Toolbox de matemática simbólica

Table of Contents

Balanceamento 1
O pacote simbólico 1
PI 4
Equações algébricas 5
Distributiva 5
Fatoração 5
Fatoração com múltiplas variáveis 5
Simplificação 6
Equações do segundo grau 6
Limite 7
Derivadas 8
Integrais 9
Mais condições 10
Integrais definidas 10
Séries de Taylor 11
Soma simbólica - Série geométrica 11
Frações parciais 12

Nesta aula, veremos o uso do Toolbox de Matemática Simbólica do MATLAB®, o *Symbolic Math Toolbox*.

Balanceamento

A partir das equações abaixo:

$$C_3H_8 + O_2 \text{ -¿ } CO_2 + H_2O$$

O pacote simbólico

É um pacote comprado à parte, que usa um paradigma diferente do normalmente usado no MATLAB®. Por padrão, o MATLAB® usa cálculos numéricos, com valores aproximados. No Toolbox de matemática simbólica, os cálculos são exatos.

```
clear all; close all; clc; format compact
```

Como exemplo de limitação da matemática numérica, vemos a comparação abaixo. Sendo *eps*

```
eps
```

```
ans =
   2.2204e-16
```

A comparação abaixo ainda é entendida como verdadeira em matemática numérica:

```
1 + eps/2 == 1
```

```
ans =
     1
```

Um exemplo é o fatorial de 52:

```
factorial(52)
```

```
ans =
   8.0658e+67
```

Para encontrar todas as casas do fatorial, usamos a função sym():

```
sym('factorial(52)')
```

```
ans =
80658175170943878571660636856403766975289505440883277824000000000000
```

Uma potência grande de 2, também mostra a diferença:

```
disp('- Numérico:')
2^100

disp('- Simbólico:')
sym('2^100')
```

```
- Numérico:
ans =
   1.2677e+30
- Simbólico:
ans =
1267650600228229401496703205376
```

Além disso, o pacote simbólico pode trabalhar com variáveis:

```
expr1 = sym('x^2 + sin(x)')
expr2 = sym('a*x + b')
```

```
expr1 =
sin(x) + x^2
expr2 =
b + a*x
```

Repare que as variáveis criadas representam expressões matemáticas, e a classe das variáveis é diferente:

```
whos expr1 expr2
```

```
  Name        Size            Bytes  Class    Attributes

  expr1       1x1               112  sym
  expr2       1x1               112  sym
```

Podemos também realizar operações matemáticas com as expressões:

```
expr3 = expr1 / expr2
```

```
expr3 =
(sin(x) + x^2)/(b + a*x)
```

Funções de visualização podem ser usadas para melhorar a exibição dos resultados:

```
disp('Função pretty()')
pretty(expr3)

disp('Função latex()')
latex(expr3)
```

```
Função pretty()
          2
sin(x) + x
-----------
  b + a x

Função latex()
ans =
\frac{\sin\!\left(x\right) + x^2}{b + a\, x}
```

Além da criação de expressões usando a função *sym()*, a função *syms* pode ser utilizada, trazendo as variáveis que serão consideradas simbólicas pelo MATLAB® dali para frente:

```
syms x y z a b
```

Repare que as variáveis são criadas no workspace com o tipo sym:

```
whos x y z a b
```

```
  Name      Size            Bytes  Class    Attributes

  a         1x1               112  sym
  b         1x1               112  sym
  x         1x1               112  sym
  y         1x1               112  sym
  z         1x1               112  sym
```

Neste caso, as expressões não precisam mais ser definidas como strings:

```
expr4 = (x*y + a^2 * b) / sin(z)
```

```
expr4 =
(b*a^2 + x*y)/sin(z)
```

Desigualdades são criadas via operadores relacionais:

```
expr5 = a*x == z^2
expr6 = a^2 * x <= sin(z^2)
```

```
expr5 =
a*x == z^2
expr6 =
a^2*x <= sin(z^2)
```

Para calcular as expressões em valores específicos das variáveis, usamos a função *subs()*:

```
subs(expr2,b,17)
```

```
ans =
a*x + 17
```

Repare que a variável "b" foi substituída pelo valor 17. Além disso, outros valores simbólicos podem ser usados na substituição:

```
subs(expr2,b,2*a*x)
```

```
ans =
3*a*x
```

Para substituir mais de um valor, usa-se células na função *subs()*:

```
subs(expr2,{a,b},{7,-1})
e = subs(expr2,{a,b,x},{7,-1,10})
```

```
ans =
7*x - 1
e =
69
```

