Sage comme calculatrice

```
>>> 1+1
```

```
2
```

```
>>> factorial(50)
```

```
30414093201713378043612608166064768844377641568960512000000000000
```

```
>>> s = add(1/n for n in (1..30))
```

```
>>> s
```

```
9304682830147/2329089562800
```

```
>>> s.n()
```

```
3.99498713092039
```

```
>>>
```

Python

```
>>> a = -2/3
```

```
>>> type(a)
```

```
<type 'sage.rings.rational.Rational'>
```

```
>>> dir(a)[140:160]
```

```
['_singular_',
'_singular_init_',
'_sub_',
'_sympy_',
'_test_category',
'_test_eq',
'_test_nonzero_equal',
'_test_not_implemented_methods',
'_test_pickling',
'_tester',
'abs',
'absolute_norm',
'additive_order',
'base_extend',
'base_ring',
'cartesian_product',
'category',
'ceil',
'charpoly',
'conjugate']
```

```
>>> a.abs(), abs(a)
```

```
(2/3, 2/3)
```

```
>>> def myfact(n):
        res = 1
        for k in srange(1, n+1):
            res = res*k
        return res
```

```
>>> myfact(50)
```

```
30414093201713378043612608166064768844377641568960512000000000000
```

```
>>>
```

Calcul symbolique à la Maple

```
>>> x, y = var('x', 'y')
    u = cos(x)*sin(y)
    u
```

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

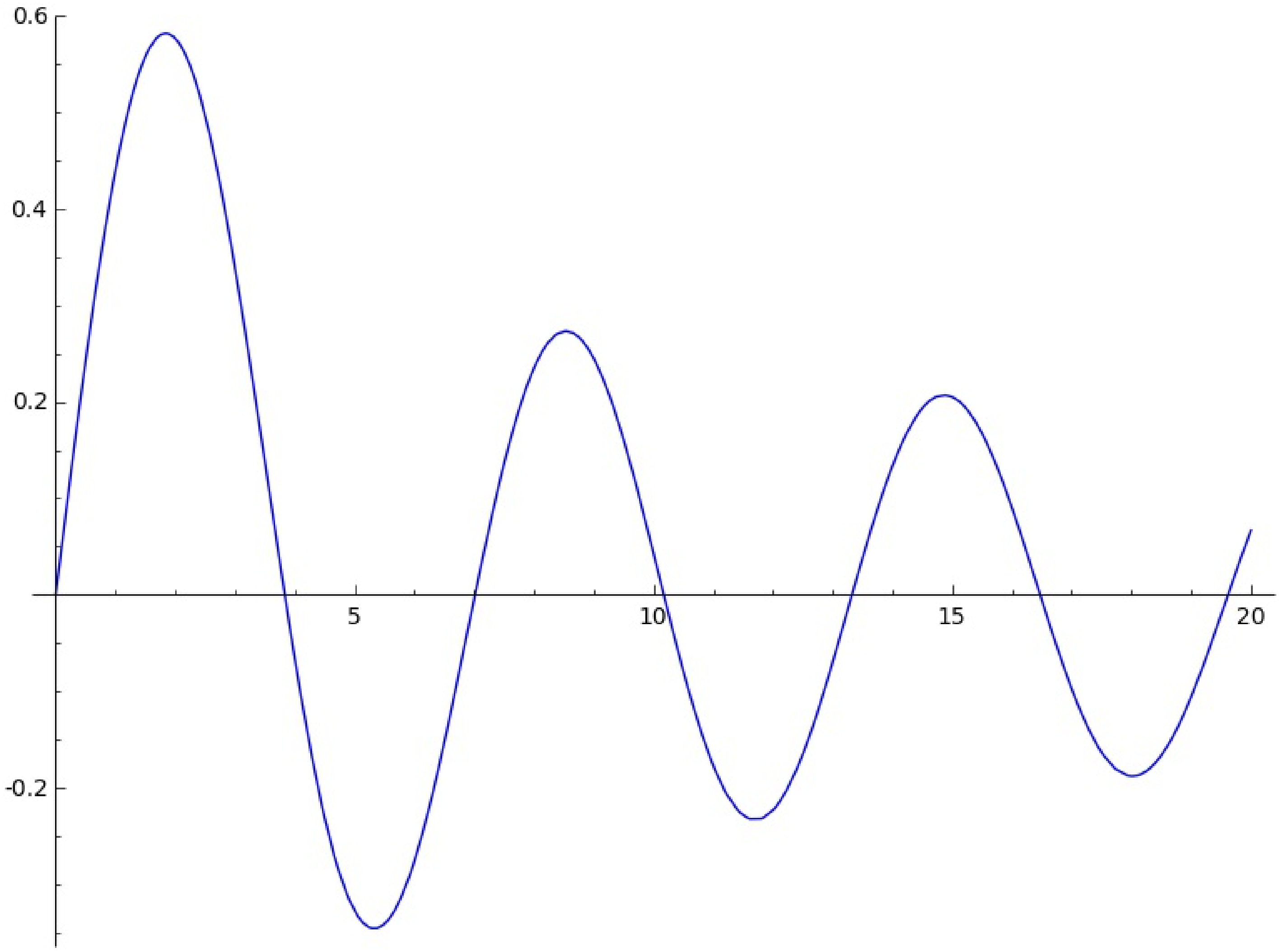
```
>>> diff(u, x)
```

```
-sin(x)*sin(y)
```

