# integral

Save | Save & quit | Discard & quit

last edited Nov 23, 2015, 2:27:55 PM by admin

File... | Action... | Data... | sage | Typeset | Load 3-D Live | Use java for 3-D

Print | Worksheet | Edit | Text | Revisions | Share | Publish

```
y=var('y')
innerint=integral(sqrt(x),x,y,e^y)
show(innerint)
```

$$-\frac{2}{3}y^{\frac{3}{2}} + \frac{2}{3}e^{\left(\frac{3}{2}y\right)}$$

```
out= integral(innerint,y,0,1)
out
```

```
4/9*e^(3/2) - 32/45
```

examle2

```
test=integral(e^(x/y),x,y^3,y)
test
```

```
y*e - y*e^(y^2)
```

```
b=integral(test,y,0,1)
show(b)
```

$$\frac{1}{2}$$

```
out2= integral((y*e),y,0,1)
show(out2)
```

$$\frac{1}{2}e$$

```
a=integral(y*e^(y^2),y,0,1)
show(a)
```

$$\frac{1}{2}e - \frac{1}{2}$$

```
out2-a
```

```
1/2
```

```
out3=integral(y*e - y*e^(y^2),y,0,1)
show(out3)
```

$$\frac{1}{2}$$

```
z=var('z')
a1=integral(1,z,0,(1/2)*(6-6*x-3*y))
mida=integral(a1,y,0,-2*x+2)
a3=integral(mida,x,0,1)
a3
```

```
1
```

```
t1=integral(sin(x)*cos(sin(y)),x,y,pi/2)
t1
```

```
cos(y)*cos(sin(y))
```

```
midt=integral(t1,y,0,pi/2)
midt
```

```
sin(1)
```

```
t3=integral(midt,z,0,pi)
show(t3)
```

$$\pi \sin(1)$$

```
numerical_integral(x^2,0,1)
```

```
(0.3333333333333333, 3.700743415417188e-15)
```

```
numerical_integral(x^2,0,1,max_points=10)
```

```
(0.3333333333333333, 3.700743415417188e-15)
```

```
numerical_integral(sin(x)^3+sin(x),0,pi)
```

```
(3.333333333333333, 3.700743415417188e-14)
```

```
integral(sin(x)^3+sin(x),0,pi)
```

```
10/3
```

```
10/3.
```

```
3.33333333333333
```

```
numerical_integral(lambda x:sin(x)^3+sin(x),0,pi)
```

```
(3.333333333333333, 3.700743415417188e-14)
```

```
(sin(x)^3+sin(x)).integral(x,0,pi)
```

```
10/3
```

```
numerical_integral(2,1,7)
```

```
(12.0, 0.0)
```

```
exp(-1/x).nintegral(x,1,2)
```

```
(0.504792217873184, 5.604319429344074e-15, 21, 0)
```

```
numerical_integral(exp(-1/x),1,2)
```

```
(0.5047922178731841, 5.6043194293440744e-15)
```

```
numerical_integral(log,0,0)
```

```
(0.0, 0.0)
```

```
numerical_integral(log,1,0)
```

```
Traceback (click to the left of this block for traceback)
...
TypeError: a float is required
```

```
numerical_integral(log,5,5)
```

```
(0.0, 0.0)
```

```
numerical_integral(log,-1,-1)
```

```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),(-2,-2))
```

```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),-2,-2)
```

```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),-2,-1)
```

```
(nan, nan)
```

```
f,g=x,x-1
numerical_integral(f-g,-2,2)
```

```
(4.0, 0.0)
```

```
w,u=x^2,x
numerical_integral(w-u,-2,2)
```

```
(5.333333333333335, 6.289076342569853e-14)
```

```
w,u=x^2,1/(x^2)
numerical_integral(w*u,-2,2)
```

```
(4.0, 0.0)
```

```
numerical_integral(SR(2),5,10)
```

```
(10.0, 0.0)
```

```
numerical_integral(SR(10),5,10)
```

```
(50.0, 0.0)
```

```
numerical_integral(SR(20),5,10)
```

```
(100.0, 0.0)
```

```
numerical_integral(SR(10),10,10)
```

```
(0.0, 0.0)
```

```
numerical_integral(SR(10),20,10)
```

```
(-100.0, 0.0)
```

```
numerical_integral(SR(10),5,10)
```

```
numerical_integral(SR(1+3j),2,3)
```

```
Traceback (click to the left of this block for traceback)
...
TypeError: unable to simplify to float approximation
```

```
SR?
```

