Capítulo 7: Resolución de problemas de cálculo

### **Encontrar la derivada de funciones**

La derivada de una función *y* = *f(x*) expresa la tasa de cambio de la variable dependiente, *y*, con respecto a la variable independiente, *x*. Se denota como *f′(x*) o *dy/dx*. Podemos hallar la derivada de una función creando un objeto de la clase Derivative. Utilicemos como ejemplo la función anterior que representa el movimiento de un coche:

➊ >>> from sympy import Symbol, Derivative  
  
>>> t = Symbol('t')  
>>> St = 5\*t\*\*2 + 2\*t + 8  
  
➋ >>> Derivative(St, t)  
Derivative(5\*t\*\*2 + 2\*t + 8, t)

Importamos la clase Derivative en ➊. En ➋, creamos un objeto de la clase Derivative. Los dos argumentos que se pasan al crear el objeto son la función St y el símbolo t, que corresponde a la variable *t*. Al igual que con la clase Limit, se devuelve un objeto de la clase Derivative, y en realidad no se calcula la derivada. Llamamos al método doit() sobre el objeto no evaluado Derivative para hallar la derivada:

>>> d = Derivative(St, t)  
>>> d.doit()  
10\*t + 2

La expresión para la derivada resulta ser 10\*t + 2. Ahora, si queremos calcular el valor de la derivada en un valor concreto de *t -digamos*, *t* = t1 o *t* = 1- podemos utilizar el método subs():

>>> d.doit().subs({t:t1})  
10\*t1 + 2  
>>> d.doit().subs({t:1})  
12

Probemos con una función arbitraria complicada con *x* como única variable:*(*x3 + x2 + *x*) ×*(*x2 + *x*).

>>> from sympy import Derivative, Symbol  
>>> x = Symbol('x')  
>>> f = (x\*\*3 + x\*\*2 + x)\*(x\*\*2+x)  
>>> Derivative(f, x).doit()  
(2\*x + 1)\*(x\*\*3 + x\*\*2 + x) + (x\*\*2 + x)\*(3\*x\*\*2 + 2\*x + 1)

Puedes considerar esta función como el producto de dos funciones independientes, lo que significa que, a mano, tendríamos que hacer uso de la *regla del producto* de la diferenciación para hallar la derivada. Pero aquí no tenemos que preocuparnos de eso, porque podemos crear un objeto de la clase Derivative para que lo haga por nosotros.

Prueba con otras expresiones complicadas, como las que implican funciones trigonométricas.

#### ***Una calculadora de derivadas***

Ahora vamos a escribir un programa calculadora de derivadas, que tomará una función como entrada y luego imprimirá el resultado de diferenciarla respecto a la variable especificada:

'''  
Derivative calculator  
'''  
  
from sympy import Symbol, Derivative, sympify, pprint  
from sympy.core.sympify import SympifyError  
  
def derivative(f, var):  
var = Symbol(var)  
d = Derivative(f, var).doit()  
pprint(d)  
  
if \_\_name\_\_=='\_\_main\_\_':  
  
➊ f = input('Enter a function: ')  
var = input('Enter the variable to differentiate with respect to: ')  
try:  
➋ f = sympify(f)  
except SympifyError:  
print('Invalid input')  
else:  
➌ derivative(f, var)

En ➊, pedimos al usuario que introduzca una función para la que se debe hallar la derivada y, a continuación, le pedimos la variable con respecto a la cual se debe diferenciar la función. En ➋, convertimos la función de entrada en un objeto SymPy utilizando la función sympify(). Llamamos a esta función en un bloque try...except para poder mostrar un mensaje de error en caso de que el usuario introduzca una entrada no válida. Si la expresión de entrada es una expresión válida, llamamos a la función derivada en ➌, pasando como argumentos la expresión convertida y la variable con respecto a la cual se va a diferenciar la función.

En la función derivative(), creamos primero un objeto Symbol que corresponde a la variable con respecto a la cual se va a diferenciar la función. Utilizamos la etiqueta var para referirnos a esta variable. A continuación, creamos un objeto Derivative que pasa tanto la función a diferenciar como el objeto símbolo var. Inmediatamente llamamos al método doit() para evaluar la derivada, y a continuación utilizamos la función pprint() para imprimir el resultado de forma que se parezca a su homólogo matemático. A continuación se muestra un ejemplo de ejecución del programa:

Enter a function: 2\*x\*\*2 + 3\*x + 1  
Enter the variable to differentiate with respect to: x  
4·x + 3

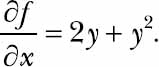
Aquí tienes un ejemplo de ejecución cuando se utiliza con una función de dos variables:

Enter a function: 2\*x\*\*2 + y\*\*2  
Enter the variable to differentiate with respect to: x  
4·x

#### ***Cálculo de derivadas parciales***

En el programa anterior vimos que es posible calcular la derivada de una función multivariable respecto a cualquier variable utilizando la clase Derivative. Este cálculo suele denominarse *diferenciación* parcial, indicando *parcial* que suponemos que sólo varía una variable, mientras que las demás son fijas.

Consideremos la función *f(x*, *y*) = *2xy* + xy2. La diferenciación parcial de *f*(*x*, *y*) respecto a *x* es



El programa anterior es capaz de hallar la derivada parcial porque sólo es cuestión de especificar la variable correcta:

Enter a function: 2\*x\*y + x\*y\*\*2  
Enter the variable to differentiate with respect to: x  
y2 + 2·y

**NOTA**

*Una suposición clave que he hecho en este capítulo es que todas las funciones de las que estamos calculando la derivada son diferenciables en sus respectivos dominios.*

[anterior](ch07_4.html)[Subtema 5 de 11: (Ver todo)](ch07.html)[siguiente](ch07_6.html)