

Handouts/Lecture 4 (Calculus and graphics)/Lecture 4: Calculus ("matan")/Limits.sagews

Author Eugene Strakhov

Date 2019-06-29T19:06:10

Project 07c06dbe-4967-451f-aa68-dd9268bd2ece

Location [Handouts/Lecture 4 \(Calculus and graphics\)/Lecture 4: Calculus \("matan"\)/Limits.sagews](#)

Original file [Limits.sagews](#)

Пределы

Для вычисления пределов в символьном (аналитическом) виде SageMath обращается к соответствующим функциям системы Maxima.

```
limit(expr, x=...)
limit(expr, x=..., dir=...)
limit(expr, x=..., taylor=True)
expr.limit(x=..., dir=..., taylor=True)
```

Для вычисления пределов в символьном (аналитическом) виде SageMath обращается к соответствующим функциям системы Maxima.

```
limit(expr, x=...)
limit(expr, x=..., dir=...)
limit(expr, x=..., taylor=True)
expr.limit(x=..., dir=..., taylor=True)
```

```
1 n = var('n', domain=ZZ)
2 # limit() как функция
3 l1 = limit((n^2+n+1)/(2*n^2-5), n=Infinity)
4 print l1
5 # limit() как метод
6 l2 = ((1+1/n)^n).limit(n=oo) # вместо 'Infinity' можно использовать 'oo' (две подряд буквы 'o')
7 print l2
```

1/2

e

Функции `show()` и `latex()` позволяют выводить ответы в красивом виде.

Функции `show()` и `latex()` позволяют выводить ответы в красивом виде.

```
8 # Красивый вывод
9 expr = (n^2+n+1)/(2*n^2-5)
10 show(r'\displaystyle{\lim\limits_{n\to\infty}'+latex(expr)+'='+latex(expr.limit(n=oo))+'}$')
```

$$\lim_{n \rightarrow \infty} \frac{n^2 + n + 1}{2n^2 - 5} = \frac{1}{2}$$

Кроме «регулярных» ответов, вы можете получить такие:

- `und` (предел не найден);
- `ind` (не определён, но ограничен);
- `Infinity`, `-Infinity` означают бесконечные пределы.

Кроме «регулярных» ответов, вы можете получить такие:

- `und` (предел не найден);
- `ind` (не определён, но ограничен);
- `Infinity`, `-Infinity` означают бесконечные пределы.

```
11 # lim() - короткий вариант ФУНКЦИИ (но не метода!) limit()
12 lim(sqrt(n+1)-sqrt(n), n=oo)
13 lim((-1)^n, n=oo) # предел не существует, однако последовательность ограничена
14 lim(-n^2, n=oo) # бесконечный предел

0
ind
```

-Infinity

Кроме собственно переменной, по которой осуществляется предельный переход, выражение может содержать и другие переменные. Вычислим

$$\lim_{x \rightarrow 1} \frac{\sqrt[m]{x} - 1}{\sqrt[k]{x} - 1}$$

Кроме собственно переменной, по которой осуществляется предельный переход, выражение может содержать и другие переменные. Вычислим

$$\lim_{x \rightarrow 1} \frac{\sqrt[m]{x} - 1}{\sqrt[k]{x} - 1}$$

```
15 # Предел с параметрами
16 x, m, k = var('x m k')
17 assume(m, 'integer', k, 'integer')
18 expr = (x^(1/m)-1)/(x^(1/k)-1)
19 expr.limit(x=1) # ответ получен в символьном виде
```

k/m

Для вычисления некоторых пределов могут понадобиться дополнительные предположения (`assume`). Попробуем найти $\lim_{x \rightarrow 0} x^a$, ничего не сказав предварительно об a :