**File:** /home/sage/sage-6.8/src/sage/symbolic/ring.pyx

**Type:** <type 'sage.symbolic.ring.SymbolicRing'>

**Definition:** SR(x=0, *args, **kwds)

**Docstring:**

```
    Symbolic Ring, parent object for all symbolic expressions.
```

# integral

Save | Save & quit | Discard & quit

last edited Nov 29, 2015, 2:25:27 AM by admin

File... | Action... | Data... | sage | Typeset | Load 3-D Live | Use java for 3-D

Print | Worksheet | Edit | Text | Revisions | Share | Publish

symbolic function integral

```
f(x,y,z)=x*y/z+sin(z)
integral(f,z)
```

```
(x, y, z) |--> x*y*log(z) - cos(z)
```

y=0, x=1,x=y

```
a=integral(integral(e^(y^2),x,0,y),y,0,1)
a
```

```
1/2*e - 1/2
```

```
var('y')
a=integral(integral(e^(x*y),x,0,y),y,0,1)
a
```

```
integrate(e^(y^2)/y - 1/y, y, 0, 1)
```

```
a=integral(integral(e^(x^2+y^2),x,0,y),y,0,1)
a
```

```
Traceback (click to the left of this block for traceback)
...
Is y positive, negative or zero?
```

```
assume(y>0)
a=integral(integral(e^(x^2+y^2),x,0,y),y,0,1)
a
```

```
-1/8*pi*erf(I)^2
```

```
float(a)
```

```
0.8591409142295225
```

x=y^4, (1,-1) to (1,1)

```
t=var('t')
(x,y)=(t^4,t)
(dx,dy)=(diff(x,t),diff(y,t))
integral(sin(x)*dx,t,-1,1)
```

```
0
```

```
a=integral(sin(y),x,y^4,1)
a
```

```
-(y^4 - 1)*sin(y)
```

```
integral(a,y,-1,1)
```

```
0
```

```
restore('x,y')
```

```
log(x)==ln(x)
```

```
log(x) == log(x)
```

```
forget()
integral(e^((-x)^2)*log(x),x)
```

```
integrate(e^(x^2)*log(x), x)
```

```
integral(e^((-x)^2)*ln(x),x,0,1)
```

```
integrate(e^(x^2)*log(x), x)
```

```
integral(ln(x),x)
```

```
x*log(x) - x
```

```
integral(e^((-x)^2),x,0,oo)
```

```
integrate(e^(x^2), x, 0, +Infinity)
```

assume

```
var('a')
integral(abs(x),x,0,a)
```

```
1/2*a*abs(a)
```

```
integral(abs(x)*x,x,0,a)
```

```
Traceback (click to the left of this block for traceback)
...
Is a positive, negative or zero?
```

```
assume(a>0)
integral(abs(x)*x,x,0,a)
```

```
1/3*a^3
```

```
forget()
```

start example of multiple integrals

```
y=var('y')
innerint=integral(sqrt(x),x,y,e^y)
show(innerint)
```

$$-\frac{2}{3}y^{\frac{3}{2}} + \frac{2}{3}e^{(\frac{3}{2}y)}$$

```
out= integral(innerint,y,0,1)
out
```

```
4/9*e^(3/2) - 32/45
```

examle2

```
integral(e^(1/x),x)
```

```
x*e^(1/x) - Ei(1/x)
```

```
ti=integral(e^(x/y),y,x,x^(1/3))
ti
```

```
x*(Ei(1) - e) - x*Ei(x^(2/3)) + x^(1/3)*e^(x^(2/3))
```

```
t2=integral(ti,x,0,1)
t2
```

```
1/2
```

ba taqeer tartib

```
test=integral(e^(x/y),x,y^3,y)
test
```

```
y*e - y*e^(y^2)
```

```
b=integral(test,y,0,1)
show(b)
```

$$\frac{1}{2}$$

Ei

```
numerical_integral(e^(1/x),0,1)
```

```
(inf, nan)
```

```
integral(e^(1/x))
```

```
x*e^(1/x) - Ei(1/x)
```

```
Ei?
```

**File:** /home/sage/sage-6.8/local/lib/python2.7/site-packages/sage/functions/exp_integ

**Type:** <class 'sage.functions.exp_integral.Function_exp_integral'>

**Definition:** Ei(coerce=True, hold=False, dont_call_method_on_arg=False, *args)

**Docstring:**

The generalized complex exponential integral Ei(z) defined by

$$\mathrm{Ei}(x) = \int_{-\infty}^{x} \frac{e^t}{t}\,dt$$

for x > 0 and for complex arguments by analytic continuation, see [AS] 5.1.2.

