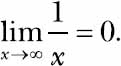
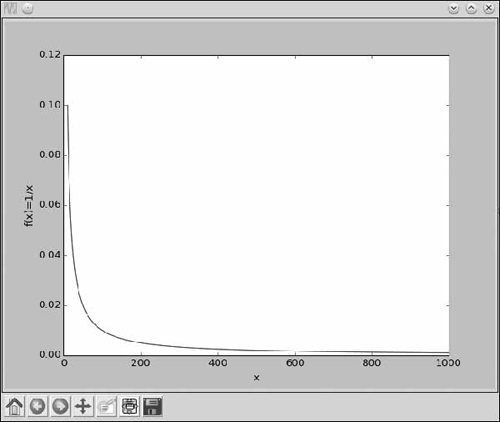
Capítulo 7: Resolución de problemas de cálculo

### **Encontrar el límite de funciones**

Una tarea habitual en cálculo es encontrar el *valor límite* (o simplemente el *límite*) de la función, cuando se supone que el valor de la variable se aproxima a un determinado valor. Considera una función *f(x*) = *1/x*, cuya gráfica se muestra en [la Figura 7-2](ch07.html#ch7fig2).

A medida que aumenta el valor de *x*, el valor de *f*(*x*) se aproxima a 0. Utilizando la notación límite, escribiríamos esto como





*Figura 7-2: Gráfica de la función 1/x* *a* medida que *aumenta* el valor de x

Podemos encontrar límites de funciones en SymPy creando objetos de la clase Limit como se indica a continuación:

➊ >>> from sympy import Limit, Symbol, S  
➋ >>> x = Symbol('x')  
➌ >>> Limit(1/x, x, S.Infinity)  
Limit(1/x, x, oo, dir='-')

En ➊, importamos las clases Limit y Symbol, así como S, que es una clase especial de SymPy que contiene la definición de infinito (positivo y negativo) y otros valores especiales. A continuación, en ➋ creamos un objeto símbolo, x, para representar *x*. Creamos el objeto Limit en ➌, pasándole tres argumentos: 1/x, la variable x, y por último el valor en el que queremos calcular el límite de la función (infinito, dado por S.Infinity).

El resultado se devuelve como un objeto *no evaluado* con el símbolo oo que denota el infinito positivo y el símbolo dir='-' que especifica que nos acercamos al límite por el lado negativo.

Para encontrar el valor del límite, utilizamos el método doit():

>>> l = Limit(1/x, x, S.Infinity)  
>>> l.doit()  
0

Por defecto, el límite se encuentra desde una dirección positiva, a menos que el valor en el que se va a calcular el límite sea infinito positivo o negativo. En el caso del infinito positivo, la dirección es negativa, y viceversa. Puedes cambiar la dirección por defecto como se indica a continuación:

>>> Limit(1/x, x, 0, dir='-').doit()  
-oo

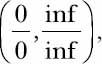
Aquí calculamos

image

y a medida que nos acercamos a 0 para *x* desde el lado negativo, el valor del límite se aproxima al infinito negativo. En cambio, si nos acercamos a 0 desde el lado positivo, el valor se aproxima al infinito positivo:

>>> Limit(1/x, x, 0, dir='+').doit()  
oo

La clase Limit también maneja funciones con límites de formas indeterminadas,



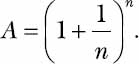
automáticamente:

>>> from sympy import Symbol, sin  
>>> Limit(sin(x)/x, x, 0).doit()  
1

Es muy probable que hayas utilizado la regla de l'Hôpital para encontrar dichos límites, pero como vemos aquí, la clase Limit se encarga de ello por nosotros.

