Mathematical Economics

Alpha Chiang

Chapter 7

Rules of Differentiation nand Their Use in Comparative Statics

In [26]:
```
from sympy import Symbol, dsolve, Function, Derivative, Eq
from sympy import exp, sin, sqrt, diff
```

7.1 Rules of Differentiation for a Function of One Variable

In [21]:
```
#  Example 1
y = Function("y")
x = Symbol('x')

display(Eq(y(x),x**3))
diff(x**3, x)
```

$$y(x) = x^3$$

Out[21]: $3x^2$

Example 4

In [4]:
```
display(Eq(y(x),1/x**3))
diff(1/x**3, x)
```

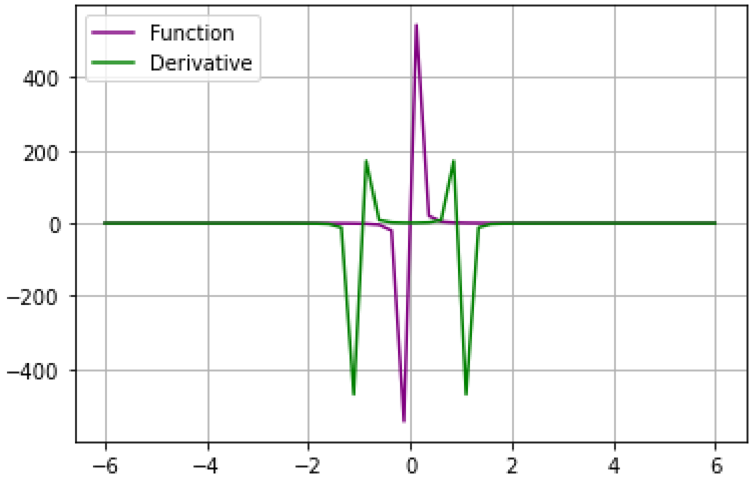
$$y(x) = \frac{1}{x^3}$$

Out[4]: $-\frac{3}{x^4}$

Using this way:

https://www.geeksforgeeks.org/how-to-calculate-and-plot-the-derivative-of-a-function-using-python-matplotlib/

In [5]:
```
import matplotlib.pyplot as plt
from scipy.misc import derivative
import numpy as np

# defining the function
def function(x):
    return 1/x**3

def deriv(x):
    return derivative(function, x)


y = np.linspace(-6, 6)


plt.plot(y, function(y), color='purple', label='Function')


plt.plot(y, deriv(y), color='green', label='Derivative')

plt.legend(loc='upper left')
plt.grid(True)
```



Example 5

Before running this code, we should run the first two code again !!!

We should do this before taking diff every time !

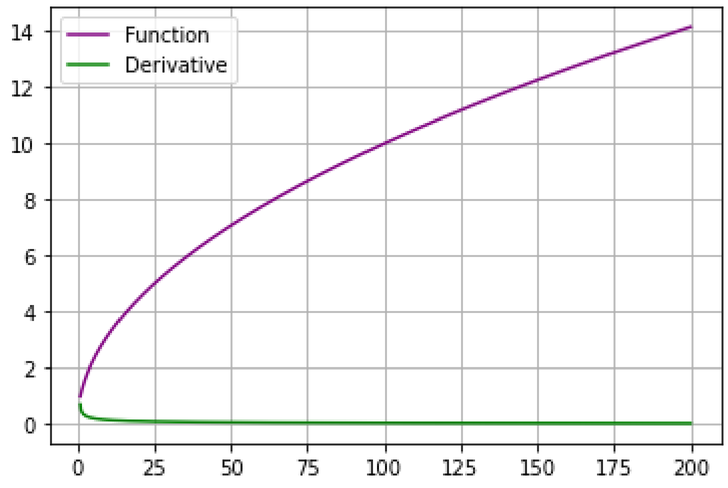
In [13]:
```
display(Eq(y(x),sqrt(x)))
diff(sqrt(x), x)
```

$$y(x) = \sqrt{x}$$

Out[13]:

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

In [14]:
```
import matplotlib.pyplot as plt
from scipy.misc import derivative
import numpy as np

def function(x):
    return x**(1/2)

def deriv(x):
    return derivative(function, x)

y = np.linspace(1, 200,1000)

plt.plot(y, function(y), color='purple', label='Function')
plt.plot(y, deriv(y), color='green', label='Derivative')
plt.legend(loc='upper left')
plt.grid(True)
```



