Introduction to Maxima: tutorial

1 What's Maxima?

Maxima is a symbolic calculation software package written in the Lisp programming language. Its roots date back to the project Macsyma (1968).

2 Expressions, variables and functions

This is an expression

(%i1) x+1;

(%o1) $x+1$

An expression can be assigned to a variable with ":"

(%i2) expr1: x + 1;

expr1 $x+1$

One can manipulate expressions

(%i3) expr1^2;

(%o3) $(x+1)^2$

The expand function tries to get rid of the parenthesis

(%i4) expand(expr1^2);

(%o4) x^2+2x+1

This is how functions are defined:

(%i5) f(x):= x·sin(x);

(%o5) $f(x):=x\ \sin(x)$

This is how equations are defined:

(%i6) eq1: a · x^3 + 3 = 0;

eq1 $a\ x^3+3=0$

Equations can be solved (algebraically) with the function solve(equation,variable)

(%i7) solve(eq1,x);

(%o7) $$\left[x=-\left(\frac{3^{5/6}\ \%i-3^{1/3}}{2\ a^{1/3}}\right), x=\frac{3^{5/6}\ \%i+3^{1/3}}{2\ a^{1/3}}, x=-\left(\frac{3^{1/3}}{a^{1/3}}\right)\right]$$

The following are constants

(%i8) %i;

(%o8) %i

(%i9) %pi;

(%o9) π

(%i10) inf;

(%o10) ∞

By default, Maxima assumes x can be complex. One can also limit the solutions of eq1 by assuming x can only be real

3 Calculus

This is how one computes a (partial) derivative diff(expression,variable)

(%i11) 'diff(x^6 − 4·x,x);

(%o11) $\frac{d}{d\ x}(x^6-4\ x)$

The apostrophe means "don't evaluate the expression". If you remove it, the expression is evaluated

(%i12) diff(x^6 − 4·x,x);

5

(%o12) $6\ x^{5}-4$

This is the second derivative   diff(expression,variable,order)

(%i13) `diff(x^6−4·x,x,2);`

(%o13) $30\ x^{4}$

This is a indefinite integral

(%i14) `integrate(x^6−4·x,x);`

(%o14) $\frac{x^{7}}{7}-2\ x^{2}$

The last two parameters add integration limits

(%i15) `integrate(x^6−4·x,x,0,1);`

(%o15) $-\left(\frac{13}{7}\right)$

One can do taylor expansions

(%i16) `taylor_expansion: taylor(1/sqrt(1−x),x,0,4);`

taylor_expansion $1+\frac{x}{2}+\frac{3\ x^{2}}{8}+\frac{5\ x^{3}}{16}+\frac{35\ x^{4}}{128}+\ldots$

and you can get rid of the "..." with expand()

(%i17) `expand(taylor_expansion);`

(%o17) $\frac{35\ x^{4}}{128}+\frac{5\ x^{3}}{16}+\frac{3\ x^{2}}{8}+\frac{x}{2}+1$

This is an example of a differential equation

(%i18) `eq2: diff(g(x),x) = g(x);`

eq2 $\frac{d}{d\ x}\ g(x)=g(x)$

And it can be solved automatically by Maxima

(%i19) `desolve(eq2,g(x));`

(%o19) $g(x)=g(0)\ \%e^{x}$

Finally, here is the calculation of the differential. For example, imagine the change of variable inside of an integral
   u = cos(x)
   du = -sin(x) dx
del(x) here means dx.

(%i20) `diff(cos(x));`

(%o20) $-(\sin(x)\ \mathrm{del}(x))$

4 Simplification and assumptions

If you compute the square root of a square, the simplification adds an absolute value

(%i21) `sqrt(y^2);`

(%o21) $|y|$

If you tell Maxima that y is positive, ...

