Søren Højsgaard

| computer algebra system, CAS, yacas, #I

Introduction

Ryacas makes the yacas computer algebra system available from within R(yacas is short for "Yet Another Computer Algebra System").

yacas is developed by Ayal Pinkhuis (who is also the maintainer) and others and is available (for various platforms) at yacas.sourceforge.org. There is a comprehensive documentation (300+ pages) of yacas (also available at yacas.sourceforge.org) and the documentation contains many examples.

The examples given here are largely taken from the Yacas documentation (especially from the introductory chapter) but organised differently.

A sample session

> yacas(expression(Factor(x

[1] "Starting Yacas!"

expression((x + 1) * (x - 1))

```
> exp1 <- expression(x^2 + 2 * x^2)
> exp2 <- expression(2 * exp0)
> exp3 <- expression(6 * pi * x)
> exp4 <- expression((exp1 * (1 - sin(exp3)))/exp2)
> yacas(exp4)
```

```
expression(3 * x^2 * (1 - sin(6 * x * pi))/(2 * exp0))
```

```
> yacas("N(Sin(1)^2 + Cos(x)^2)")
```

```
expression(cos(x)^2 + 0.7080734182)
```

```
> yacas("N(Sin(1)^2 + Cos(x)^2)", retclass = "character")
```

```
expression(cos(x)^2 + 0.7080734182)
```

```
> yacas("D(x)Sin(x)")
```

```
expression(cos(x))
```

```
> yacas("Integrate(x,a,b)Sin(x)")
```

```
expression(cos(a) - cos(b))
```

```
> yacas("texp := Taylor(x,0,3) Exp(x)")
```

```
expression(x + x^2/2 + x^3/6 + 1)
```

```
> yacas("PrettyForm(texp)")
```

```
     2    3
    x    x
x + -- + -- + 1
    2    6
<OMOBJ>
  <OMS cd="logic1" name="true"/>
</OMOBJ>
```

```
> yacas("TexForm(texp)")
```

```
$x + \frac{x ^{2}}{2}  + \frac{x ^{3}}{6}  + 1$
<OMOBJ>
  <OMS cd="logic1" name="true"/>
</OMOBJ>
```

```
> yacas("Expand((1+x)^3)")
```

```
expression(x^3 + 3 * x^2 + 3 * x + 1)
```

Yacas calculations.1Yacas calculations.1 1. Simple Yacas calcul

1.2. Setting and clearing a variable

The function Set() and the operator := can both be used to assign values to global variables.

```
> yacas("n := (10 + 2) * 5")

expression(60)

> yacas("n := n+n")

expression(120)

> yacas("Set(z, Cos(a))")

expression(TRUE)

> yacas("z+z")

expression(2 * cos(a))
```

To clear a variable binding execute Clear():

```
> yacas(expression(n))

expression(120)

> yacas("Clear(n)")

expression(TRUE)

> yacas(expression(n))

expression(n)
```

1.3. Symbolic and numerical evaluations, precision

Evaluations are generally exact:

```
> yacas("Exp(0)")

expression(1)

> yacas("Exp(1)")

expression(exp(1))

> yacas("Sin(Pi/4)")

expression(root(1/2, 2))

> yacas("355/113")

expression(355/113)
```

To obtain a numerical evaluation (approximation), the N() function can be used:

```
> yacas("N(Exp(1))")

expression(2.7182818284)

> yacas("N(Sin(Pi/4))")

expression(0.70710678118)

> yacas("N(355/113)")

expression(3.1415929203)
```

The N() function has an optional second argument, the required precision:

```
> yacas("N(355/133,20)")

expression(2.66917293233083)
```

The command Precision(n) can be used to specify that all floating point numbers should have a fixed precision of n digits:

```
> yacas("Precision(5)")

expression(TRUE)

> yacas("N(355/113)")

expression(3.14159)
```

1.4. Rational numbers

Rational numbers will stay rational as long as the numerator and denominator are integers:

```
> yacas(expression(55/10))

expression(11/2)
```

