Manual do KAlgebra

Aleix Pol
Tradução: Luiz Fernando Ranghetti
Tradução: André Marcelo Alvarenga

Índice

1 Introdução **7**

2 Sintaxe **9**

3 Usando a calculadora **11**

4 Gráficos 2D **13**
4.1 Sintaxe . . . 13
4.2 Recursos . . . 13

5 Gráficos 3D **15**

6 Dicionário **17**

7 Comandos suportados pelo KAlgebra **18**
7.1 plus . . . 18
7.2 times . . . 18
7.3 minus . . . 18
7.4 divide . . . 18
7.5 quotient . . . 19
7.6 power . . . 19
7.7 root . . . 19
7.8 factorial . . . 19
7.9 and . . . 19
7.10 or . . . 20
7.11 xor . . . 20
7.12 not . . . 20
7.13 gcd . . . 20
7.14 lcm . . . 20
7.15 rem . . . 21
7.16 factorof . . . 21
7.17 max . . . 21
7.18 min . . . 21
7.19 lt . . . 21
7.20 gt . . . 21
7.21 eq . . . 22

7.22 neq . . . 22
7.23 leq . . . 22
7.24 geq . . . 22
7.25 implies . . . 22
7.26 approx . . . 23
7.27 abs . . . 23
7.28 floor . . . 23
7.29 ceiling . . . 23
7.30 sin . . . 23
7.31 cos . . . 24
7.32 tan . . . 24
7.33 sec . . . 24
7.34 csc . . . 24
7.35 cot . . . 24
7.36 sinh . . . 24
7.37 cosh . . . 25
7.38 tanh . . . 25
7.39 sech . . . 25
7.40 csch . . . 25
7.41 coth . . . 25
7.42 arcsin . . . 25
7.43 arccos . . . 26
7.44 arctan . . . 26
7.45 arccot . . . 26
7.46 arccosh . . . 26
7.47 arccsc . . . 26
7.48 arccsch . . . 26
7.49 arcsec . . . 27
7.50 arcsech . . . 27
7.51 arcsinh . . . 27
7.52 arctanh . . . 27
7.53 exp . . . 27
7.54 ln . . . 28
7.55 log . . . 28
7.56 conjugate . . . 28
7.57 arg . . . 28
7.58 real . . . 28
7.59 imaginary . . . 29
7.60 sum . . . 29
7.61 product . . . 29
7.62 diff . . . 29
7.63 card . . . 29
7.64 scalarproduct . . . 30
7.65 selector . . . 30

7.66 union . . . 30
7.67 forall . . . 30
7.68 exists . . . 30
7.69 map . . . 31
7.70 filter . . . 31
7.71 transpose . . . 31

8 Créditos e licença **32**

Resumo

KAlgebra é um aplicativo que pode substituir a sua calculadora gráfica. Ele tem funcionalidades numéricas, lógicas, simbólicas e analíticas que lhe permitem calcular expressões matemáticas e desenhar graficamente os resultados em 2D ou 3D. O KAlgebra é baseado na Mathematical Markup Language (MathML). Entretanto, não é preciso conhecer o MathML para usar o KAlgebra.