EXERCISE 7.1 -- Q3(b) --

In [15]:

```
c = Symbol('c')
a = Symbol('a')
b = Symbol('b')
y = Function("y")
u = Symbol('u')
display(Eq(y(u), a*u**(b)))
diff(a*u**(b),u)
```

$$y(u) = au^b$$

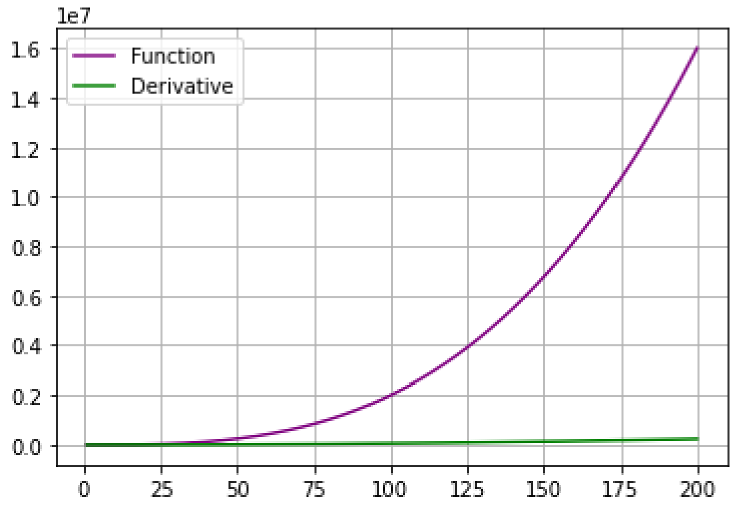
Out[15]:

$$\frac{abu^b}{u}$$

In [16]:

```
import matplotlib.pyplot as plt
from scipy.misc import derivative
import numpy as np

def function(u,a = 2, b = 3):
    return a*u**(b)

def deriv(x):
    return derivative(function, x)

y = np.linspace(1, 200,1000)
```

```
plt.plot(y, function(y), color='purple', label='Function')
plt.plot(y, deriv(y), color='green', label='Derivative')
plt.legend(loc='upper left')
plt.grid(True)
```



7.2 Rules of Differentiation Involving Two or More Functions of the Same Variable

Example 2

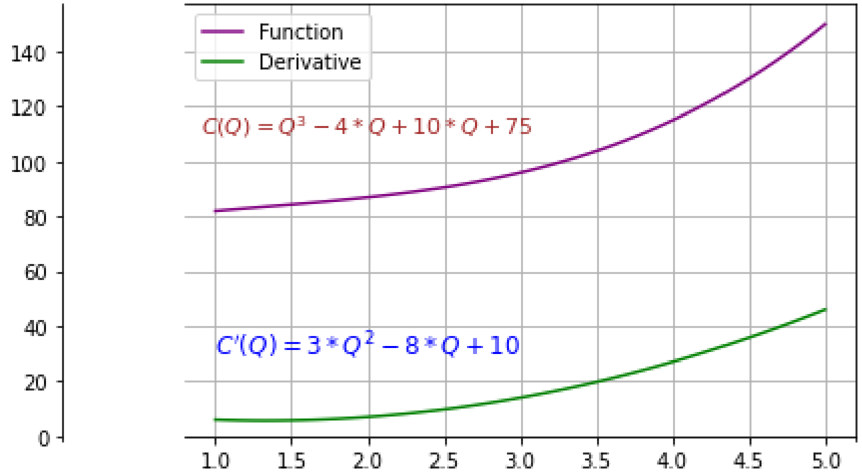
In [17]:
```
C = Function('C')
Q = Symbol('Q')
x = Symbol("x")

display(Eq(C(Q),Q**3 - 4*Q**2 + 10*Q + 75))
diff(Q**3 - 4*Q**2 + 10*Q + 75,Q)
```