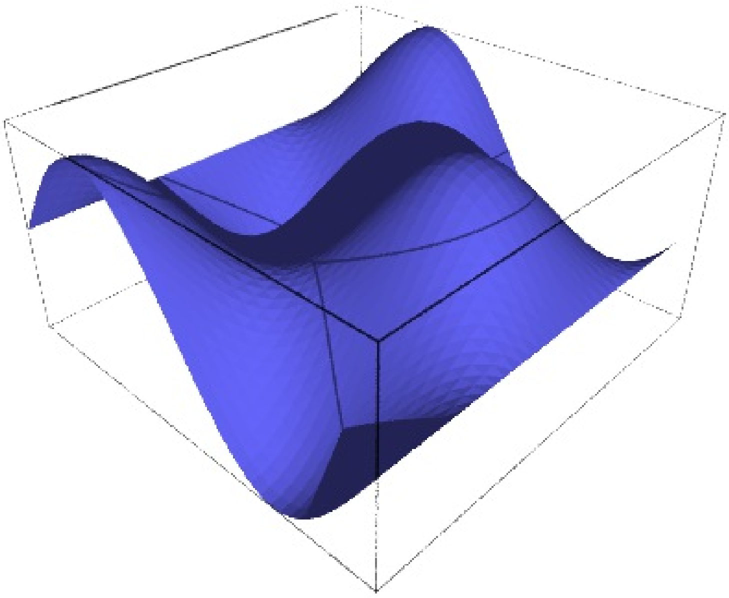
```
>>>   u.derivative(y)

      cos(x)*cos(y)

>>>   u.series(x, order=5)

      (sin(y)) + (-1/2*sin(y))*x^2 + (1/24*sin(y))*x^4 + Order(x^5)

>>>   u(x=1)

      cos(1)*sin(y)

>>>   u(x=sqrt(2), y=1).n(prec=1000)

      0.131222094408801673541794663949435572581514799978818130494047391392750052887318298141843711494656
      04288553757327376174379326766605682621211798986525637145053119092297681960434525400692467056441133
      781847596004378690230103677152626720456712901449325329568055173470137249434273963974796978559495522
      11331345
>>>
```