Capítulo 1

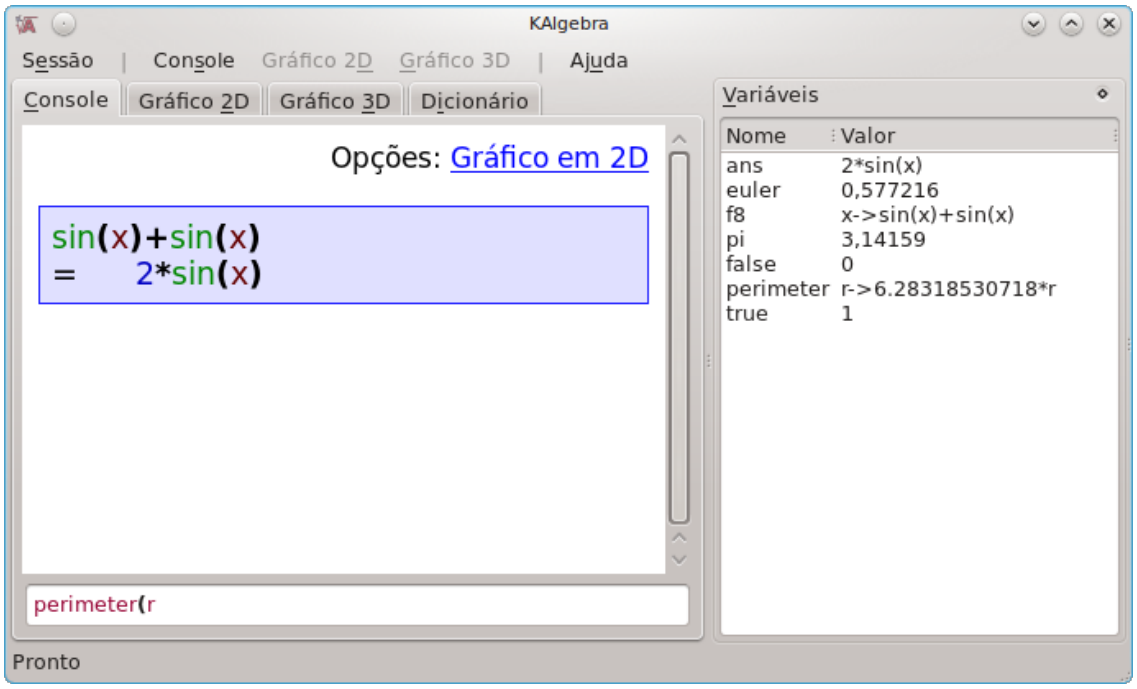
Introdução

O KAlgebra tem diversas funcionalidades que permitem ao usuário efetuar todos os tipos de operações matemáticas e mostrá-las de forma gráfica. No passado, este programa foi orientado para o MathML. Agora ele pode ser usado por todos os que tenham algum conhecimentos de matemática para resolver problemas simples e avançados.

Ele inclui algumas funcionalidades como por exemplo:

- Uma calculadora para avaliar funções matemáticas de maneira fácil e rápida.
- Capacidades de programação para séries avançadas de cálculos.
- Capacidades da linguagem que incluem a definição de funções e a completação automática da sintaxe.
- As funções de cálculo incluem o cálculo simbólico de derivadas, o cálculo vetorial e a manipulação de listas.
- Gráficos de funções com cursores dinâmicos para a descoberta gráfica de raízes e outros tipos de análises.
- Gráficos 3D para visualizações úteis de funções em 3D.
- Um dicionário de operadores incorporado para uma referência rápida das funções disponíveis.

Aqui está uma imagem do KAlgebra em ação:



Quando o usuário inicia uma sessão do KAlgebra, é apresentada uma única janela que consiste em uma aba de **Calculadora**, uma aba **Gráfico 2D** e outra **Gráfico 3D**, assim como uma aba de **Dicionário**. Em cada uma destas abas, você encontrará um campo de texto onde poderá digitar as suas funções ou fazer os seus cálculos e também um campo de visualização que apresenta os resultados.

Em qualquer tempo, o usuário poderá gerenciar a sua sessão com as opções do menu **Sessão**:

Sessão → Novo (Ctrl+N)

Abre uma nova janela do KAlgebra.

Sessão → Modo tela inteira (Ctrl+Shift+F)

Alterna a janela do KAlgebra para o modo em tela cheia. Também é possível usar este modo com o botão na parte superior à direita na janela do KAlgebra.

Sessão → Sair (Ctrl+Q)

Encerra o programa.

Capítulo 2

Sintaxe

O KAlgebra usa uma sintaxe algébrica intuitiva para introduzir as funções do usuário, semelhante a que é usada nas calculadoras gráficas modernas. Esta seção apresenta os operadores fundamentais que estão disponíveis no KAlgebra. O autor do KAlgebra modelou esta sintaxe com base no Maxima e no Maple para que os usuários possam se familiarizar com estes programas.

Para os usuários que estejam interessados no funcionamento interno do KAlgebra, as expressões introduzidas pelo usuário são convertidas para MathML pela infraestrutura. Uma compreensão rudimentar das capacidades suportadas pelo MathML dará um grande avanço sobre as capacidades internas do KAlgebra.

Aqui está uma lista dos operadores disponíveis que temos até agora:

- `+ - * /` : Soma, subtração, multiplicação e divisão.
- `^, **` : Potência, você poderá usar qualquer um deles. Também é possível usar os caracteres unicode ². As potências também são uma forma de calcular raízes, como pode ser feito em `a**(1/b)`
- `->` : lambda. É a forma de indicar uma ou mais variáveis livres que serão associadas a uma função. Por exemplo, na expressão **`comprimento:=(x,y)->(x*x+y*y)^0.5`**, o operador 'lambda' é usado para definir que o `x` e o `y` serão preenchidos quando for usada a função 'comprimento'.
- `x=a..b` : Isto é usado quando é necessário definir um intervalo [variável +limite-superior + limite-inferior). Isto significa que o `x` vai de `a` a `b`.
- `()` : É usado para especificar uma maior prioridade.
- `abc(parâmetros)` : Funções. Quando o processador encontrar uma função, verifica se o `abc` é um operador. Se for, será tratado como tal; se não for, será tratado como uma função do usuário.
- `:=` : Definição. É usada para definir o valor de uma variável. Você poderá fazer coisas do tipo **`x:=3`**, **`x:=y`**, sendo que o `y` pode ser definido ou não, ou ainda **`perímetro:=r->2*pi*r`**.
- `?` : Definição de condições. Esta é a forma de definir operações condicionais no KAlgebra. Dito de outra forma, esta é uma maneira de especificar uma condição if, elseif, else. Se introduzir a condição antes do '`?`', será usada apenas se for verdadeira; se encontrar um '`?`' sem qualquer condição, irá entrar na última instância. Por exemplo: **`condição { x=0 ? 0, x=1 ? x+1, ? x**2 }`**
- `{ }` : Contenedor MathML. Pode ser usado para definir um contenedor. É principalmente útil para lidar com as definições de operações condicionais.

- `=` `>` `>=` `<` `<=` : Comparações dos valores para 'igual', 'maior', 'maior ou igual', 'menor' ou 'menor ou igual', respectivamente.

