به نام خدا

حد

مشتق

انتگرال

سری

ارائه دهنده : معصومه جوکار

1-limits

return the limit as the variable *v* approaches *a* from the given direction.

•dir - (default: None); dir may have the value ‘plus’ (or ‘+’ or ‘right’) for a limit from above, ‘minus’ (or ‘-‘ or ‘left’) for a limit from below, or may be omitted (implying a two-sided limit is to be computed).

•taylor - (default: False); if True, use Taylor series, which allows more limits to be computed (but may also crash in some obscure cases due to bugs in Maxima).

- Note

The output may also use ‘und’ (undefined), ‘ind’ (indefinite but bounded), and ‘infinity’ (complex infinity).

Limits

$\lim_{x \to a} f(x) =$ `limit(f(x), x=a)`

```
limit(sin(x)/x, x=0)
```

$\lim_{x \to a^+} f(x) =$ `limit(f(x), x=a, dir='plus')`

```
limit(1/x, x=0, dir='plus')
```

$\lim_{x \to a^-} f(x) =$ `limit(f(x), x=a, dir='minus')`

```
limit(1/x, x=0, dir='minus')
```

exampeles:

```
sage: x = var('x')
sage: f = (1+1/x)^x
sage: f.limit(x = oo)
e
sage: f.limit(x = 5)
7776/3125
sage: f.limit(x = 1.2)
2.06961575467...
sage: f.limit(x = I, taylor=True)
(-I + 1)^I
```

More examples:

```
sage: limit(x*log(x), x = 0, dir='+')
0
sage: lim((x+1)^(1/x), x = 0)
e
sage: lim(e^x/x, x = oo)
+Infinity
sage: lim(e^x/x, x = -oo)
0
sage: lim(-e^x/x, x = oo)
-Infinity
sage: lim((cos(x))/(x^2), x = 0)
+Infinity
```

Here ind means “indefinite but bounded”:

```
sage: lim(sin(1/x), x = 0)
ind
```

```
sage: limit(floor(x), x=0, dir='-')
-1
sage: limit(floor(x), x=0, dir='+')
0
sage: limit(floor(x), x=0)
und
```

```
sage: limit(1/x, x=0)
Infinity
sage: limit(1/x, x=0, dir='+')
+Infinity
sage: limit(1/x, x=0, dir='-')
-Infinity
```

- These calculation for the exponential function shows a pretty serious annoyance in *Sage:* It needs to know whether is an
- integer, even though the result of the calculation doesn't depend on the answer to this questions

  We therefore end up having to
- use *Sage's* **assume()** function to give both possible answers for the type of .

```
var('h')
limit((sin(x+h) - sin(x))/h, h=0)
```

$$\cos(x)$$

```
limit((exp(x+h) - exp(x))/h, h=0)
```

```
Traceback (click to the left of this block for traceback)
...
Is x an integer?
```

```
assume(x, 'integer')
limit((exp(x+h) - exp(x))/h, h=0)
```

$$e^x$$

```
forget()
assume(x, 'noninteger')
limit((exp(x+h) - exp(x))/h, h=0)
```

$$e^x$$

```
forget()
```

```
var('a')
limit((x^(1/3) - a^(1/3))/(x-a), x=a)
```

$$\frac{1}{3}\,\frac{1}{a^{\left(\frac{2}{3}\right)}}$$

Here's one last limit, just for fun. You might think about how to do it by hand.

```
limit(x^x,x=0)
```

$$1$$

2-derivative

- *Sage* can also compute derivatives, either of expressions or of functions. **derivative()**, **differentiate()**, and **diff()** are three different names for the same function (or method) for doing this.

```
nestrad = sqrt(x+sqrt(x+sqrt(x)))
nestrad
```

$$\sqrt{x+\sqrt{x+\sqrt{x}}}$$

```
diff(nestrad,x)
```

$$\frac{1}{8}\,\frac{\left(\frac{\left(\frac{1}{\sqrt{x}}+2\right)}{\sqrt{x+\sqrt{x}}}+4\right)}{\sqrt{x+\sqrt{x+\sqrt{x}}}}$$

```
_.simplify_rational()
```

$$\frac{1}{8}\,\frac{\left(4\,\sqrt{x+\sqrt{x}}\sqrt{x}+2\,\sqrt{x}+1\right)}{\sqrt{x+\sqrt{x}}\sqrt{x+\sqrt{x+\sqrt{x}}}\sqrt{x}}$$