Graphiques

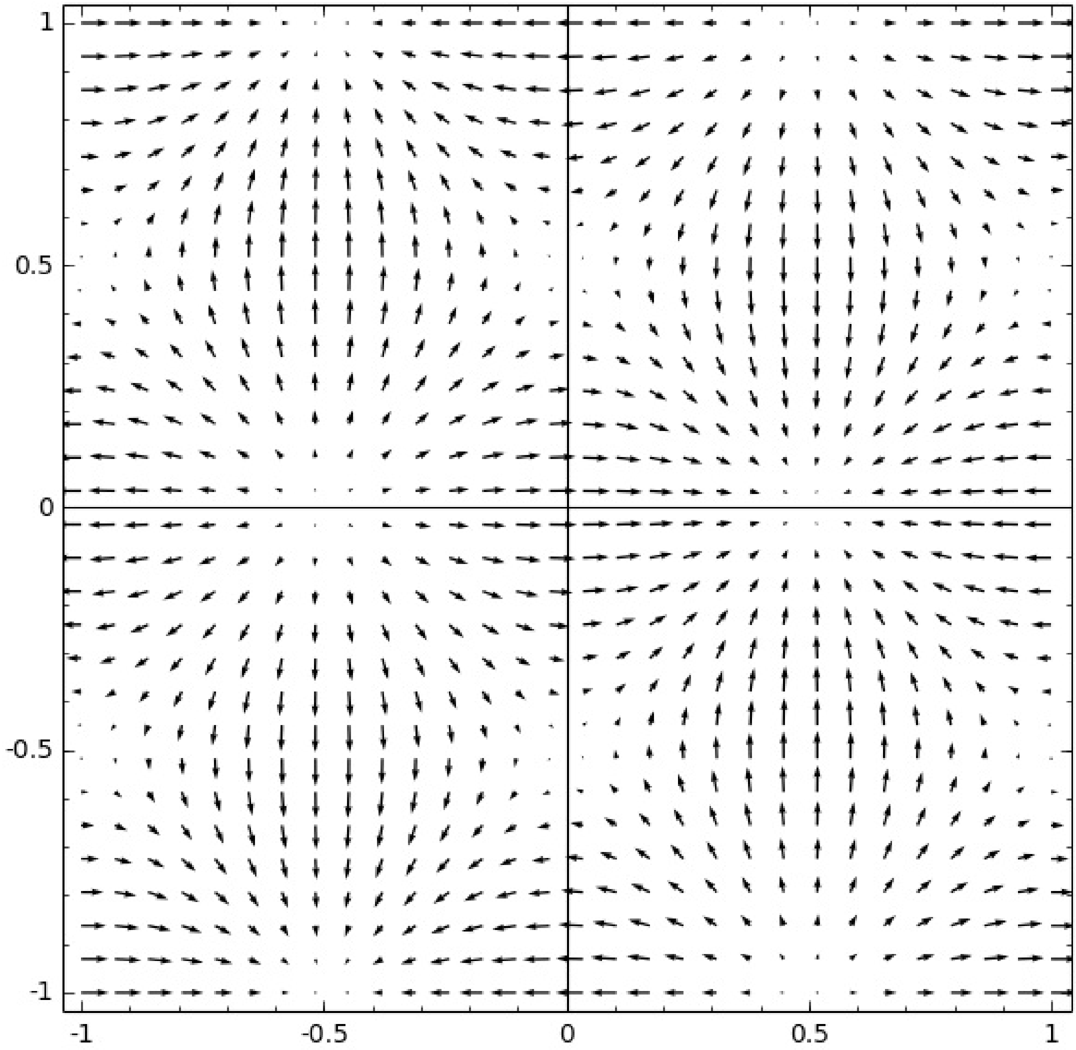
```
>>>   plot(Bessel(1,'J'), 0, 20)
```



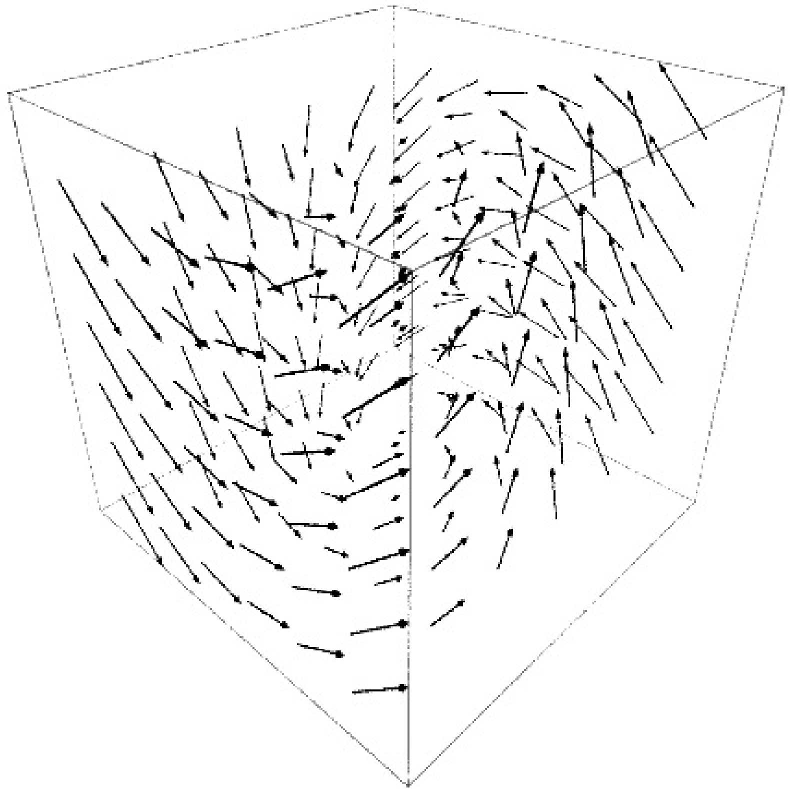
```
>>>   u, v = var('u', 'v')
      f = sin(pi*u) * cos(pi*v)
      plot3d(f, (u, -1, 1), (v, -1, 1))
```



```
>>>   plot_vector_field(f.gradient(), (u,-1,1), (v,-1,1), aspect_ratio=1,
      plot_points=30)
```



```
>>>   x,y,z=var('x y z')
      myplot = plot_vector_field3d(
      (-y, 0, x), (x,-2,2), (y,-2,2), (z,-2,2),
      colors='black', plot_points=6)
      show(myplot, viewer="tachyon")
```



```
>>>
```