Agora poderá perguntar: para que interessa então o MathML? É simples: com ele, poderá usar funções como `cos()`, `sin()`, outras funções trigonométricas, o `sum()` ou o `product()`. Não interessa o seu tipo. Poderá usar o `plus()`, `times()` e tudo o que tiver o seu operador. As funções booleanas estão também implementadas, pelo que poderá fazer algo do gênero `or(1,0,0,0,0)`.

Capítulo 3

Usando a calculadora

A calculadora do KAlgebra é útil como uma calculadora com esteroides. O usuário poderá introduzir expressões para avaliar no modo **Calcular** ou **Avaliar**, dependendo da seleção do menu **Calculadora**.

No modo de avaliação, o KAlgebra simplifica a expressão, mesmo quando vê uma variável não definida. O modo de cálculo do KAlgebra calcula tudo e, se encontrar uma variável não definida, apresenta um erro.

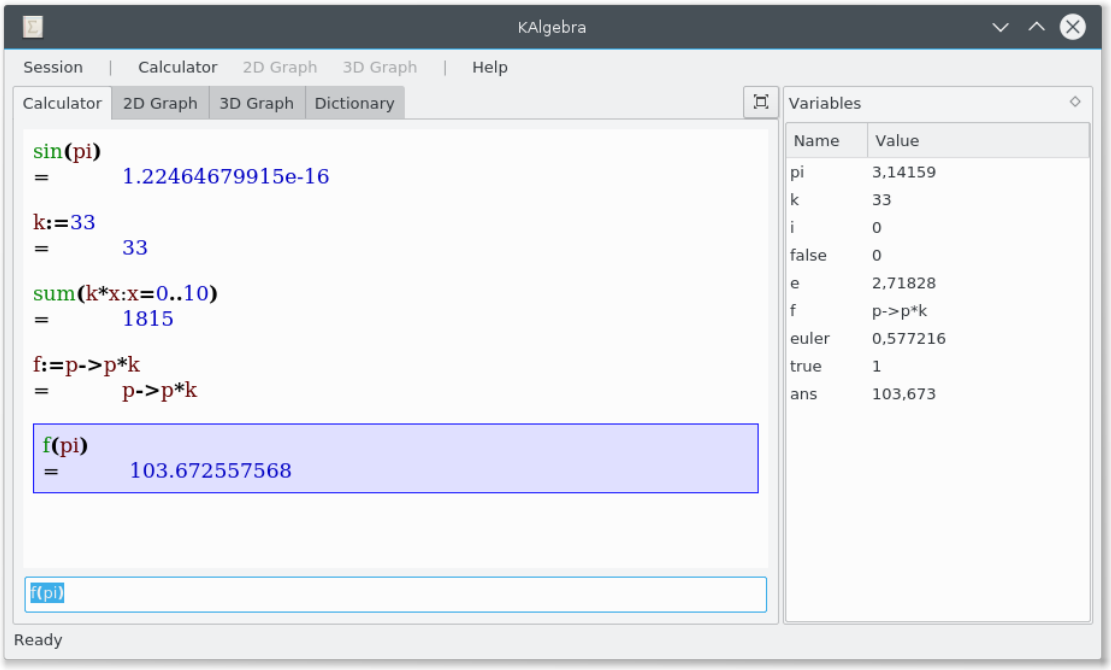
Além de mostrar as equações introduzidas pelo usuário e os resultados na área da Calculadora, todas as variáveis declaradas são apresentadas em uma área persistente à direita. Ao clicar duas vezes sobre uma variável, você poderá ver uma janela que lhe permite alterar os seus valores (apenas uma forma de enganar o registro).

A variável '`ans`' é especial. Sempre que introduzir uma expressão, o valor da variável '`ans`' será alterado para o último resultado.

As funções seguintes são exemplos que podem ser introduzidos no campo de texto da janela da calculadora:

- **`sin(pi)`**
- **`k:=33`**
- **`sum(k*x :  x=0..10)`**
- **`f:=p->p*k`**
- **`f(pi)`**

Segue-se uma imagem da janela da calculadora depois de introduzir as expressões de exemplo acima:



Um usuário poderá controlar a execução de uma série de cálculos usando as opções do menu **Calculadora**:

Calculadora → **Carregar script... (Ctrl+L)**

Executa as instruções de forma sequencial a partir de um arquivo. Útil se você quiser definir algumas bibliotecas ou terminar algum trabalho anterior.

Calculadora → **Scripts recentes**

Exibe um submenu que permitirá escolher os scripts executados recentemente.

Calculadora → **Salvar script... (Ctrl+G)**

Salva as instruções que escreveu desde o início da sessão, para poder reutilizá-las. Gera arquivos de texto, de modo a serem fáceis de alterar com qualquer editor de texto, como o Kate.

Calculadora → **Exportar registro... (Ctrl+S)**

Salva o registro com todos os resultados em um arquivo HTML, para poder ser impresso ou publicado.

Calculadora → **Inserir ans... (F3)**

Insere a variável '`ans`' e isso tornará mais fácil reutilizar valores antigos.

