

# Ryacas – an R interface to the `yacas` computer algebra system

Rob Goedman,      Gabor Grothendieck,      Søren Højsgaard,  
Ayal Pinkus

September 22, 2006

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>A sample session</b>	<b>2</b>
<b>3</b>	<b>Simple Yacas calculations</b>	<b>3</b>
3.1	Setting and clearing a variable . . . . .	3
3.2	Symbolic and numerical evaluations, precision . . . . .	4
3.3	Rational numbers . . . . .	5
3.4	Symbolic calculation . . . . .	5
3.5	Complex numbers and the imaginary unit . . . . .	5
3.6	Recall the most recent line – the <code>%</code> operator . . . . .	5
3.7	Printing with <code>PrettyForm</code> , <code>PrettyPrint</code> , <code>TeXForm</code> and <code>TeXForm</code> . . . . .	6
<b>4</b>	<b>Commands</b>	<b>7</b>
4.1	Factorial . . . . .	7
4.2	Taylor expansions . . . . .	7
4.3	Solving equations . . . . .	7
4.3.1	Solving equations symbolically . . . . .	7
4.3.2	Solving equations numerically . . . . .	7
4.4	Expanding polynomials . . . . .	8
4.5	Simplifying an expression . . . . .	8
4.6	Analytical derivatives . . . . .	8
4.7	Integration . . . . .	8
4.8	Limits . . . . .	9
4.9	Variable substitution . . . . .	9
4.10	Solving ordinary differential equations . . . . .	9
<b>5</b>	<b>Matrices</b>	<b>9</b>
5.1	Inverse . . . . .	10
5.2	Determinant . . . . .	10

## 1 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 at [yacas.sourceforge.org](http://yacas.sourceforge.org) for various platforms. There is a comprehensive documentation (300+ pages) of `yacas` (also available at [yacas.sourceforge.org](http://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 are organised differently.

## 2 A sample session

Algebraic calculations:

```
> yacas(expression((10 + 2) * 5 + 7^7))  
[1] "Starting Yacas!"  
expression(823603)  
  
> yacas(expression(1/14 + 5/21 * (30 - (1 + 1/2) *  
+      5^2)))  
expression(-12/7)
```

Numerical evaluations:

```
> yacas(expression(N(-12/2)))  
expression(-6)
```

Working with symbolic expressions:

```
> yacas(expression(Factor(x^2 - 1)))  
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))
```

Working with numerical expressions: ....

Combining symbolic and numerical expressions:

```
> yacas("N(Sin(1)^2 + Cos(x)^2)")  
expression(cos(x)^2 + 0.7080734182)  
  
> yacas("N(Sin(1)^2 + Cos(x)^2)", retclass = "character")  
"+" ("^(cos (x ),2 ),0.7080734182 )
```

Differentiation:

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

Integration: [!!! This is odd: I thought yacas was case sensitive...]]]

```

> yacas("A")
expression(A)
> yacas("A:='ssss' ")
expression("'ssss'")
> yacas("a")
expression(a)
> yacas("Integrate(x,a,b)Sin(x)")
In function "Check" : CommandLine(1) : "Found bound variable A which should have been unbound, in MatchLin
> yacas("Clear(A)")
expression(TRUE)
> yacas("A")
expression(A)
> yacas("Integrate(x,a,b)Sin(x)")
expression(cos(a) - cos(b))

```

Expanding polynomials:

```

> yacas("Expand((1+x)^3)")
expression(x^3 + 3 * x^2 + 3 * x + 1)

```

Taylor expansion:

```

> yacas("texp := Taylor(x,0,3) Exp(x)")
expression(x + x^2/2 + x^3/6 + 1)

```

Printing the result in nice forms:

```

> yacas("PrettyForm(texp)")
      2    3
     x    x
x + -- + -- + 1
   2    6
> yacas("TeXForm(texp)", retclass = "unquote")
$x + \frac{x ^{2}}{2} + \frac{x ^{3}}{6} + 1$

```

## 3 Simple Yacas calculations

### 3.1 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)

```

### 3.2 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(355/113, 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)
```

### 3.3 Rational numbers

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

```
> yacas(expression(55/10))
expression(11/2)
```

### 3.4 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)
```

### 3.5 Complex numbers and the imaginary unit

The imaginary unit  $i$  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)))
```

### 3.6 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)

```

### 3.7 Printing with PrettyForm, PrettyPrint, TexForm and TeX-Form

There are different ways of displaying the output. The (standard) yacas form is:

```

> yacas("A:={{a,b},{c,d}}")
expression(list(list(a, b), list(c, d)))
> yacas("B:= (1+x)^2+k^3")
expression((x + 1)^2 + k^3)
> yacas("A")
expression(list(list(a, b), list(c, d)))
> yacas("B")
expression((x + 1)^2 + k^3)

```

The Pretty form is:

```

> yacas("PrettyForm(A)")
/
| ( a ) ( b ) |
|             |
| ( c ) ( d ) |
\             /

> yacas("PrettyForm(B)")
      2    3
( x + 1 ) + k

```

An alternative is the PrettyPrinter [!!! Why does this give the same result as before??? Earlier I got XML output as well... Is something not reset???