Для вычисления некоторых пределов могут понадобиться дополнительные предположения (`assume`). Попробуем найти $\lim_{x \rightarrow 0} x^a$, ничего не сказав предварительно об a :

```
20 x, a = var('x a')
21 (x^a).limit(x=0)
```

```
Error in lines 2-2
Traceback (most recent call last):
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/smc_sagews/sage_server.py", line 982, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in 
  File "sage/symbolic/expression.pyx", line 11709, in sage.symbolic.expression.Expression.limit (/projects/sage/sage-7.5/src/build/cythonized/sage/symbolic/expression.cpp:64904)
    return limit(self, *args, **kwargs)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/calculus/calculus.py", line 1248, in limit
    l = maxima.sr_limit(ex, v, a)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 972, in sr_limit
    self._missing_assumption(s)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 1017, in _missing_assumption
    raise ValueError(outstr)
ValueError: Computation failed since Maxima requested additional constraints; using the 'assume' command before evaluation *may* help
(example of legal syntax is 'assume(a>0)', see  $as \sum e?$  for more details)
Is a positive, negative or zero?
```

Заметьте, что мы видим на экране **не сообщение об ошибке**. То есть формально это так, а в данном случае система просто «пытается узнать» от пользователя дополнительные сведения.

Итак, является a положительным, отрицательным или нулём?

Заметьте, что мы видим на экране **не сообщение об ошибке**. То есть формально это так, но в данном случае система просто «пытается узнать» от пользователя дополнительные сведения.

Итак, является a положительным, отрицательным или нулём?

```
22 assume(a>0)
23 (x^a).limit(x=0)

Error in lines 2-2
Traceback (most recent call last):
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/smc_sagews/sage_server.py", line 982, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in 
  File "sage/symbolic/expression.pyx", line 11709, in sage.symbolic.expression.Expression.limit (/projects/sage/sage-7.5/src/build/cythonized/sage/symbolic/expression.cpp:64904)
    return limit(self, *args, **kwargs)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/calculus/calculus.py", line 1248, in limit
    l = maxima.sr_limit(ex, v, a)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 972, in sr_limit
    self._missing_assumption(s)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 1017, in _missing_assumption
    raise ValueError(outstr)
ValueError: Computation failed since Maxima requested additional constraints; using the 'assume' command before evaluation *may* help
(example of legal syntax is 'assume(a>0)', see  $as \sum e?$  for more details)
Is a an integer?
```

Теперь надо ответить, является ли a целым числом.

Теперь надо ответить, является ли a целым числом.

```

24 assume(a, 'integer')
25 (x^a).limit(x=0)

Error in lines 2-2
Traceback (most recent call last):
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/smc_sagews/sage_server.py", line 982, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in 
  File "sage/symbolic/expression.pyx", line 11709, in sage.symbolic.expression.Expression.limit (/projects/sage/sage-
7.5/src/build/cythonized/sage/symbolic/expression.cpp:64904)
    return limit(self, *args, **kwargs)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/calculus/calculus.py", line 1248, in limit
    l = maxima.sr_limit(ex, v, a)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 972, in sr_limit
    self._missing_assumption(s)
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/interfaces/maxima_lib.py", line 1017, in _missing_assumption
    raise ValueError(outstr)
ValueError: Computation failed since Maxima requested additional constraints; using the 'assume' command before evaluation *may* help
(example of legal syntax is 'assume(a>0)', see  $\sum e?$  for more details)
Is a an even number?

```

Наконец, является ли a чётным?

Наконец, является ли a чётным?

```

26 assume(a, 'odd') # нечётное
27 (x^a).limit(x=0)

```

0

Предел вычислен. Теперь надо не забыть убрать все предположения.

Предел вычислен. Теперь надо не забыть убрать все предположения.

```

28 forget()

```

Для вычисления односторонних пределов используется параметр `dir`.

- `dir='+'` или `dir='right'` — предел справа;
- `dir='-'` или `dir='left'` — предел слева.

Для вычисления односторонних пределов используется параметр `dir`.

- `dir='+'` или `dir='right'` — предел справа;
- `dir='-'` или `dir='left'` — предел слева.

```

29 # Односторонние пределы
30 lim(1/x, x=0, dir='+')
31 lim(1/x, x=0, dir='-')

```

+Infinity

-Infinity

Замечание. К Python-функциям нахождение пределов неприменимо.