Calculadora → **Calcular**

Um botão de opção para definir o **Modo de execução** para cálculo.

Calculadora → **Avaliar**

Um botão de opção para definir o **Modo de execução** para avaliação.

Capítulo 4

Gráficos 2D

Para adicionar um novo gráfico 2D no KAlgebra, selecione a aba **Gráfico 2D** e clique na aba **Adicionar** para adicionar uma nova função. Depois, o seu foco irá para um campo de texto onde você poderá escrever a sua função.

4.1 Sintaxe

Se quiser usar uma função típica `f(x)`, não é necessário especificá-la; mas se quiser uma função `f(y)` ou uma função polar, terá de adicionar `y->` e `q->` como variáveis de fronteira.

Exemplos:

- **`sen(x)`**
- **`x²`**
- **`y->sen(y)`**
- **`q->3*sen(7*q)`**
- **`t->vector{sin t, t**2}`**

Se você tiver digitado a função, clique no botão **OK** para mostrar o gráfico na janela principal.

4.2 Recursos

Você poderá ter vários gráfico na mesma janela. Basta usar o botão **Adicionar** quando estiver no modo de lista. Você poderá atribuir a cada gráfico a sua própria cor.

A janela poderá ser ampliada e movida com o mouse. Usando a roda do mouse você poderá ampliar e reduzir a mesma. Poderá também selecionar uma área com o botão esquerdo do mouse, ficando apenas esta área ampliada. Mova a visão com as teclas de seta do teclado.

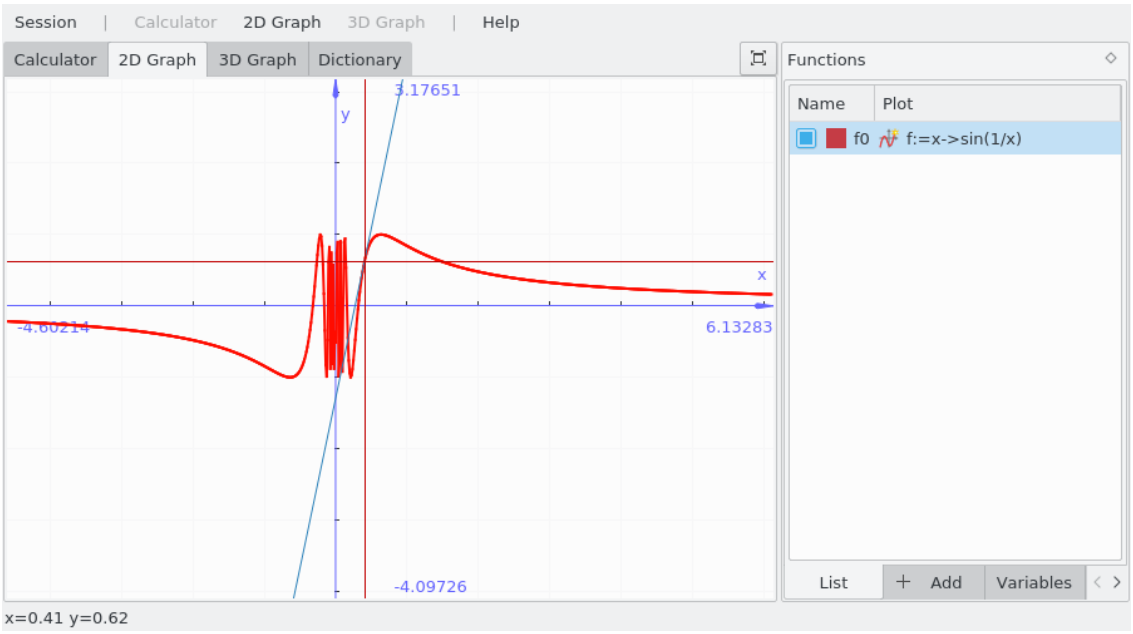
> NOTA
> A área de visualização dos gráficos 2D pode ser definida de forma explícita usando a aba **Visualizador** dentro da aba **Gráfico 2D**.

Na aba **Lista**, na parte inferior, você pode abrir uma aba de **Edição** para editar ou remover uma função com clique duplo e marcar ou desmarcar a opção após o nome da função para a mostrar ou ocultá-la.

No menu do **Gráfico 2D**, você poderá encontrar estas opções:

- **Grade**: Mostra ou oculta a grade
- **Manter proporção**: Mantém a proporção durante o zoom
- **Salvar**: Salva (**Ctrl**+**S**) o gráfico como um arquivo de imagem
- **Ampliar/Reduzir**: Amplia (**Ctrl**++) e reduz (**Ctrl**+-)
- **Tamanho real**: Restaura a janela com o nível de ampliação original
- **Resolução**: Seguido por uma lista de botões de opção para selecionar uma resolução para os gráficos

Abaixo está uma captura de tela de um usuário cujo cursor está na raiz mais à direita da função `sin(1/x)`. O usuário que gerou o gráfico usou uma resolução muito precisa para criar o desenho (conforme ele oscila em frequências cada vez mais altas próximo a origem). Existe também uma funcionalidade de cursor ao vivo em qualquer lugar que você mova o cursor sobre um ponto. Ela mostra os valores `x` e `y` no canto inferior esquerdo da tela. Uma 'linha tangente' ao vivo é plotada na função na localização do cursor atual.