```

> yacas("PrettyPrinter(\"PrettyForm\")")
True
> yacas("A")
/
| ( a ) ( b ) |
|             |
| ( c ) ( d ) |
\             /

> yacas("PrettyPrinter()")
True;
> yacas("A")
{{a,b},{c,d}};

```

The output can be displayed in TeX form as well:

```
> yacas("TeXForm(B)")
"$\left( x + 1\right) ^{2} + k ^{3}$";
> yacas("TeXForm(B)")

$$\left( x + 1 \right) ^{2} + k ^{3}$$

```

## 4 Commands

### 4.1 Factorial

```
> yacas("40!")
815915283247897734345611269596115894272000000000;
```

### 4.2 Taylor expansions

Expand  $\text{Exp}(x)$  in three terms around 0 and a:

```
> yacas("Taylor(x,0,3) Exp(x)")
x+x^2/2+x^3/6+1;
> yacas("Taylor(x,a,3) Exp(x)")
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  $\text{Exp}(x)$  around  $x=0$  (which is the Taylor expansion of  $\text{Ln}(y)$  around  $y=1$ ):

```
> yacas("InverseTaylor(x,0,2)Exp(x)")
x-1-(x-1)^2/2;
> yacas("Taylor(y,1,2)Ln(y)")
y-1-(y-1)^2/2;
```

### 4.3 Solving equations

#### 4.3.1 Solving equations symbolically

Solve equations symbolically with:

```
> yacas("Solve(x/(1+x) == a, x)")
{x==a/(1-a)};
> yacas("Solve(x^2+x == 0, x)")
{x==0,x==(-1)};
```

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

#### 4.3.2 Solving equations numerically

To solve an equation (in one variable) like  $\text{Sin}(x)-\text{Exp}(x)=0$  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)")
-3.18306;
```

## 4.4 Expanding polynomials

```
> yacas("Expand((1+x)^3)")  
x^3+3*x^2+3*x+1;
```

## 4.5 Simplifying an expression

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

```
> yacas("(x+y)^3-(x-y)^3")  
(x+y)^3-(x-y)^3;  
> yacas("Simplify(%)")  
6*x^2*y+2*y^3;
```

## 4.6 Analytical derivatives

Analytical derivatives of functions can be evaluated:

```
> yacas("D(x) Sin(x)")  
Cos(x);  
> yacas("D(x) D(x) Sin(x)")  
-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)")  
-Sin(x);
```

## 4.7 Integration

!!! Problem arises because A was defined above (a is not defined, though)

```
> yacas("Integrate(x,a,b)Sin(x)")  
In function "Check" : CommandLine(1) : "Found bound variable A which should have been unbound, in MatchLin  
> yacas("Integrate(x,a,b)Ln(x)+x")  
In function "Check" : CommandLine(1) : "Found bound variable A which should have been unbound, in MatchLin  
> yacas("Integrate(x)1/(x^2-1)")  
In function "Check" : CommandLine(1) : "Found bound variable A which should have been unbound, in MatchLin  
> yacas("Integrate(x)Sin(a*x)^2*cos(b*x)")  
In function "Check" : CommandLine(1) : "Found bound variable A which should have been unbound, in MatchLin
```



## 4.8 Limits

```
> yacas("Limit(x,0)Sin(x)/x")
1;
> yacas("Limit(n,Infinity)(1+(1/n))^n")
Exp(1);
> yacas("Limit(h,0) (Sin(x+h)-Sin(x))/h")
Cos(x);
```

## 4.9 Variable substitution

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

## 4.10 Solving ordinary differential equations

```
> yacas("OdeSolve(y' ==4*y)")
C254*Exp(2*x)+C258*Exp((-2)*x);
> yacas("OdeSolve(y' ==8*y)")
C288*Exp(8*x);
```

## 5 Matrices

```
> yacas("E4:={ {u1,u1,0},{u1,0,u2},{0,u2,0} }")
{{u1,u1,0},{u1,0,u2},{0,u2,0}};
> yacas("PrettyForm(E4)")
/
| ( u1 ) ( u1 ) ( 0 ) |
|
| ( u1 ) ( 0 ) ( u2 ) |
|
| ( 0 ) ( u2 ) ( 0 ) |
\
```

## 5.1 Inverse

```

> yacas("E4i:=Inverse(E4)")
{{u2^2/(u1*u2^2),0,(-u1*u2)/(u1*u2^2)},{0,0,(u1*u2)/(u1*u2^2)},{(-u1*u2)/(u1*u2^2),(u1*u2)/(u1*u2^2),u1^2/
> yacas("Simplify(E4i)")
{{1/u1,0,(-1)/u2},{0,0,1/u2},{(-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 / |
| | | | |
\ | | | | /

```

## 5.2 Determinant

```

> yacas("Determinant(E4)")
-u1*u2^2;
> yacas("Determinant(E4i)")
(-u1*u2*u1*u2^3)/(u1*u2^2)^3;
> yacas("Simplify(E4i)")
{{1/u1,0,(-1)/u2},{0,0,1/u2},{(-1)/u2,1/u2,u1/u2^2}};
> yacas("Simplify(Determinant(E4i))")
(-1)/(u1*u2^2);

```