Repare que, mesmo substituindo todas as variáveis, o retorno da função ainda é uma variável simbólica:

```
whos e
```

```
  Name       Size             Bytes  Class     Attributes

  e          1x1                112  sym
```

Para converter o resultado para um número, usamos a função *double()*

```
double(e)
```

```
ans =
    69
```

PI

Para usar o \pi em substituições, podemos fazer de duas formas. Na primeira, a substituição é feita com o numérico 3,14..., na segunda, o valor exato de pi é usado. O recomendado é sempre usar aspas:

```
sym(pi+1)
sym('pi+1')
```

```
ans =
1165754695714211/281474976710656
ans =
pi + 1
```

A função *subs()* pode ser usada para substituir todas as variáveis que possuem definição numérica no workspace, para isso, basta não passar nenhum parâmetro adicional:

```
expr2
b = 17    % Numérico
subs(expr2)
```

```
expr2 =
b + a*x
```

```
b =
    17
ans =
a*x + 17
```

Equações algébricas

O toolbox de matemática simbólica pode ser usado para manipulações algébricas.

```
syms x
```

Distributiva

Seja a expressão:

```
expr1 = x*(x+1)
```

```
expr1 =
x*(x + 1)
```

Para fazer a distribuição de x, usa-se a função *expand()*:

```
expr2 = expand(expr1)
```

```
expr2 =
x^2 + x
```

Fatoração

Seja a expressão

```
expr2
```

```
expr2 =
x^2 + x
```

Vemos que *x* é um fator comum, por isso pode ser fatorado:

```
factor(expr2)
```

```
ans =
[ x, x + 1]
```

Para retornar à forma retornada pela função *expand()*, usamos a função *prod()*:

```
prod(factor(expr2))
```

```
ans =
x*(x + 1)
```

Fatoração com múltiplas variáveis

Para casos onde temos mais de uma variável, a função *collect()* faz a fatoração para a variável escolhida. Seja a expressão abaixo:

```
expr3 = x*y^2 + (1+x)*y + y^2
```

```
expr3 =
y*(x + 1) + x*y^2 + y^2
```

A fatoração em termos de *x* e de *y* é:

```
disp('"x" como fator comum:')
collect(expr3,x)

disp('"y" como fator comum:')
collect(expr3,y)
```

```
"x" como fator comum:
ans =
(y^2 + y)*x + y^2 + y
"y" como fator comum:
ans =
(x + 1)*y^2 + (x + 1)*y
```

Simplificação

Simplificar uma expressão nem sempre é claro. Mas o toolbox tem uma versão própria, que pode ajudar em algumas aplicações. Para isso, usa-se a função *simplify()*

```
simplify(expr3)
```

```
ans =
y*(x + 1)*(y + 1)
```

Outros exemplos interessantes são:

```
expr4 = cos(x)^2 + sin(x)^2
simplify(expr4)

expr5 = (x^2 + x)/(2*x)
simplify(expr5)
```

```
expr4 =
cos(x)^2 + sin(x)^2
ans =
1
expr5 =
(x^2 + x)/(2*x)
ans =
x/2 + 1/2
```

Equações do segundo grau

O toolbox também calcula raízes de equações do segundo grau:

```
syms a b c
eq2grau = a*x^2 + b*x + c
```

```
eq2grau =
```

```
a*x^2 + b*x + c
```

Para resolver, usa-se a função *solve()*, que aplica Bhaskara e nos retorna um vetor com os valores exatos das raízes:

```
solve(eq2grau)

ans =
 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

Limite

Podemos usar operações de cálculo também de forma simbólica. Por exemplo, para calcular o limite abaixo:

$$lim_{xtowards0}\frac{\sin(x)}{x}$$

Podemos fazer numericamente, mas recebemos um NaN:

```
sin(0)/0

ans =
   NaN
```

Sabemos por L'Hopital que o limite resulta em 1. Ao aplicar um número que se aproxima de zero, temos:

```
format long
sin(0.000001)/0.000001
format

ans =
   0.999999999999833
```

Vemos que o número se aproxima de 1, mas isto não é uma prova que ele é 1. Se ao invés disso usarmos a matemática simbólica, temos uma resolução exata:

```
syms x
limit(sin(x)/x)


ans =

1
```

Outro exemplo é o limite de:

$$\lim_{xtowards\inf}(1+\frac{1}{x})^x = e$$