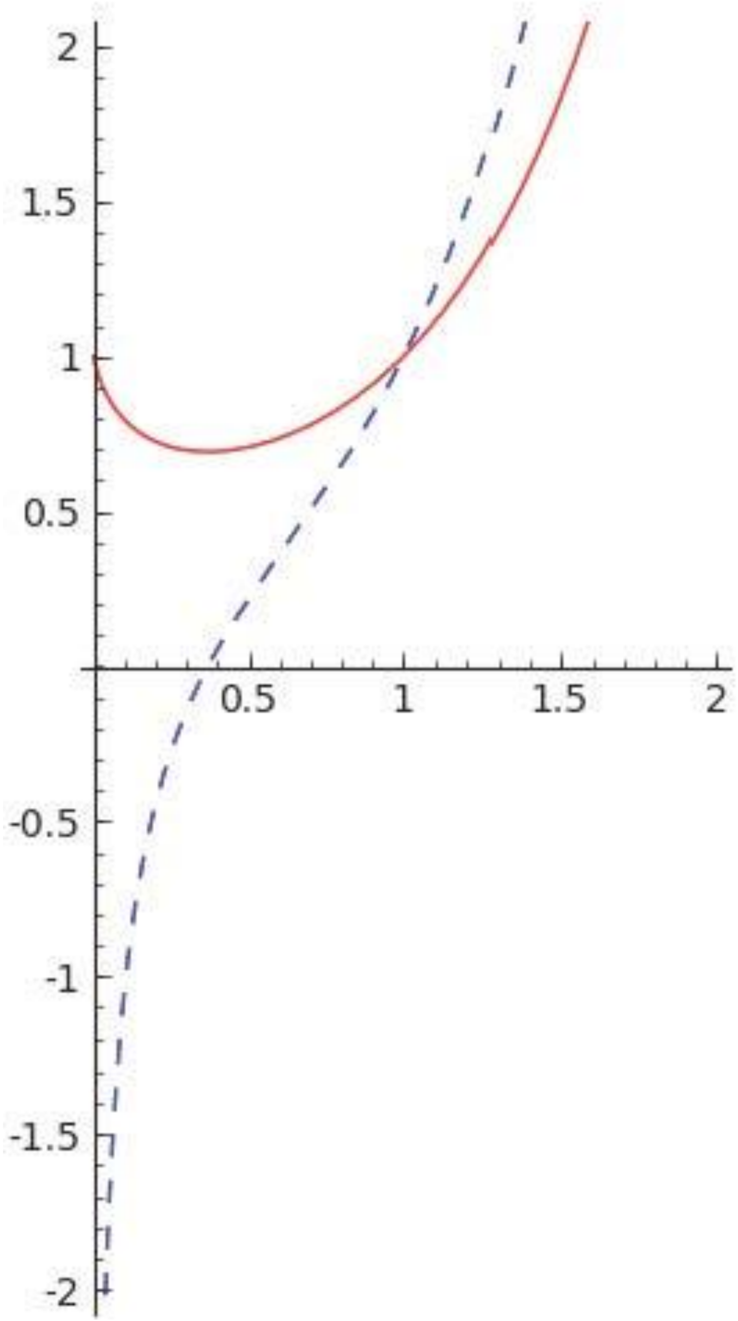
```
f(x)=x^x
f
```

$$x \mapsto x^x$$

```
fprime = diff(f)
fprime
```

$$x \mapsto (\log(x) + 1)x^x$$

```
pf = plot(f, (x,0,2), color='red', linestyle='-')
pfprime = plot(fprime, (x,0,2), color='blue', linestyle='--')
show(pf+pfprime,aspect_ratio=1, ymin=-2, ymax=2)
```



```
diff(sin(x),x)
```

$$\cos(x)$$

```
diff(sin(x),x,4)
```

$$\sin(x)$$

```
sin(x).diff(x,21)
```

$$\cos(x)$$

```
f(x)=sin(x)
diff(f)
```

$$x \mapsto \cos(x)$$

```
diff(f,x,4)
diff(f,x,4)(pi/4)
```

$$x \mapsto \sin(x)$$

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

Sage can even handle differentiation rules for arbitrary functions, like this:

```
function('f g h')
var('x')
```