Замечание. К Python-функциям нахождение пределов неприменимо.

```

32 def sgn(x) :
33     if x > 0 :
34         return 1
35     elif x < 0 :
36         return -1
37     else :
38         return 0
39 lim(sgn, x=1) # ошибка

```

Error in lines 8-8

Traceback (most recent call last):

```

  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/smc_sagews/sage_server.py", line 982, in execute
    exec compile(block+'\n', '', 'single') in namespace, locals
  File "", line 1, in 
  File "/projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/calculus/calculus.py", line 1230, in limit
    ex = SR(ex)
  File "sage/structure/parent.pyx", line 955, in sage.structure.parent.Parent.__call__ (/projects/sage/sage-
7.5/src/build/cythonized/sage/structure/parent.c:9852)
    return mor._call_(x)
  File "sage/structure/coerce_maps.pyx", line 110, in sage.structure.coerce_maps.DefaultConvertMap_unique._call__ (/projects/sage/sage-
7.5/src/build/cythonized/sage/structure/coerce_maps.c:4883)
    raise
  File "sage/structure/coerce_maps.pyx", line 105, in sage.structure.coerce_maps.DefaultConvertMap_unique._call__ (/projects/sage/sage-
7.5/src/build/cythonized/sage/structure/coerce_maps.c:4744)

```

```

    return C._element_constructor(x)
File "sage/symbolic/ring.pyx", line 382, in sage.symbolic.ring.SymbolicRing._element_constructor_ (/projects/sage/sage-7.5/src/build/cythonized/sage/symbolic/ring.cpp:7045)
    raise TypeError(f"unable to convert {x!r} to a symbolic expression")
TypeError: unable to convert  to a symbolic expression
*** WARNING: Code contains non-ascii characters ***

```

40 limit?

File: /projects/sage/sage-7.5/local/lib/python2.7/site-packages/sage/calculus/calculus.py
Signature : limit(ex, dir=None, taylor=False, algorithm='maxima', **kwds)
Docstring :
Return the limit as the variable v approaches a from the given direction.

```

    expr.limit(x = a)
    expr.limit(x = a, dir='+')

```

INPUT:

```

* "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).

* "**argv" - 1 named parameter

```

Note: The output may also use 'und' (undefined), 'ind' (indefinite but bounded), and 'infinity' (complex infinity).

EXAMPLES:

```

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
sage: f(x=1.2)
2.0696157546720...
sage: f(x=I)
(-I + 1)^I
sage: CDF(f(x=I))
2.0628722350809046 + 0.7450070621797239*I
sage: CDF(f.limit(x = I))
2.0628722350809046 + 0.7450070621797239*I

```

Notice that Maxima may ask for more information:

```

sage: var('a')
a
sage: limit(x^a,x=0)
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(a>0)', see `assume?` for more details)
Is a positive, negative or zero?

```

With this example, Maxima is looking for a LOT of information:

```

sage: assume(a>0)
sage: limit(x^a,x=0)
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(a>0)', see `assume?` for more details)
Is a an integer?
sage: assume(a,'integer')
sage: limit(x^a,x=0)
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(a>0)', see `assume?` for more details)
Is a an even number?
sage: assume(a,'even')
sage: limit(x^a,x=0)
0
sage: forget()

```

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
sage: lim(sqrt(x^2+1) - x, x = oo)
0
sage: lim(x^2/(sec(x)-1), x=0)
2
sage: lim(cos(x)/(cos(x)-1), x=0)
-Infinity
sage: lim(x*sin(1/x), x=0)
0
sage: limit(e^(-1/x), x=0, dir='right')
0
sage: limit(e^(-1/x), x=0, dir='left')
+Infinity

sage: f = log(log(x))/log(x)
sage: forget(); assume(x<-2); lim(f, x=0, taylor=True)
0
sage: forget()

```

Here ind means "indefinite but bounded":

```

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

```