1.5. Complex numbers and the imaginary unit

The imaginary unit is denoted I and complex numbers can be entered as either expressions involving I or explicitly Complex(a,b) for a+ib.

```
> yacas("I^2")

expression(-1)

> yacas("7+3*I")

expression(complex_cartesian(7, 3))
```

```
> yacas("Conjugate(%)")

expression(complex_cartesian(7, -3))

> yacas("Exp(3*I)")

expression(complex_cartesian(cos(3), sin(3)))
```

1.6. Symbolic calculation

Some exact manipulations :

```
> yacas("1/14+5/21*(30-(1+1/2)*5^2)")

expression(-12/7)

> yacas("0+x")

expression(x)

> yacas("x+1*y")

expression(x + y)

> yacas("Sin(ArcSin(alpha))+Tan(ArcTan(beta))")

expression(alpha + beta)
```

% operator.2% operator.21.7. Recall the most recent line - the % operator

The operator % automatically recalls the result from the previous line.

```
> yacas("(1+x)^3")

expression((x + 1)^3)

> yacas("%")

expression((x + 1)^3)

> yacas("z:= %")

expression((x + 1)^3)
```

1.8. PrettyForm and TeXForm

Results can be output on the screen in a more readable form with the function PrettyForm():

```
> yacas("xxx:= (1+x)^2+k^3")

expression((x + 1)^2 + k^3)
```

```
> yacas("PrettyForm(xxx)")

         2    3
( x + 1 )  + k
<OMOBJ>
  <OMS cd="logic1" name="true"/>
</OMOBJ>
```

The output can be exported to TeX with TeXForm(), e.g.

!!!The following gives an error; why?

```
yacas("TeXForm(xxx)")
```

2. Commands

2.1. Factorial

```
> yacas("40!")

expression(8.15915283247898e+47)
```

2.2. Taylor expansions

Expand Exp(x) in three terms around 0 and a:

```
> yacas("Taylor(x,0,3) Exp(x)")

expression(x + x^2/2 + x^3/6 + 1)

> yacas("Taylor(x,a,3) Exp(x)")

expression(exp(a) + exp(a) * (x - a) + (x - a)^2 * exp(a)/2 +
    (x - a)^3 * exp(a)/6)
```

The InverseTaylor() function builds the Taylor series expansion of the inverse of an expression. For example, the Taylor expansion in two terms of the inverse of Exp(x) around x=0 (which is the Taylor expansion of Ln(y) around y=1):

```
> yacas("InverseTaylor(x,0,2)Exp(x)")

expression(x - 1 - (x - 1)^2/2)

> yacas("Taylor(y,1,2)Ln(y)")

expression(y - 1 - (y - 1)^2/2)
```

2.3. Solving equations

Solve equations symbolically with:

!!!This gives an error; why:

```
yacas("Solve(x/(1+x) == a, x)")
Error in parse(file, n, text, prompt) : syntax error in "list (equivalent x"...
yacas("Solve(x^2+x == 0, x)")
```

(Note the use of the == operator, which does not evaluate to anything, to denote an "equation" object.) Solve() is rather limited.

To solve an equation (in one variable) like Sin(x)-Exp(x)=0 numerically taking 0.5 as initial guess and an accuracy of 0.0001 do:

```
> yacas("Newton(Sin(x)-Exp(x),x, 0.5, 0.0001)")

expression(-3.18306)
```

2.4. Expanding polynomials

```
> yacas("Expand((1+x)^3)")

expression(x^3 + 3 * x^2 + 3 * x + 1)
```

2.5. Simplifying an expression

The function Simplify() attempts to reduce an expression to a simpler form.

```
> yacas("(x+y)^3-(x-y)^3")

expression((x + y)^3 - (x - y)^3)

> yacas("Simplify(%)")

expression(6 * (x^2 * y) + 2 * y^3)
```

2.6. Analytical derivatives

Analytical derivatives of functions can be evaluated:

```
> yacas("D(x) Sin(x)")

expression(cos(x))

> yacas("D(x) D(x) Sin(x)")

expression(-sin(x))
```

The D function also accepts an argument specifying how often the derivative has to be taken, e.g:

```
> yacas("D(x,2)Sin(x)")

expression(-sin(x))
```