Capítulo 5

Gráficos 3D

Para desenhar um gráfico 3D com o KAlgebra, selecione a aba **Gráficos 3D**, onde irá ver um campo de texto na base que será usado para digitar a sua função. O `z` ainda não pode ser definido. Por enquanto o KAlgebra só suporta gráficos 3D que dependam explicitamente apenas de `x` e `y`, como por exemplo **`(x,y)->x*y`**, onde o **`z=x*y`**.

Exemplos:

- **`(x,y)->sin(x)*sin(y)`**
- **`(x,y)->x/y`**

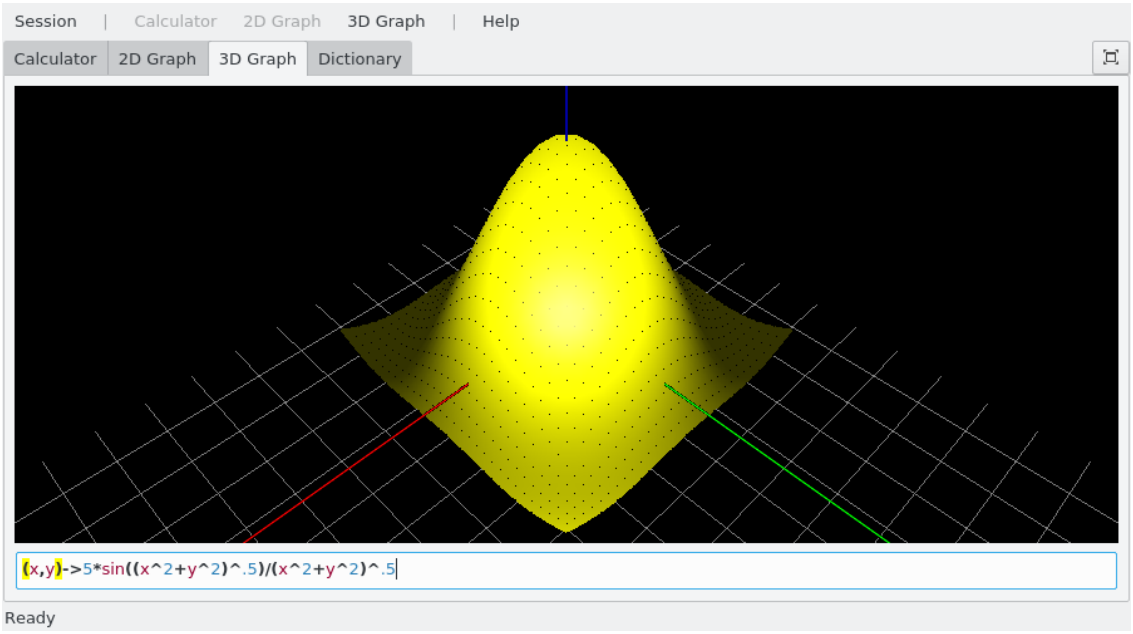
A janela pode ser ampliada e movida com o mouse. Usando a roda do mouse você poderá ampliar e reduzir a mesma. Mantenha o botão esquerdo do mouse clicado e mova o mouse para rodar o gráfico.

As teclas de cursores para a **Esquerda** e **Direita** giram o gráfico em torno do eixo `z`, enquanto as teclas para **Acima** e para **Abaixo** giram em torno do eixo horizontal da janela. Pressione a tecla **W** para ampliar o gráfico e em **S** para reduzi-lo.

No menu **Gráfico 3D**, você irá encontrar estas opções:

- **Salvar**: Salva (**Ctrl**+**S**) o gráfico como um arquivo de imagem ou documento suportado
- **Redefinir visualização**: Restaura a janela com o nível de ampliação original no menu de **Gráfico 3D**
- Você pode desenhar os gráficos com os estilos de **Pontos**, **Linhas** ou **Sólido** no menu **Gráfico 3D**

Abaixo está uma imagem do função chamada 'sombrero'. Este gráfico em particular é mostrado usando o estilo em linha de gráfico 3D.