EXAMPLES:

```
sage: Ei(10)
Ei(10)
sage: Ei(I)
Ei(I)
sage: Ei(3+I)
Ei(I + 3)
sage: Ei(1.3)
2.72139888023202
sage: Ei(10r)
Ei(10)
sage: Ei(1.3r)
2.7213988802320235
```

The branch cut for this function is along the negative real axis:

```
sage: Ei(-3 + 0.1*I)
```

```
-0.0129379427181693 + 3.13993830250942*I
sage: Ei(-3 - 0.1*I)
-0.0129379427181693 - 3.13993830250942*I
```

The precision for the result is deduced from the precision of the input. Convert t a result with higher precision is desired:

```
sage: Ei(RealField(300)(1.1))
2.16737827956340282358378734233807621497112737591639704719490
```

ALGORITHM: Uses mpmath.

TESTS:

Show that the evaluation and limit issue in trac ticket #13271 is fixed:

```
sage: var('Z')
Z
sage: (Ei(-Z)).limit(Z=oo)
0
sage: (Ei(-Z)).limit(Z=1000)
Ei(-1000)
sage: (Ei(-Z)).limit(Z=1000).n()
-5.07089306023517e-438
```

```
tw=integral(cos(sin(y)),y)
tw
```

```
integrate(cos(sin(y)), y)
```

```
numerical_integral(cos(sin(y)),0,x)
```

```
Traceback (click to the left of this block for traceback)
...
TypeError: unable to simplify to float approximation
```

```
numerical_integral(cos(sin(y)),0,pi/2)
```

```
(1.2019697153172064, 1.3344544528473072e-14)
```

2gane

```
q1=integral(sin(x)*cos(sin(y)),y)
q1
```

```
integrate(cos(sin(y)), y)*sin(x)
```

```
q1=integral(sin(x)*cos(sin(y)),x,y,pi/2)
q1
```

```
cos(y)*cos(sin(y))
```

```
q2=integral(q1,y,0,pi/2)
q2
```

```
sin(1)
```

```
q3=integral(q2,z,0,pi)
q3
```

```
pi*sin(1)
```

example3

example

```
h1=integral(2*x*e^y/y,x,y,y^(1/2))
h1
```

```
-(y^2 - y)*e^y/y
```

```
h2=integral(h1,y,0,1)
h2
```

```
e - 2
```

example4

```
qt=integral(x/((1+x*y)^2),x,y,1)
qt
```

```
Traceback (click to the left of this block for traceback)
...
Is y*(y+1) positive, negative or zero?
```

```
assume( (y*(y+1))>0)
qt=integral(x/((1+x*y)^2),x,y,1)
qt
```

```
Traceback (click to the left of this block for traceback)
...
Is y*(y+1) positive, negative or zero?
```

```
qt=integral(x/((1+x*y)^2),y,0,x)
qt
```

```
-x*(1/(x^3 + x) - 1/x)
```

```
qt1=integral(qt,x,0,1)
qt1
```

```
-1/4*pi + 1
```

example5

```
z1=integral(sqrt(1-y^4),y,x,x^(1/3))
z1
```

```
integrate(sqrt(-y^4 + 1), y, x, x^(1/3))
```

```
z2=integral(sqrt(1-y^4),x,y^3,y)
z2
```

```
-sqrt(-y^4 + 1)*(y^3 - y)
```

```
z3=integral(z2,y,0,1)
z3
```

```
1/8*pi - 1/6
```

```
integral(cos(sin(y)),y,0,1)
```

```
integrate(cos(sin(y)), y, 0, 1)
```

```
numerical_integral(cos(sin(y)),0,1)
```

```
(0.8687400395769761, 9.6449519435212 86e-15)
```

```