2.7. Integration

```
> yacas("Integrate(x,a,b)Sin(x)")

expression(cos(a) - cos(b))

> yacas("Integrate(x,a,b)Ln(x)+x")

expression(b * log(b) - b + b^2/2 - (a * log(a) - a + a^2/2))

> yacas("Integrate(x)1/(x^2-1)")

expression(log(2 * (x - 1))/2 - log(2 * (x + 1))/2)

> yacas("Integrate(x)Sin(a*x)^2*Cos(b*x)")

expression((2 * sin(b * x)/b - (sin(-2 * x * a - b * x)/(-2 *
    a - b) + sin(-2 * x * a + b * x)/(-2 * a + b)))/4)
```

2.8. Limits

```
> yacas("Limit(x,0)Sin(x)/x")

expression(1)

> yacas("Limit(n,Infinity)(1+(1/n))^n")

expression(exp(1))

> yacas("Limit(h,0) (Sin(x+h)-Sin(x))/h")

expression(cos(x))
```

2.9. Variable substitution

```
> yacas("Subst(x,Cos(a))x+x")

expression(2 * cos(a))
```

2.10. Solving ordinary differential equations

```
> yacas("OdeSolve(y''==4*y)")

expression(C235 * exp(2 * x) + C239 * exp(-2 * x))

> yacas("OdeSolve(y'==8*y)")

expression(C269 * exp(8 * x))
```

3. Matrices

```
> yacas("E4:={ {u1,u1,0},{u1,0,u2},{0,u2,0} }")

expression(list(list(u1, u1, 0), list(u1, 0, u2), list(0, u2,
    0)))

> yacas("PrettyForm(E4)")

/                        \
| ( u1 ) ( u1 ) ( 0 )    |
|                        |
| ( u1 ) ( 0 )  ( u2 )   |
|                        |
| ( 0 )  ( u2 ) ( 0 )    |
\                        /
<OMOBJ>
  <OMS cd="logic1" name="true"/>
</OMOBJ>
```

Inverse:

```
> yacas("E4i:=Inverse(E4)")

expression(list(list(u2^2/(u1 * u2^2), 0, -(u1 * u2)/(u1 * u2^2)),
    list(0, 0, u1 * u2/(u1 * u2^2)), list(-(u1 * u2)/(u1 * u2^2),
        u1 * u2/(u1 * u2^2), u1^2/(u1 * u2^2))))

> yacas("Simplify(E4i)")

expression(list(list(1/u1, 0, -1/u2), list(0, 0, 1/u2), list(-1/u2,
    1/u2, u1/u2^2)))

> yacas("PrettyForm(Simplify(E4i))")

/                         \
| / 1  \ ( 0 )  / -1 \    |
| | -- |        | -- |    |
| \ u1 /        \ u2 /    |
|                         |
| ( 0 )  ( 0 )  / 1  \    |
|               | -- |    |
|               \ u2 /    |
|                         |
| / -1 \ / 1  \ / u1  \   |
| | -- | | -- | | --- |   |
| \ u2 / \ u2 / |   2 |   |
|               \ u2  /   |
\                         /
<OMOBJ>
  <OMS cd="logic1" name="true"/>
</OMOBJ>
```

Determinant:

```
> yacas("Determinant(E4)")
```

```
expression(-(u1 * u2^2))

> yacas("Determinant(E4i)")

expression(-(u1 * u2 * (u1 * u2^3))/(u1 * u2^2)^3)

> yacas("E4i:=Inverse(E4)")

expression(list(list(u2^2/(u1 * u2^2), 0, -(u1 * u2)/(u1 * u2^2)),
    list(0, 0, u1 * u2/(u1 * u2^2)), list(-(u1 * u2)/(u1 * u2^2),
        u1 * u2/(u1 * u2^2), u1^2/(u1 * u2^2))))

> yacas("Simplify(E4i)")

expression(list(list(1/u1, 0, -1/u2), list(0, 0, 1/u2), list(-1/u2,
    1/u2, u1/u2^2)))

> yacas("Simplify(Determinant(E4i))")

expression(-1/(u1 * u2^2))
```

Note that there are two issues here: The two calculations of the inverse E4i look different - but they reduce to the same after using the Simplify() function. Secondly, there is a problem in calculating the determinant for the first version of E4i while there is not for the second.