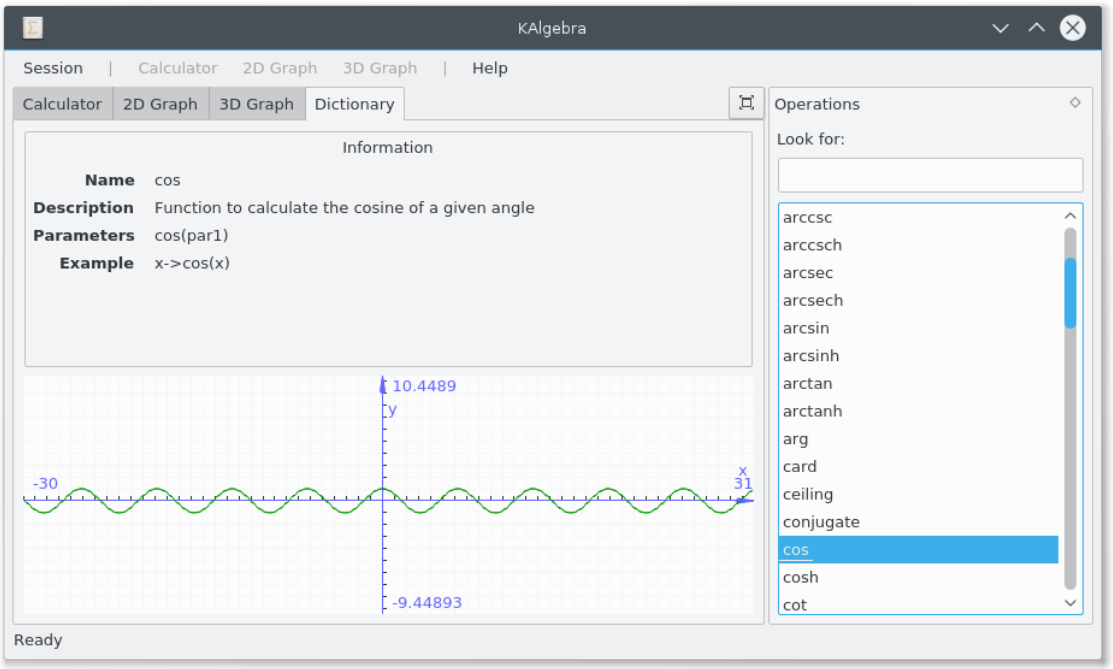

Session | Calculator 2D Graph 3D Graph | Help
Calculator 2D Graph 3D Graph Dictionary
(x,y)->5*sin((x^2+y^2)^.5)/(x^2+y^2)^.5
Ready

Capítulo 6

Dicionário

O dicionário fornece uma lista de todas as funções embutidas no KAlgebra. Ele pode ser usada para encontrar a definição de uma operação e seus parâmetros de entrada. É um local útil para ir para encontrar os diversos recursos do KAlgebra.

Abaixo está uma captura de tela do dicionário do KAlgebra procurando a função cosine.



Capítulo 7

Comandos suportados pelo KAlgebra

7.1 plus

- Nome: plus
- Descrição: Adição
- Parâmetros: plus(... parâmetros, ...)
- Exemplo: x->x+2

7.2 times

- Nome: times
- Descrição: Multiplicação
- Parâmetros: times(... parâmetros, ...)
- Exemplo: x->x*2

7.3 minus

- Nome: minus
- Descrição: Subtração. Irá remover todos os valores do primeiro.
- Parâmetros: minus(... parâmetros, ...)
- Exemplo: x->x-2

7.4 divide

- Nome: divide
- Descrição: Divisão
- Parâmetros: divide(par1, par2)
- Exemplo: x->x/2

7.5 quotient

- Nome: quotient
- Descrição: Quociente
- Parâmetros: quotient(par1, par2)
- Exemplo: x->quotient(x, 2)

7.6 power

- Nome: power
- Descrição: Potência
- Parâmetros: power(par1, par2)
- Exemplo: x->x^2

7.7 root

- Nome: root
- Descrição: Raiz
- Parâmetros: root(par1, par2)
- Exemplo: x->root(x, 2)

7.8 factorial

- Nome: factorial
- Descrição: Fatorial. fatorial(n)=n!
- Parâmetros: factorial(par1)
- Exemplo: x->factorial(x)

7.9 and

- Nome: and
- Descrição: 'E' booleano
- Parâmetros: and(... parâmetros, ...)
- Exemplo: x->piecewise { and(x>-2, x<2) ? 1, ? 0 }

7.10 or

- Nome: or
- Descrição: 'ou' booleano
- Parâmetros: or(... parâmetros, ...)
- Exemplo: x->piecewise { or(x>2, x>-2) ? 1, ? 0 }

7.11 xor

- Nome: xor
- Descrição: 'ou exclusivo' booleano
- Parâmetros: xor(... parâmetros, ...)
- Exemplo: x->piecewise { xor(x>0, x<3) ? 1, ? 0 }

7.12 not

- Nome: not
- Descrição: 'não' booleano
- Parâmetros: not(par1)
- Exemplo: x->piecewise { not(x>0) ? 1, ? 0 }

7.13 gcd

- Nome: gcd
- Descrição: Máximo divisor comum
- Parâmetros: gcd(... parâmetros, ...)
- Exemplo: x->gcd(x, 3)

7.14 lcm

- Nome: lcm
- Descrição: Mínimo múltiplo comum
- Parâmetros: lcm(... parâmetros, ...)
- Exemplo: x->lcm(x, 4)

7.15 rem

- Nome: rem
- Descrição: Resto
- Parâmetros: rem(par1, par2)
- Exemplo: x->rem(x, 5)

7.16 factorof

- Nome: factorof
- Descrição: O fator de
- Parâmetros: factorof(par1, par2)
- Exemplo: x->factorof(x, 3)

7.17 max

- Nome: max
- Descrição: Máximo
- Parâmetros: max(... parâmetros, ...)
- Exemplo: x->max(x, 4)

7.18 min

- Nome: min
- Descrição: Mínimo
- Parâmetros: min(... parâmetros, ...)
- Exemplo: x->min(x, 4)