$$x$$

```
diff(f(x)*g(x),x)
```

$$f\left(x\right) D[0]\left(g\right)\left(x\right) + g\left(x\right) D[0]\left(f\right)\left(x\right)$$

```
diff(f(x)/g(x), x)
```

$$-\frac{f(x)D[0](g)(x)}{g(x)^2} + \frac{D[0](f)(x)}{g(x)}$$

```
diff(f(x)*g(x)*h(x),x)
```

$$f\left(x\right) g\left(x\right) D[0]\left(h\right)\left(x\right) + f\left(x\right) h\left(x\right) D[0]\left(g\right)\left(x\right) + g\left(x\right) h\left(x\right) D[0]\left(f\right)\left(x\right)$$

```
diff(f(g(x)),x)
```

$$D[0]\left(f\right)\left(g\left(x\right)\right) D[0]\left(g\right)\left(x\right)$$

diff(sqrt(f(x)),x)

$$\frac{1}{2}\,\frac{D[0](f)(x)}{\sqrt{f(x)}}$$

EXAMPLES: We differentiate a callable symbolic function:

```
sage: f(x,y) = x*y + sin(x^2) + e^(-x)
sage: f
(x, y) |--> x*y + e^(-x) + sin(x^2)
sage: derivative(f, x)
(x, y) |--> 2*x*cos(x^2) + y - e^(-x)
sage: derivative(f, y)
(x, y) |--> x
```

We differentiate a polynomial:

```
sage: t = polygen(QQ, 't')
sage: f = (1-t)^5; f
-t^5 + 5*t^4 - 10*t^3 + 10*t^2 - 5*t + 1
sage: derivative(f)
-5*t^4 + 20*t^3 - 30*t^2 + 20*t - 5
sage: derivative(f, t)
-5*t^4 + 20*t^3 - 30*t^2 + 20*t - 5
sage: derivative(f, t, t)
-20*t^3 + 60*t^2 - 60*t + 20
sage: derivative(f, t, 2)
-20*t^3 + 60*t^2 - 60*t + 20
sage: derivative(f, 2)
-20*t^3 + 60*t^2 - 60*t + 20
```

We differentiate a symbolic expression:

```
sage: var('a x')
(a, x)
sage: f = exp(sin(a - x^2))/x
sage: derivative(f, x)
-2*cos(-x^2 + a)*e^(sin(-x^2 + a)) - e^(sin(-x^2 + a))/x^2
sage: derivative(f, a)
cos(-x^2 + a)*e^(sin(-x^2 + a))/x
```

Syntax for repeated differentiation:

```
sage: R.<u, v> = PolynomialRing(QQ)
sage: f = u^4*v^5
sage: derivative(f, u)
4*u^3*v^5
sage: f.derivative(u)    # can always use method notation
too
4*u^3*v^5
```

```
sage: derivative(f, u, u)
12*u^2*v^5
sage: derivative(f, u, u, u)
24*u*v^5
sage: derivative(f, u, 3)
24*u*v^5
```

```
sage: derivative(f, u, v)
20*u^3*v^4
sage: derivative(f, u, 2, v)
60*u^2*v^4
sage: derivative(f, u, v, 2)
80*u^3*v^3
sage: derivative(f, [u, v, v])
80*u^3*v^3
```

Partial derivative notation in Sage

It’s confusing at first, but a simple example will make it clear.

```
var 'x y'
function 'test', x, y
show test x, y
show diff test x, y , x
show diff test x, y , x, 2
show diff test x, y , y
show diff test x, y , y, 2
```

I defined a function of two variables called **test**. When you run this in the notebook interface, you get:

```
test(x,y)
D[0](test)(x,y)
D[0,0](test)(x,y)
D[1](test)(x,y)
D[1,1](test)(x,y)
```

The capital D denotes a derivative. The numbers in brackets indicate which variable the derivative is with respect to. In this example, 0 denotes x and 1 denotes y.
The number of times a number is repeated indicates the order of the derivative.
D[0] is the first derivative with respect to x, **D[1]** is the first derivative with respect to y, **D[0,0]** is the second derivative with respect to x and **D[1,1]** is the second derivative with respect to y.