Éléments et parents

```
>>>   type(1)
      <type 'sage.rings.integer.Integer'>
>>>   parent(1)
      Integer Ring
>>>   ZZ
      Integer Ring
>>>   parent(1) is ZZ
      True
>>>   parent(1/1)
      Rational Field
>>>   type(ZZ)
      <type 'sage.rings.integer_ring.IntegerRing_class'>
>>>   type(QQ)
      <class 'sage.rings.rational_field.RationalField_with_category'>
>>>   ZZ.cardinality()
      +Infinity
>>>   MyParent = QQ.cartesian_product(ZZ); MyParent
      The cartesian product of (Rational Field, Integer Ring)
```

```
>>> MyParent.is_ring()
True
>>> MyParent.an_element()
(1/2, 1)
>>> QQ.category()
Category of quotient fields
>>>
```

Quelques parents

```
>>> Integers()
Integer Ring
>>> Rationals()
Rational Field
>>> R = IntegerModRing(10); R
Ring of integers modulo 10
>>> R(8)^2
4
>>> Reals()
Real Field with 53 bits of precision
>>> Complexes()
Complex Field with 53 bits of precision
>>> MatrixSpace(RDF, 2, 3)
Full MatrixSpace of 2 by 3 dense matrices over Real Double Field
>>> PolynomialRing(QQ, 'x')
Univariate Polynomial Ring in x over Rational Field
>>> PolynomialRing(QQ, 'x, y, z')
Multivariate Polynomial Ring in x, y, z over Rational Field
>>> MatrixSpace(PolynomialRing(ZZ, 'x'), 2)
Full MatrixSpace of 2 by 2 dense matrices over Univariate Polynomial Ring in x over Integer Ring
>>> _.random_element()
[ -x^2 -x]
[ -6*x^2 - 3*x - 3 -16*x^2 + 2*x + 4]
>>>
```

Conversions

```
>>> RDF
Real Double Field
>>> RDF(42)
42.0
>>> RDF(42).parent()
Real Double Field
>>> ZZ(1.0)
1
```

```
>>> ZZ(1.5)

    ---------------------------------------------------------------------------
    TypeError Traceback (most recent call last)
    <ipython-input-69-562488d24cf3> in <module>()
    ----> 1 ZZ(RealNumber('1.5'))

    /home/marc/co/sage/local/lib/python2.7/site-packages/sage/structure/parent.so in
    sage.structure.parent.Parent.__call__ (build/cythonized/sage/structure/parent.c:9603)()

    /home/marc/co/sage/local/lib/python2.7/site-packages/sage/structure/coerce_maps.so in
    sage.structure.coerce_maps.NamedConvertMap._call_
    (build/cythonized/sage/structure/coerce_maps.c:5577)()

    /home/marc/co/sage/local/lib/python2.7/site-packages/sage/rings/real_mpfr.so in
    sage.rings.real_mpfr.RealNumber._integer_ (build/cythonized/sage/rings/real_mpfr.c:15923)()

    TypeError: Attempt to coerce non-integral RealNumber to Integer

>>>
```

