc. *, 1993.

Wolfram S. *Mathematica*, A System for Doing Mathematics by Computers. Addison-Wesley, 1a Ed. 1988; 2a Ed. 1991.

Wolfram S. *Mathematica* Reference Guide for *Mathematica* Version 2. Addison-Wesley, 1991.

REVISTAS

Mathematica in Education (Quarterly). P. R. Wellin, Editor in Chief, Telos/Springer-Verlag

Mathematica World (Monthly Electronic Notebook Magazine on 3.5" diskette) Hunt S. M. Editor, Osmond College, The University of Melbourne.

The *Mathematica* Journal (Quarterly). Miller Freeman, Inc.

* Wolfram Research, Inc., 100 Trace Center Drive, Champaign, IL 61820-7237. Telephone: 001-217-396-0700; Fax: 001-217-396-0747; Email: info@wri.com

Este trabalho foi editado e composto no MS-Word2.0 for Windows, em Arial de 11 pontos e Arial de 16 pontos; impresso com uma impressora Hewlett-Packard LaserJet4L.

São José dos Campos, novembro de 1994.