#### ***Interés compuesto continuo***

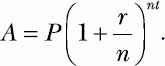
Supongamos que has depositado 1 $ en un banco. Este depósito es el *capital*, que te paga un interés *-en*este caso, un interés del 100% que se compone n veces al año durante 1 año. La cantidad que obtendrás al cabo de 1 año viene dada por



El destacado matemático James Bernoulli descubrió que, a medida que aumenta el valor de *n*, el término (1 + *1/n)n* se aproxima al valor de *e, la*constante que podemos verificar hallando el límite de la función:

>>> from sympy import Limit, Symbol, S  
>>> n = Symbol('n')  
>>> Limit((1+1/n)\*\*n, n, S.Infinity).doit()  
E

Para cualquier importe principal *p*, cualquier tipo *r* y cualquier número de años *t*, el interés compuesto se calcula mediante la fórmula



Suponiendo un interés compuesto continuo, podemos hallar la expresión de *A* de la siguiente manera:

>>> from sympy import Symbol, Limit, S  
>>> p = Symbol('p', positive=True)  
>>> r = Symbol('r', positive=True)  
>>> t = Symbol('t', positive=True)  
>>> Limit(p\*(1+r/n)\*\*(n\*t), n, S.Infinity).doit()  
p\*exp(r\*t)

Creamos tres objetos símbolo, que representan el importe principal, p, el tipo de interés, r, y el número de años, t. También le decimos a SymPy que estos símbolos asumirán valores positivos pasando el argumento de la palabra clave positive=True al crear los objetos Symbol. Si no lo especificamos, SymPy no sabe nada sobre los valores numéricos que puede asumir el símbolo y puede que no sea capaz de evaluar el límite correctamente. A continuación, introducimos la expresión del interés compuesto para crear el objeto Limit y lo evaluamos mediante el método doit(). El límite resulta ser p\*exp(r\*t), lo que nos indica que el interés compuesto crece exponencialmente con el tiempo para el tipo de interés fijo.

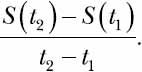
#### ***Tasa de variación instantánea***

Considera un coche que se desplaza por una carretera. Acelera uniformemente de forma que la distancia recorrida, *S*, viene dada por la función

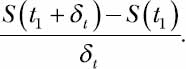
*S(t*) = 5t2 + *2t* + 8.

En esta función, la variable independiente es *t*, que representa el tiempo transcurrido desde que el coche empezó a moverse.

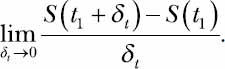
Si medimos la distancia recorrida en el tiempo t1 y en el tiempo t2 de forma que t2 > t1, podemos calcular la distancia recorrida por el coche en 1 unidad de tiempo mediante la expresión



También se denomina tasa de variación media de la función *S(t*) respecto a la variable *t*, o dicho de otro modo, velocidad media. Si escribimos t2 como t1 + δt *-dondeδt* es la diferencia entre t2 y t1 en unidades de tiempo- podemos reescribir la expresión de la velocidad media como



Esta expresión también es una función con t1 como variable. Ahora bien, si además suponemos que *δt* es realmente pequeña, de modo que se aproxima a 0, podemos utilizar la notación de límite para escribirla como



Ahora evaluaremos el límite anterior. En primer lugar, vamos a crear los distintos objetos de expresión:

>>> from sympy import Symbol, Limit  
>>> t = Symbol('t')  
➊ >>> St = 5\*t\*\*2 + 2\*t + 8  
  
>>> t1 = Symbol('t1')  
>>> delta\_t = Symbol('delta\_t')  
  
➋ >>> St1 = St.subs({t: t1})  
➌ >>> St1\_delta = St.subs({t: t1 + delta\_t})

Primero definimos la función *S(t*) en ➊. A continuación, definimos dos símbolos, t1 y delta\_t, que corresponden a t1 y *δt*. Utilizando el método subs(), hallamos entonces *S*(t1) y *S(*t1 + *δt*) sustituyendo el valor de t por t1 y t1\_delta\_t en ➋ y ➌, respectivamente.

Ahora, evaluemos el límite:

>>> Limit((St1\_delta-St1)/delta\_t, delta\_t, 0).doit()  
10\*t1 + 2

El límite resulta ser 10\*t1 + 2, y es la tasa de cambio de *S(t*) en el tiempo t1, o la tasa de cambio instantánea. Este cambio se denomina más comúnmente *velocidad instant* ánea del coche en el instante de tiempo t1.

El límite que hemos calculado aquí se denomina *derivada* de una función, y podemos calcularlo directamente utilizando la clase Derivative de SymPy.

[anterior](ch07_3.html)[Subtema 4 de 11: (Ver todo)](ch07.html)[siguiente](ch07_5.html)