example

```
y=var('y')
f(x,y)=3*x^3*y+x^2-4*y^3
```

show(f)

$$(x,y) \mapsto 3x^3y + x^2 - 4y^3$$

- f.diff(x) is the partial with respect to x and f.diff(y) is the partial with respect to y.

```
f.diff(x)
```

```
(x, y) |--> 9*x^2*y + 2*x
```

```
f.diff(y)
```

```
(x, y) |--> 3*x^3 - 12*y^2
```

- If you want to take higher partials, just add in more variables. Thus f.diff(x,y) is the same as fxy.

```
f.diff(x,y)
```

```
(x, y) |--> 9*x^2
```

Implicit differentiation displays extraneous x variable.

The problem is, given y = 9*x^(1/2) - 2*y^(3/5), find dy/dx The answer is supposed to be dy/dx = (45*y^(2/5)) / (10*x^(1/2)*y^(2/5)+12*x^(1/2))

When I enter the following syntax, there is an extraneous character, (x), displayed:

```
y=function('y',x)
temp=diff(9*x^(1/2) - 2*y^(3/5) - y)
solve (temp,diff(y))
show(solve (temp,diff(y)))
```

Indefinite Integrals

We can use Sage to evaluate many indefinite integrals. Recall that if ***f* is a function, then**

$$\int f(x)\,dx = F(x) + C,$$

where F is an antiderivative of f (assuming it has one). The syntax for integrating a function in Sage is as follows:

```
integral("function", "variable of integration")
```

where "function" is the function that you want to integrate (without the quotes) and "variable of integration" is the variable that you want to integrate with respect to (typically x).

For example, if you want to find the indefinite integral of $y=\sin(x)$, then you would type:

```
integral(sin(x),x)
```

- Try typing the expression above in an empty Sage cell and then clicking "evaluate" (or typing "shift+enter")

- One thing that you should notice is that the answer is missing the constant of integration (i.e. "+C"). You'll just have to remember that this is always missing.

- We can also integrate functions that we have previously defined.

- In an empty Sage cell, define the function $f(x)=3x2-2x3$ by typing the following.

```
f(x)=3*x^2-2*x^3
```

- Next, enter the following into an empty Sage cell and evaluate it.

```
integral(f(x),x)
```

Enter the following into a Sage cell.

```
f_int(x)=integral(f(x),x)
```

It will look like nothing happened, but all we did was tell Sage to let `f_int` denote the indefinite integral of f (I made up the notation `f_int`).

To see what `f_int` actually is, type the following into a Sage cell:

```
show(f_int(x))
```

To differentiate `f_int`, type the following into a Sage cell.

```
diff(f_int(x),x)
```

If you did everything correctly, the answer that you got should be equal to the original function f.

Definite Integrals

Thankfully, we can compute the exact values of definite integrals using Sage. The syntax for evaluating a definite integral using Sage is similar to the syntax for indefinite integrals, except that we need to specify the interval:

```
integral("function", "variable of integration", a,b)
```

EXAMPLES:

```
sage: x = var('x')
sage: h = sin(x)/(cos(x))^2
sage: h.integral(x)
1/cos(x)
```

```
sage: f = x^2/(x+1)^3
sage: f.integral(x)
1/2*(4*x + 3)/(x^2 + 2*x + 1) + log(x + 1)
```

```
sage: f = x*cos(x^2)
sage: f.integral(x, 0, sqrt(pi))
0
sage: f.integral(x, a=-pi, b=pi)
0
```

```
sage: f(x) = sin(x)
sage: f.integral(x, 0, pi/2)
1
```

Constraints are sometimes needed:

```
sage: var('x, n')
(x, n)
sage: integral(x^n,x)
Traceback (most recent call last):
...
ValueError: Computation failed since Maxima requested
additional
constraints; using the 'assume' command before evaluation
*may* help (example of legal syntax is 'assume(n>0)', see
`assume?`
for more details)
Is n equal to -1?
sage: assume(n > 0)
sage: integral(x^n,x)
x^(n + 1)/(n + 1)
sage: forget()
```