$$C(Q) = Q^3 - 4Q^2 + 10Q + 75$$

Out[17]: $3Q^2 - 8Q + 10$

In [18]:
```
import matplotlib.pyplot as plt
from scipy.misc import derivative
import numpy as np

def function(Q):
    return Q**3 - 4*Q**2 + 10*Q + 75
```

```python
def deriv(Q):
    return derivative(function, Q)

y = np.linspace(1, 5)

plt.plot(y, function(y), color='purple', label='Function')
plt.plot(y, deriv(y), color='green', label='Derivative')
plt.legend(loc='upper left')
plt.gca().spines['left'].set_position('zero',)
plt.gca().spines['bottom'].set_position('zero',)
plt.legend(loc='upper left')
plt.text(2, 30, r"$C'(Q)= 3*Q^2 -8*Q +10$",
         horizontalalignment='center',
         fontsize=12, color='blue')
plt.text(2, 110, r'$C(Q)=Q^3- 4*Q +10*Q +75$',
         horizontalalignment='center',
         fontsize=11, color='brown')
plt.grid(True)
```



Relationship Between Marginal-Cost and Average-Cost Functions

--Figure 7.3--

In [28]:
```python
C = Symbol('C')
Q = Symbol('Q')
x = Symbol("x")
M = Symbol("M")
```

```python
A = Symbol("A")
M_C = Symbol("MC")
display(Eq(M_C,Q**3 - 12*Q**2 + 60*Q))
display(diff(Q**3 - 12*Q**2 + 60*Q,Q))
AC = (Q**2 - 12*Q+ 60)
AC
```