```

```
numerical_integral(x^2,0,1)
```

```
(0.3333333333333333, 3.700743415417188e-15)
```

```
numerical_integral(x^2,0,1,max_points=10)
```

```
(0.3333333333333333, 3.700743415417188e-15)
```

```
numerical_integral(sin(x)^3+sin(x),0,pi)
```

```
(3.333333333333333, 3.700743415417188e-14)
```

```
integral(sin(x)^3+sin(x),0,pi)
```

```
10/3
```

```
10/3.
```

```
3.33333333333333
```

```
numerical_integral(lambda x:sin(x)^3+sin(x),0,pi)
```

```
(3.333333333333333, 3.700743415417188e-14)
```

```
(sin(x)^3+sin(x)).integral(x,0,pi)
```

```
10/3
```

```
numerical_integral(2,1,7)
```
```
(12.0, 0.0)
```

```
exp(-1/x).nintegral(x,1,2)
```
```
(0.504792217873184, 5.604319429344074e-15, 21, 0)
```

```
numerical_integral(exp(-1/x),1,2)
```
```
(0.5047922178731841, 5.6043194293440744e-15)
```

```
numerical_integral(log,0,0)
```
```
(0.0, 0.0)
```

```
numerical_integral(log,1,0)
```
```
Traceback (click to the left of this block for traceback)
...
TypeError: a float is required
```

```
numerical_integral(log,5,5)
```
```
(0.0, 0.0)
```

```
numerical_integral(log,-1,-1)
```
```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),(-2,-2))
```
```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),-2,-2)
```
```
(0.0, 0.0)
```

```
integral(sqrt(x),x,1,2)
```
```
4/3*sqrt(2) - 2/3
```

```
numerical_integral(sqrt(x),-1,-1)
```
```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),-2,-2)
```
```
(0.0, 0.0)
```

```
numerical_integral(sqrt(x),1,2)
```
```
(1.21895141649746, 1.3533079284949289e-14)
```

```
numerical_integral(sqrt(x),2,1)
```

```
    (-1.21895141649746, 1.3533079284949289e-14)
```

```
integral(sqrt(x),2,1)
```
```
    -4/3*sqrt(2) + 2/3
```

```
integral(sqrt(x),1,2)
```
```
    4/3*sqrt(2) - 2/3
```

```
4/3*sqrt(2.)
```
```
    1.88561808316413
```

```
ee=sqrt(2.)
ee
```
```
    1.41421356237310
```

```
4/3*ee
```
```
    1.88561808316413
```

```
2/3.
```
```
    0.666666666666667
```

```
1.88561808316413-0.666666666666667
```
```
    1.21895141649746
```

```
f,g=x,x-1
numerical_integral(f-g,-2,2)
```
```
    (4.0, 0.0)
```

```
w,u=x^2,x
numerical_integral(w-u,-2,2)
```
```
    (5.333333333333335, 6.289076342569853e-14)
```

```
w,u=x^2,1/(x^2)
numerical_integral(w*u,-2,2)
```
```
    (4.0, 0.0)
```

```
numerical_integral(SR(1),5,10)
```
```
    (5.0, 0.0)
```

```
SR?
```

**File:** /home/sage/sage-6.8/src/sage/symbolic/ring.pyx

**Type:** <type 'sage.symbolic.ring.SymbolicRing'>

**Definition:** SR(x=0, *args, **kwds)

**Docstring:**

```
    Symbolic Ring, parent object for all symbolic expressions.
```

```
SR(5.2)
```
```
5.20000000000000
```

```
SR(10)
```
```
10
```

```
numerical_integral(10,5,10)
```
```
(50.0, 0.0)
```

```
numerical_integral(SR(20),5,10)
```
```
(100.0, 0.0)
```

```
numerical_integral(SR(10),10,10)
```
```
(0.0, 0.0)
```

```
numerical_integral(SR(10),20,10)
```
```
(-100.0, 0.0)
```

```
SR(1+3j)
```
```
1.00000000000000 + 3.00000000000000*I
```

```
numerical_integral(SR(1+3j),2,3)
```
```
Traceback (click to the left of this block for traceback)
...
TypeError: unable to simplify to float approximation
```

```
SR?
```

**File:** /home/sage/sage-6.8/src/sage/symbolic/ring.pyx

**Type:** <type 'sage.symbolic.ring.SymbolicRing'>

**Definition:** SR(x=0, *args, **kwds)

**Docstring:**

```
    Symbolic Ring, parent object for all symbolic expressions.
```