```
limit((1+1/x)^x,x,inf)
```

```
ans =

exp(1)
```

Derivadas

Podemos também resolver derivadas. Para isto, usamos a função *diff()*:

```
syms a b x
diff(a*x^2 + sin(b*x),x)


ans =

2*a*x + b*cos(b*x)
```

Para derivadas de ordem superior, basta incluir mais um parâmetro:

```
disp('- Segunda ordem:')
diff(a*x^2 + sin(b*x),x,2)
disp('- Terceira ordem:')
diff(a*x^2 + sin(b*x),x,3)
disp('- Quarta ordem:')
diff(a*x^2 + sin(b*x),x,4)
disp('- Quinta ordem:')
diff(a*x^2 + sin(b*x),x,5)

- Segunda ordem:

ans =

2*a - b^2*sin(b*x)

- Terceira ordem:

ans =

-b^3*cos(b*x)

- Quarta ordem:

ans =

b^4*sin(b*x)

- Quinta ordem:

ans =

b^5*cos(b*x)
```

Integrais

Integrais podem ser calculadas com a função *int()*

```
int(x^3,x)
int(x^-1,x)
```

```
ans =

x^4/4


ans =

log(x)
```

Veja o caso abaixo:

```
func1 = x^a
```

```
func1 =

x^a
```

A sua integral é

```
int(x^a,x)
```

```
ans =

piecewise([a == -1, log(x)], [a ~= -1, x^(a + 1)/(a + 1)])
```

Repare que o MATLAB® retorna múltiplos resultados, dependendo do valor de *a*:

```
pretty(int(x^a,x))
```

```
{ log(x)  if  a == -1
{
{  a + 1
{ x
{ ------  if  a ~= -1
{  a + 1
```

No caso em que $a = -1$$, o denominador da fração daria zero, impedindo sua solução. Por isso, há duas condições. Quando sabemos a região onde *a* é definido, podemos informar isto através da função *assume()*

```
assume(a > 0)
```

Agora a integral é:

```
int(x^a,x)
```

```
ans =

x^(a + 1)/(a + 1)
```

Para ver todas as condições passadas:

```
assumptions(a)
```

```
ans =

0 < a
```

Mais condições

Sabemos que *cos(2 \pi x) = 1*, para x E inteiros, mas o MATLAB® não retorna esta condição naturalmente:

```
simplify(cos(2*pi*x))
```

```
ans =

cos(2*pi*x)
```

Incluindo a condição de que x E inteiros, temos:

```
assume(x,'integer')
simplify(cos(2*pi*x))
```

```
ans =

1
```

Integrais definidas

Para passar intervalos onde a integral será definida:

```
int(x^a,x,0,3)
```

```
ans =

(3*3^a)/(a + 1)
```

Podemos combinar condições de variaveis com operadores lógicos:

```
assume((a > 0) & (a < 2))
assumptions(a)
```

```
ans =

[ 0 < a, a < 2]
```

Séries de Taylor

Para a série abaixo:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$$

```
taylor(exp(x),x,'Order',5)
```

```
ans =

x^4/24 + x^3/6 + x^2/2 + x + 1
```

Soma simbólica - Série geométrica

Para a série abaixo:

$$\sum_{x=0}^{\inf} 1/2^x = 1 + 1/2 + 1/4 + ... = 2$$

```
symsum((1/2)^x,x,0,inf)
```

```
ans =

2
```

$$\sum_{x=0}^{\inf} r^x = 1 + r + r^2 + ...$$

```
syms r
symsum(r^x,x,0,inf)
```

```
ans =

piecewise([1 <= r, Inf], [abs(r) < 1, -1/(r - 1)])
```

Para trazer uma condição para r, usamos a função assume

```
assume(abs(r) < 1)
symsum(r^x,x,0,inf)
```

```
ans =

-1/(r - 1)
```

Frações parciais

Frações parciais são muito usadas em engenharias

```
expr = x^2 / (x^3 - 3 * x + 2)

pretty(expr)

partfrac(expr,x)

pretty(partfrac(expr,x))
```

```
expr =

x^2/(x^3 - 3*x + 2)

      2
     x
------------
 3
x  - 3 x + 2


ans =

5/(9*(x - 1)) + 1/(3*(x - 1)^2) + 4/(9*(x + 2))

    5            1            4
--------- + ---------- + ---------
9 (x - 1)            2   9 (x + 2)
            3 (x - 1)
```

Published with MATLAB® R2015a