$$MC = Q^3 - 12Q^2 + 60Q$$

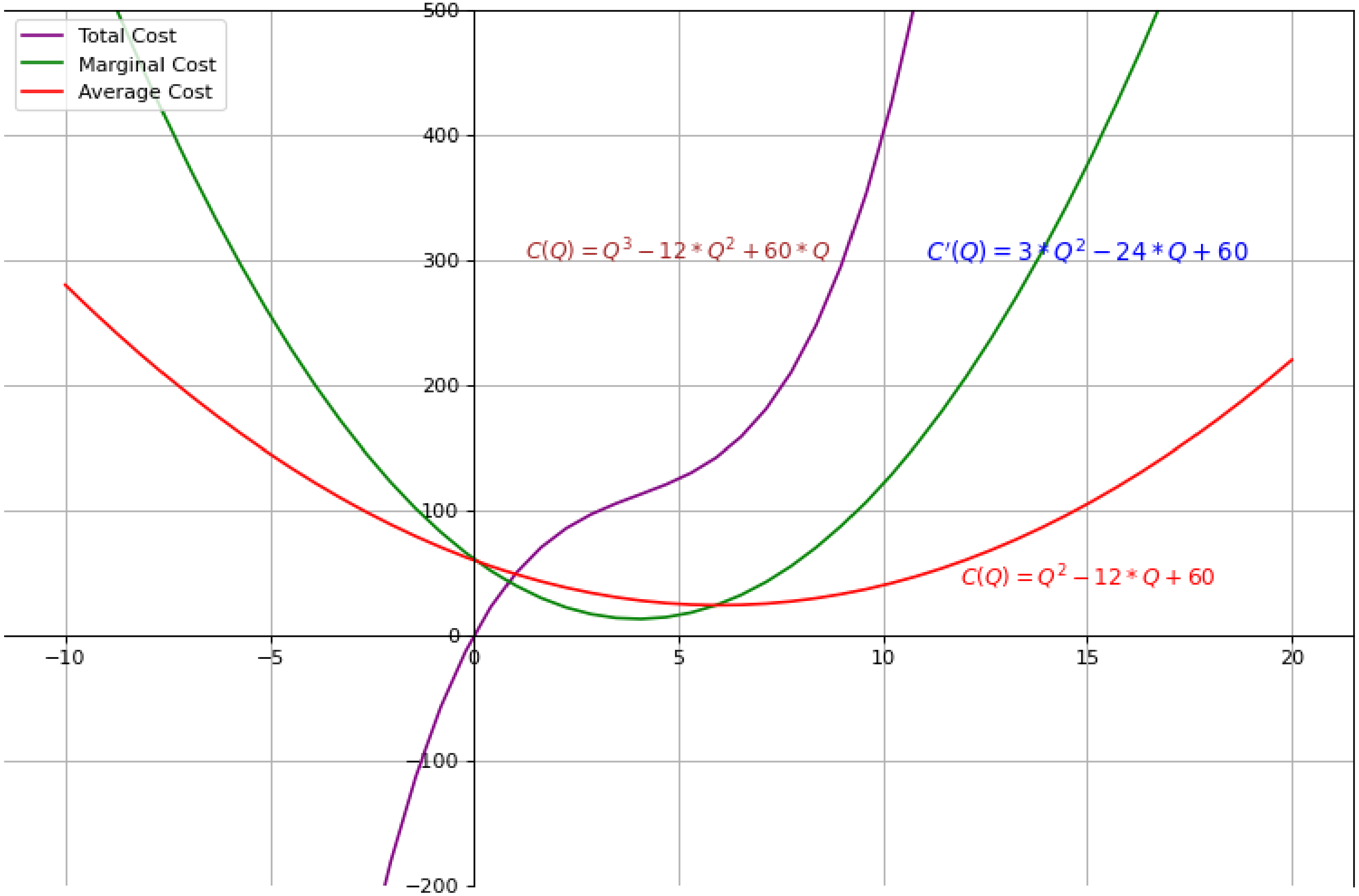
$$3Q^2 - 24Q + 60$$

Out[28]: $$Q^2 - 12Q + 60$$

In [29]:

```python
import matplotlib.pyplot as plt
from scipy.misc import derivative
import numpy as np
from matplotlib.pyplot import figure

def function(Q):
    return Q**3 - 12*Q**2 + 60*Q

def deriv(Q):
    return derivative(function, Q)

def Avecost(Q):
    return (Q**2 -12*Q + 60)

figure(figsize=(12, 8), dpi=80)
y = np.linspace(-10,20)
plt.ylim((-200,500))
plt.plot(y, function(y), color='purple', label='Total Cost')
plt.plot(y, deriv(y), color='green', label='Marginal Cost')
plt.plot(y, Avecost(y), color='red', label='Average Cost')
plt.legend(loc='upper left')
plt.gca().spines['left'].set_position('zero',)

plt.gca().spines['bottom'].set_position('zero',)
plt.legend(loc='upper left')
plt.text(15, 300, r"$C'(Q)= 3*Q^2 -24*Q +60$",
         horizontalalignment='center',
         fontsize=12, color='blue')
plt.text(5, 300, r'$C(Q)=Q^3 -12*Q^2 +60*Q$',
         horizontalalignment='center',
         fontsize=11, color='brown')
```

```python
plt.text(15, 40, r'$C(Q)=Q^2 -12*Q +60$',
         horizontalalignment='center',
         fontsize=11, color='red')
plt.grid(True)
```



In [30]:

```python
import sympy as sy
eq1 = Eq(deriv(Q),Avecost(Q))
eq1
```

Out[30]: $-0.5(Q-1.0)^3 + 6.0(Q-1.0)^2 + 0.5(Q+1.0)^3 - 6.0(Q+1.0)^2 + 60.0 = Q^2 - 12Q + 60$

In [31]:
```
display(sy.solve(eq1))
display(deriv(5.91))
```

```
[0.0845240525773498, 5.91547594742265]
23.944299999999997
```

In [32]:
```
from sympy import Symbol, dsolve, Function, Derivative, Eq
from scipy.misc import derivative
y = Symbol("y")
x = Symbol('x')
f = Function("f")
f2 = Function("f2")

def f(y):
    return 3*y**2

def deriv1(y):
    return derivative(f, y)

def f2(x):
    return 2*x + 5

def deriv2(x):
    return derivative(f2, x)

Chain = deriv1(y)*deriv2(x)
Chain
```

Out[32]: $-3.0(y-1.0)^2 + 3.0(y+1.0)^2$

In [33]:
```
def f(y):
    return y - 3

def deriv1(y):
    return derivative(f, y)

def f2(x):
    return x**3

def deriv2(x):
    return derivative(f2, x)

Chain = deriv1(y)*deriv2(x)
Chain
```

Out[33]: $-0.5(x-1.0)^3 + 0.5(x+1.0)^3$

In [35]:
```
from sympy import symbols
x, y, z = symbols('x y z')
```

In [37]:
```
z = 3*y**2
y = 2*x + 5
diff(z, x)
```

Out[37]: $24x + 60$

In [39]:
```
z = y - 3
y = x**3
diff(z, x)
```

Out[39]: $3x^2$

7.4 Partial Differentiation Techniques of Partial Differentiation

Example 1

In [40]:
```
from sympy import symbols, diff
x1, x2 = symbols('x_1 x_2')
f = Function("f")
f1 = 3*x1**2 + x1*x2 + 4*x2**2
eq1 = Eq(f(x1,x2),3*x1**2 + x1*x2 + 4*x2**2)
display(eq1)
display(diff(f1, x1))
display(diff(f1,x2))
```