(%i22) `assume(y > 0);`

(%o22) $[y>0]$

... then it simplifies the expression without the abs()

(%i23) `sqrt(y^2);`

(%o23) y

You can specify the domain of a variable with the function domain()

(%i24) `domain(z,complex);`

(%o24) real (z , complex)

(%i25) **domain(t,real);**

(%o25) real (t , real)

(%i26) **domain(n,integer);**

(%o26) real (n , integer)

(%i27) **z: exp(%i·t);**

z $\%e^{\%i\ t}$

(%i28) **conjugate(z);**

(%o28) $\%e^{-(\%i\ t)}$

For rewriting an expression, expand() makes as many operations as possible to get rid of the parentheses

(%i29) **expand((x+1)^2);**

(%o29) $x^2 + 2\ x + 1$

Factor does the opposite job

(%i30) **factor(x^2 − 1);**

(%o30) $(x - 1)\ (x + 1)$

ratsimp() simplifies an expression

(%i31) **expr2: (x + 1)/(x^2 − 1);**

expr2 $\frac{x+1}{x^2-1}$

(%i32) **ratsimp(expr2);**

(%o32) $\frac{1}{x-1}$

5 Advanced expression manipulation

(%i33) **assume(x>0);**

(%o33) $[x > 0]$

Maxima expressions are internally nested lists whose zeroth element is the operation or function and the rest are the operands or variables. For example

(%i34) **2 · x + 3;**

(%o34) $2\ x + 3$

Is represented as (+ 3 (* 2 x))

(%i35) **?print(2·x + 3);**

```
((MPLUS SIMP) 3 ((MTIMES SIMP) 2 $X))
```

(%o35) $2\ x + 3$

We can do very precise expression manipulation with two functions: part() and substpart(). The first one picks a part of the original expression, which we can change, and the second function replaces that part with our manual modifications.

For example, consider the expression

(%i36) **expr3: 5 − cos(x)·sqrt(a^2 + x^2)/x;**

expr3 $5 - \frac{\sqrt{x^2 + a^2}\ \cos(x)}{x}$

We want to force factor an x^2 only inside of the square root. The functions "factor" or "ratsimp" won't do the job.

(%i37) **?print(expr3);**

```
((MPLUS SIMP) 5
 ((MTIMES SIMP) -1 ((MEXPT SIMP) $X -1)
  ((MEXPT SIMP) ((MPLUS SIMP) ((MEXPT SIMP) $A 2) ((MEXPT SIMP) $X 2))
   ((RAT SIMP) 1 2))
  ((%COS SIMP) $X)))
```

(%o37) $5 - \frac{\sqrt{x^2 + a^2}\ \cos(x)}{x}$

We use part() to separate the first depth level of the expression

(%i38) `part(expr3,0);`

(%o38) +

(%i39) `part(expr3,1);`

(%o39) 5

(%i40) `part(expr3,2);`

(%o40) $$-\left(\frac{\sqrt{x^2+a^2}\cos(x)}{x}\right)$$

The function dpart() can be used to highlight with a box the part of the expression that has been selected.

The first argument of dpart() is the expression, the rest of the arguments represent the "address" or "phone number" of the term we want to manipulate, inside of the expression tree.

(%i41) `dpart(expr3,2,1);`

(%o41) $$5-\left(\boxed{\frac{\sqrt{x^2+a^2}\cos(x)}{x}}\right)$$

(%i42) `to_be_factored: part(expr3,2,1,1,1,1);`

to_be_factored x^2+a^2

If you want to select more than one term, you can use a list. For the previous cell, an alternative way to selelect the same terms would have been

(%i43) `part(expr3,2,1,1,1,1,[1,2]);`

(%o43) x^2+a^2

Now that we have selected the inside of the square root, we multiply and divide the terms by x^2, checking that the function expand() will simplify the answer only where we want. This way we force the factor of x^2 out, while the division was carried out only inside of the parentheses

(%i44) `factored: x^2·expand(1/x^2·to_be_factored);`

factored $$\left(\frac{a^2}{x^2}+1\right)x^2$$

This function, substpart() inserts the edited expression back into the original expression. Tip:
always check that the "phone number" that you put in substpart() is the same as when you obtained
the part with the function part()

(%i45) `expr4: substpart(factored,expr3,2,1,1,1,1);`

expr4 $$5-\sqrt{\frac{a^2}{x^2}+1}\cos(x)$$

An application for this technique is a selective taylor expansion of only one term in an expression. For example, we want to expand only the square root, for small x.

(%i46) `part(expr4,2,1,1);`

(%o46) $$\sqrt{\frac{a^2}{x^2}+1}$$

(%i47) `expr5: expand(taylor(part(expr4,2,1,1),x,0,1));`

expr5 $$\frac{x}{2\,a}+\frac{a}{x}$$

(%i48) `substpart(expr5,expr4,2,1,1);`

(%o48) $$5-\left(\frac{x}{2\,a}+\frac{a}{x}\right)\cos(x)$$