Note that an exception is raised when a definite integral is divergent:

```
sage: forget() # always remember to forget assumptions you no longer need
sage: integrate(1/x^3,(x,0,1))
```

```
Traceback (most recent call last):
...
ValueError: Integral is divergent.
sage: integrate(1/x^3,x,-1,3)
Traceback (most recent call last):
...
ValueError: Integral is divergent.
```

The examples in the Maxima documentation:

```
sage: var('x, y, z, b')
(x, y, z, b)
sage: integral(sin(x)^3, x)
1/3*cos(x)^3 - cos(x)
sage: integral(x/sqrt(b^2-x^2), b)
x*log(2*b + 2*sqrt(b^2 - x^2))
sage: integral(x/sqrt(b^2-x^2), x)
-sqrt(b^2 - x^2)
sage: integral(cos(x)^2 * exp(x), x, 0, pi)
3/5*e^pi - 3/5
sage: integral(x^2 * exp(-x^2), x, -oo, oo)
1/2*sqrt(pi)
```

Multiple integrals in Sage

```
##In the first two cells we do a double integral example

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

```
-2/3*y^(3/2) + 2/3*e^(3/2*y)
```

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

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

```
##Here's a triple integral example
x=var('x');
y=var('y');
z=var('z');
innerint=integral(1,z,0,(1/2)*(6-6*x-3*y));
middleint=integral(innerint,y,0,-2*x+2);
outerint=integral(middleint,x,0,1);
outerint
```

```
1
```

Double integral

```
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
```

Triple integrals

```
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)
```

$\pi \sin(1)$

To integrate the function *x2* from 0 to 1, we do

```
sage: numerical_integral(x^2, 0, 1, max_points=100)
(0.3333333333333333, 3.700743415417188e-15)
```

To integrate the function $\sin(x)^3 + \sin(x)$ we do

```
sage: numerical_integral(sin(x)^3 + sin(x),  0, pi)
(3.333333333333333, 3.700743415417188e-14)
```

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

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

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

```
(0.8687400395769761, 9.644951943521286e-15)
```

Series and summations

An infinite series is a summation of a sequence with an infinite number of terms. Truncated series are useful for approximating functions. In this section, we'll look at the capabilities that Sage has for computing infinite sequences and computing their sums.

```
var('x, n, k')

f(x) = sin(x) / x^2
f.show()

print("Power series expansion around x=1:")
s(x) = f.series(x==1, 3)
s.show()

print("Sum of alternating harmonic series:")
h(k) = (-1)^(k + 1) * 1 / k
print h.sum(k, 1, infinity)

print("Sum of binomial series:")
h(k) = binomial(n, k)
print h.sum(k, 1, infinity)

print("Sum of harmonic series:")
h(k) = 1 / k
print h.sum(k, 1, infinity)    # Diverges
```

The results are shown in the following screenshot:

$$x \mapsto \frac{\sin(x)}{x^2}$$

```
Power series expansion around x=1:
```

$$x \mapsto (\sin(1)) + (-2\sin(1) + \cos(1))(x-1) + (\frac{5}{2}\sin(1) - 2\cos(1))(x-1)^2 + O\left((x-1)^3\right)$$

```
Alternating harmonic series:
k |--> log(2)
Binomial series:
k |--> 2^n - 1
Harmonic series:
Traceback (click to the left of this block for traceback)
...
ValueError: Sum is divergent.
```

What just happened?

We started by defining a function and using the `series` method to compute a power series around the point x=1. The first argument to `series` is the point at which to create the series, and the second argument is the order of the computed series. Notice that Sages uses "big O" notation to denote the order of the series.

We then created several infinite series and computed their sums. Sage can compute the sum of any convergent series using the `sum` method. The first argument to `sum` is the summation variable, the second argument is the lower endpoint, and the third argument is the upper endpoint of the series. We used this method to compute the sum of the alternating harmonic series and the binomial series. However, we ran into trouble when we tried to compute the sum of the harmonic series, which is divergent. Fortunately, Sage handled this problem gracefully.

ما از جهانیان، هیچ نمی خواهیم، جز احترام. وینستون چرچیل

یادت باشد اگر چشمت به این دست نوشته افتاد
به خاطر بیاوری که آن‌هایی که هر روز می‌بینی و مراوده می‌کنی
همه انسان هستند و دارای خصوصیات یک انسان، با نقابی متفاوت،
اما همگی جایزالخطا.
«مهاتما گاندی»