$f(x_1, x_2) = 3x_1^2 + x_1x_2 + 4x_2^2$

$6x_1 + x_2$

$x_1 + 8x_2$

In [41]:
```
from sympy import *
res1 = diff(f1, x1)
res1.subs({x1:1, x2:3})
```

Out[41]: 9

In [42]:
```
res2 = diff(f1, x2)
res2.subs({x1:1, x2:3})
```

Out[42]: 25

Example 3

In [43]:
```
from sympy import symbols, diff
u, v, y = symbols('u v y')
f = Function("f")
f2 = (3*u - v)/(u**2 + 3*v)
eq2 = Eq(y,(3*u - v)/(u**2 + 3*v))
display(eq2)
display(diff(f2, u))
display(diff(f2,v))
```

$$y = \frac{3u - v}{u^2 + 3v}$$

$$-\frac{2u\left(3u - v\right)}{\left(u^2 + 3v\right)^2} + \frac{3}{u^2 + 3v}$$

$$-\frac{3\left(3u - v\right)}{\left(u^2 + 3v\right)^2} - \frac{1}{u^2 + 3v}$$

In [44]:
```
res1 = diff(f2, u)
res1.subs({u:2, v:2})
```

Out[44]: $\frac{7}{50}$

In [45]:
```
res2 = diff(f2, v)
res2.subs({u:2, v:2})
```

Out[45]: $-\frac{11}{50}$

EXERCISE 7.4 -- Q5 --

In [46]:
```
from sympy import symbols, diff
x1, x2  = symbols('x_1 x_2')
U = Function("U")
f1 = (x1 + 2)**(2) * (x2 + 3 )**3
eq1 = Eq(U(x1,x2),(x1 + 2)**(2) * (x2 + 3 )**3)
display(eq1)
display(diff(f1, x1))
display(diff(f1,x2))
```

$$U(x_1, x_2) = (x_1 + 2)^2 (x_2 + 3)^3$$

$$(2x_1 + 4)(x_2 + 3)^3$$

$$3(x_1 + 2)^2 (x_2 + 3)^2$$

In [47]:
```
res1 = diff(f1, x1)
res1.subs({x1:3, x2:3})
```

Out[47]: 2160

In [48]:
```
res2 = diff(f1, x2)
res2.subs({x1:3, x2:3})
```

Out[48]: 2700

7.5 Applications to Comparative-Static Analysis National-Income Model

In [49]:
```
from sympy import symbols, diff
Y,alpha,beta,gamma,I0,G0,delta  = symbols('Y \\alpha \\beta \\gamma I_0 G_0 \\delta')
U = Function("U")
Y = (alpha - beta*gamma + I0 + G0)/(1 - beta + beta*delta)
eq1 = Eq(Y,(alpha - beta*gamma + I0 + G0)/(1 - beta + beta*delta))
display(eq1)
display(diff(Y, G0))
display(diff(Y, gamma))
display(diff(Y, delta))
```

True

$$\frac{1}{\beta\delta - \beta + 1}$$

$$-\frac{\beta}{\beta\delta - \beta + 1}$$

$$-\frac{\beta\left(G_{0} + I_{0} + \alpha - \beta\gamma\right)}{\left(\beta\delta - \beta + 1\right)^{2}}$$

In [50]:

```
from sympy import symbols,Matrix
x, y, z = symbols('x,y,z')
A = Matrix(([diff(Y, G0)], [diff(Y, gamma)],[diff(Y, delta)]))
A
```

Out[50]:

$$\left[\begin{matrix}\frac{1}{\beta\delta - \beta + 1}\\ -\frac{\beta}{\beta\delta - \beta + 1}\\ -\frac{\beta\left(G_{0} + I_{0} + \alpha - \beta\gamma\right)}{\left(\beta\delta - \beta + 1\right)^{2}}\end{matrix}\right]$$

Furkan zengin