Coercitions

(= conversions canoniques automatiques)

```
>>> a = 42
    a, a.parent()

    (42, Integer Ring)

>>> b = a + 1/2
    b, b.parent()

    (85/2, Rational Field)

>>> c = b + 1/2
    (c, c.parent())

    (43, Rational Field)

>>> d = ZZ(c)
    (d, d.parent())

    (43, Integer Ring)

>>> M = MatrixSpace(ZZ, 3); M

    Full MatrixSpace of 3 by 3 dense matrices over Integer Ring

>>> obj = M.identity_matrix() + 1/2

>>> obj

    [3/2 0 0]
    [ 0 3/2 0]
    [ 0 0 3/2]

>>> obj.parent()

    Full MatrixSpace of 3 by 3 dense matrices over Rational Field

>>>
```

Changement de parent

```
>>> M = MatrixSpace(RDF, 2)

>>> M

    Full MatrixSpace of 2 by 2 dense matrices over Real Double Field

>>> mat = M.random_element(); mat

    [ -0.9291095239848672 0.7144542993819145]
    [-0.4285627437025185 0.6202965684624147]

>>> mat.inverse()

    [-2.2962684025138707 2.6448297735670647]
    [-1.5865043339815423 3.439459366492412]
```

```
>>> MatrixSpace(Reals(100), 2)(mat)

[-0.92910952398486723602388792642 0.71445429938191451491036332300]
[-0.42856627437025185400898408261 0.62029656846241465473212883808]

>>> mat.change_ring(Reals(100)).inverse()

[-2.2962684025138705309483785961 2.6448297735670647696478834303]
[-1.5865043339815423680868343285 3.4394593664924119799809269914]

>>> mat.change_ring(RealIntervalField(100)).inverse()

[-2.2962684025138705309483785962? 2.6448297735670647696478834304?]
[-1.5865043339815423680868343285? 3.4394593664924119799809269915?]

>>> mat.change_ring(QQ).inverse()

[-13962576660516606866441011513656 09/6080550795033757319974108878776 20
1086623093404875740276372284624 8/4108480266914700891874397891065]
[ -4823410094657982469766952928449 3/3040275397516878659987054439388 1
2826190187217801667329077965702/8216960533829401783748795782 13]

>>>
```

Flottants, intervalles...

```
>>> RDF

Real Double Field

>>> RDF.precision()

53

>>> x = RDF(1/3)

>>> sin(x)

0.3271946967961522

>>> x * 2^1024

+infinity

>>> RR

Real Field with 53 bits of precision

>>> y = RR(1/3)

>>> y * 2^1024

5.99231044954105e307

>>> QQ(y)

1/3

>>> y.exact_rational()

6004799503160661/18014398509481984

>>> y.sign_mantissa_exponent()

(1, 6004799503160661, -54)

>>> sqrt(-x)

0.5773502691896257*I

>>> sqrt(-x).parent()

Complex Double Field

>>> sqrt(-y).parent()

Complex Field with 53 bits of precision

>>> Reals(prec=230, rnd='RNDU')

Real Field with 230 bits of precision and rounding RNDU
```

```
>>> Reals(prec=230, rnd='RNDZ')

    Real Field with 230 bits of precision and rounding RNDZ

>>> RealIntervalField(prec=100)

    Real Interval Field with 100 bits of precision

>>> ComplexIntervalField(prec=100)

    Complex Interval Field with 100 bits of precision

>>> RIF, CIF

    (Real Interval Field with 53 bits of precision,
    Complex Interval Field with 53 bits of precision)

>>> 0.42*1/3

    0.140000000000000

>>> 0.420000000000000000000000000000000*1/3

    0.140000000000000000000000000000000

>>> type(0.42)

    <type 'sage.rings.real_mpfr.RealLiteral'>

>>> RR(1/3) + RIF(1/3)

    0.6666666666666667?

>>> RLF

    Real Lazy Field

>>> mypi = RLF(pi)
    mypi

    3.141592653589794?

>>> mypi.n(200)

    3.1415926535897932384626433832795028841971693993751058209749

>>> CLF

    Complex Lazy Field

>>>
```