7.19 lt

- Nome: lt
- Descrição: Menor que. lt(a,b)=a<b
- Parâmetros: lt(par1, par2)
- Exemplo: x->piecewise { x<4 ? 1, ? 0 }

7.20 gt

- Nome: gt
- Descrição: Maior que. gt(a,b)=a>b
- Parâmetros: gt(par1, par2)
- Exemplo: x->piecewise { x>4 ? 1, ? 0 }

7.21 eq

- Nome: eq
- Descrição: Igual. eq(a,b) = a=b
- Parâmetros: eq(par1, par2)
- Exemplo: x->piecewise { x=4 ? 1, ? 0 }

7.22 neq

- Nome: neq
- Descrição: Diferente. neq(a,b)=a≠b
- Parâmetros: neq(par1, par2)
- Exemplo: x->piecewise { x!=4 ? 1, ? 0 }

7.23 leq

- Nome: leq
- Descrição: Menor ou igual. leq(a,b)=a≤b
- Parâmetros: leq(par1, par2)
- Exemplo: x->piecewise { x<=4 ? 1, ? 0 }

7.24 geq

- Nome: geq
- Descrição: Maior ou igual. geq(a,b)=a≥b
- Parâmetros: geq(par1, par2)
- Exemplo: x->piecewise { x>=4 ? 1, ? 0 }

7.25 implies

- Nome: implies
- Descrição: Implicação booleana
- Parâmetros: implies(par1, par2)
- Exemplo: x->piecewise { implies(x<0, x<3) ? 1, ? 0 }

7.26 approx

- Nome: approx
- Descrição: Aproximação. approx(a)=a±n
- Parâmetros: approx(par1, par2)
- Exemplo: x->piecewise { approx(x, 4) ? 1, ? 0 }

7.27 abs

- Nome: abs
- Descrição: Valor absoluto. abs(n)=|n|
- Parâmetros: abs(par1)
- Exemplo: x->abs(x)

7.28 floor

- Nome: floor
- Descrição: Valor por defeito. floor(n)=⌊n⌋
- Parâmetros: floor(par1)
- Exemplo: x->floor(x)

7.29 ceiling

- Nome: ceiling
- Descrição: Valor por excesso. ceil(n)=⌈n⌉
- Parâmetros: ceiling(par1)
- Exemplo: x->ceiling(x)

7.30 sin

- Nome: sin
- Descrição: Função para calcular o seno de um determinado ângulo
- Parâmetros: sin(par1)
- Exemplo: x->sin(x)

7.31 cos

- Nome: cos
- Descrição: Função para calcular o cosseno de um dado ângulo
- Parâmetros: cos(par1)
- Exemplo: x->cos(x)

7.32 tan

- Nome: tan
- Descrição: Função para calcular a tangente de um dado ângulo
- Parâmetros: tan(par1)
- Exemplo: x->tan(x)

7.33 sec

- Nome: sec
- Descrição: Secante
- Parâmetros: sec(par1)
- Exemplo: x->sec(x)

7.34 csc

- Nome: csc
- Descrição: Co-secante
- Parâmetros: csc(par1)
- Exemplo: x->csc(x)

7.35 cot

- Nome: cot
- Descrição: Co-tangente
- Parâmetros: cot(par1)
- Exemplo: x->cot(x)

7.36 sinh

- Nome: sinh
- Descrição: Seno hiperbólico
- Parâmetros: sinh(par1)
- Exemplo: x->sinh(x)

7.37 cosh

- Nome: cosh
- Descrição: Cosseno hiperbólico
- Parâmetros: cosh(par1)
- Exemplo: x->cosh(x)

7.38 tanh

- Nome: tanh
- Descrição: Tangente hiperbólica
- Parâmetros: tanh(par1)
- Exemplo: x->tanh(x)

7.39 sech

- Nome: sech
- Descrição: Secante hiperbólica
- Parâmetros: sech(par1)
- Exemplo: x->sech(x)

7.40 csch

- Nome: csch
- Descrição: Co-secante hiperbólica
- Parâmetros: csch(par1)
- Exemplo: x->csch(x)

7.41 coth

- Nome: coth
- Descrição: Cotangente hiperbólica
- Parâmetros: coth(par1)
- Exemplo: x->coth(x)

7.42 arcsin

- Nome: arcsin
- Descrição: Arco-seno
- Parâmetros: arcsin(par1)
- Exemplo: x->arcsin(x)

7.43 arccos

- Nome: arccos
- Descrição: Arco-cosseno
- Parâmetros: arccos(par1)
- Exemplo: x->arccos(x)

7.44 arctan

- Nome: arctan
- Descrição: Arco-tangente
- Parâmetros: arctan(par1)
- Exemplo: x->arctan(x)

7.45 arccot

- Nome: arccot
- Descrição: Arco-cotangente
- Parâmetros: arccot(par1)
- Exemplo: x->arccot(x)

7.46 arccosh

- Nome: arccosh
- Descrição: Arco-cosseno hiperbólico
- Parâmetros: arccosh(par1)
- Exemplo: x->arccosh(x)

7.47 arccsc

- Nome: arccsc
- Descrição: Arco-cossecante
- Parâmetros: arccsc(par1)
- Exemplo: x->arccsc(x)

7.48 arccsch

- Nome: arccsch
- Descrição: Arco-cossecante hiperbólica
- Parâmetros: arccsch(par1)
- Exemplo: x->arccsch(x)

7.49 arcsec

- Nome: arcsec
- Descrição: Arco-secante
- Parâmetros: arcsec(par1)
- Exemplo: x->arcsec(x)

7.50 arcsech

- Nome: arcsech
- Descrição: Arco-secante hiperbólica
- Parâmetros: arcsech(par1)
- Exemplo: x->arcsech(x)

7.51 arcsinh

- Nome: arcsinh
- Descrição: Arco-seno hiperbólico
- Parâmetros: arcsinh(par1)
- Exemplo: x->arcsinh(x)

7.52 arctanh

- Nome: arctanh
- Descrição: Arco-tangente hiperbólica
- Parâmetros: arctanh(par1)
- Exemplo: x->arctanh(x)

7.53 exp

- Nome: exp
- Descrição: Expoente (e^x)
- Parâmetros: exp(par1)
- Exemplo: x->exp(x)

7.54 ln

- Nome: ln
- Descrição: Logaritmo de base-e
- Parâmetros: ln(par1)
- Exemplo: x->ln(x)

7.55 log

- Nome: log
- Descrição: Logaritmo de base-10
- Parâmetros: log(par1)
- Exemplo: x->log(x)

7.56 conjugate

- Nome: conjugate
- Descrição: Conjugado
- Parâmetros: conjugate(par1)
- Exemplo: x->conjugate(x*i)

7.57 arg

- Nome: arg
- Descrição: Argumento
- Parâmetros: arg(par1)
- Exemplo: x->arg(x*i)

7.58 real

- Nome: Parte real
- Descrição: Real
- Parâmetros: real(par1)
- Exemplo: x->real(x*i)

7.59 imaginary

- Nome: imaginary
- Descrição: Parte imaginária
- Parâmetros: imaginary(par1)
- Exemplo: x->imaginary(x*i)

7.60 sum

- Nome: sum
- Descrição: Somatório
- Parâmetros: sum(par1 : var=de..até)
- Exemplo: x->x*sum(t*t:t=0..3)

7.61 product

- Nome: product
- Descrição: Produtório
- Parâmetros: product(par1 : var=de..até)
- Exemplo: x->product(t+t:t=1..3)

7.62 diff

- Nome: diff
- Descrição: Derivada
- Parâmetros: diff(par1 : var)
- Exemplo: x->(diff(x^2:x))(x)

7.63 card

- Nome: card
- Descrição: Cardinalidade
- Parâmetros: card(par1)
- Exemplo: x->card(vector { x, 1, 2 })

7.64 scalarproduct

- Nome: scalarproduct
- Descrição: Produto escalar
- Parâmetros: scalarproduct(... parâmetros, ...)
- Exemplo: x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]

7.65 selector

- Nome: selector
- Descrição: Seleciona o par1-ésimo elemento da lista ou vetor 'par2'
- Parâmetros: selector(par1, par2)
- Exemplo: x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]

7.66 union

- Nome: union
- Descrição: Junta vários itens do mesmo tipo
- Parâmetros: union(... parâmetros, ...)
- Exemplo: x->union(list { 1, 2, 3 }, list { 4, 5, 6 })[rem(floor(x), 5)+3]

7.67 forall

- Nome: forall
- Descrição: Para todos
- Parâmetros: forall(par1 : var)
- Exemplo: x->piecewise { forall(t:t@list { true, false, false }) ? 1, ? 0 }

7.68 exists

- Nome: exists
- Descrição: Existe
- Parâmetros: exists(par1 : var)
- Exemplo: x->piecewise { exists(t:t@list { true, false, false }) ? 1, ? 0 }

7.69 map

- Nome: map
- Descrição: Aplica uma função para cada elemento em uma lista
- Parâmetros: map(par1, par2)
- Exemplo: x->map(x->x+x, list { 1, 2, 3, 4, 5, 6 })[rem(floor(x), 5)+3]

7.70 filter

- Nome: filter
- Descrição: Remove todos os elementos que não corresponde a uma condição
- Parâmetros: filter(par1, par2)
- Exemplo: x->filter(u->rem(u, 2)=0, list { 2, 4, 3, 4, 8, 6 })[rem(floor(x), 5)+3]

7.71 transpose

- Nome: transpose
- Descrição: Transpose
- Parâmetros: transpose(par1)
- Exemplo: x->transpose(matrix { matrixrow { 1, 2, 3, 4, 5, 6 } })[rem(floor(x), 5)+3][1]

Capítulo 8

Créditos e licença

- Direitos autorais do programa, 2005-2009 Aleix Pol

Direitos autorais da documentação 2007 Aleix Pol aleixpol@gmail.com

Tradução:

Luiz Fernando Ranghetti elchevive@opensuse.org

André Marcelo Alvarenga alvarenga@kde.org

Esta documentação é licenciada sob os termos da Licença de Documentação Livre GNU.

Este programa é licenciado sob os termos da Licença Pública Geral GNU.