Finding Derivatives

1. Find the derivative of each of the following functions.

a) $3x^5 - 10x^2 + 8$	j) $x^2 e^x$
b) $(5x^{12} + 2)(\pi - \pi^2 x^4)$	k) $\cos x + e^x$
c) $\sqrt{u} - 3/u^3 + 2u^7$	l) $\sin x / \cos x$
d) $mx + b$ $(m, b \text{ constant})$	m) $e^x \ln x$
e) $.5 \sin x + \sqrt[3]{x} + \pi^2$	n) $\dfrac{2^x}{10 + \sin x}$
f) $\dfrac{\pi - \pi^2 x^4}{5x^{12} + 2}$	o) $\sin(e^x \cos x)$
g) $2\sqrt{x} - \dfrac{1}{\sqrt{x}}$	p) $6e^{\cos t} / 5\sqrt[3]{t}$
h) $\tan z\,(\sin z - 5)$	q) $\ln(x^2 + xe^x)$
i) $\dfrac{\sin x}{x^2}$	r) $\dfrac{5x^2 + \ln x}{7\sqrt{x} + 5}$

$\boxed{1}$ $3x^5 - 10x^2 + 8$ 1-1=0

$15x^4 - 20x + 0$

$\boxed{15x^4 - 20x}$

b) $\overset{a}{(5x^{12}+2)}\overset{b}{(\pi - \pi^2x^4)}$

$a' = 60x^{11}$, $b' = -4\pi^2x^3$

$f'(x) = a \cdot b' + b \cdot a' =$

$-4\pi^2x^3(5x^{12}+2) + 60x^{11}(\pi - \pi^2x^4)$

$= -20\pi^2x^{15} - 8\pi^2x^3 + 60\pi x^{11} - 60\pi^2x^{15}$

$= \boxed{-80\pi^2x^{15} + 60\pi x^{11} - 8\pi^2x^3}$

c) $\sqrt{u} - 3\frac{1}{u^3} + 2u^7$

$u^{1/2}$ $3u^{-3}$

$= \boxed{\frac{1}{2\sqrt{u}} + 9\frac{1}{u^4} + 14u^6}$

$\boxed{d}$ $mx + b$ 1-1=0 1-1=0

m 0

$\boxed{y' = m}$

e) $.\overset{a}{5}\overset{b}{\sin}(x) + \sqrt[3]{x} + \pi^2$

$a' = 0$ $b' = \cos(x)$ const, 0

$0 \cdot \sin(x) + .5\cos(x)$ $x^{1/3}$

$\boxed{y' = .5\cos(x) + \frac{1}{3\sqrt[3]{x^2}}}$

f) $\dfrac{\pi - \pi^2x^4}{5x^{12}+2}$ a b

$a' = -4\pi^2x^3$ $b' = 60x^{11}$ quotient rule

$y' = \dfrac{-4\pi^2x^3(5x^{12}+2) - 60x^{11}(\pi - \pi^2x^4)}{(5x^{12}+2)^2}$

simplify

$= \dfrac{-20\pi^2x^{15} - 8\pi^2x^3 - 60\pi x^{11} + 60\pi^2x^{15}}{25x^{24} + 10x^{12} + 10x^{12} + 4}$ → not useful I'm putting it back

$\boxed{= \dfrac{40\pi^2x^{15} - 60\pi x^{11} - 8\pi^2x^3}{(5x^{12}+2)^2}}$

g) $2\sqrt{x} - \frac{1}{\sqrt{x}}$

$= 2x^{1/2} - x^{-1/2}$

diff $y' = x^{-1/2} + \frac{1}{2}x^{-3/2}$

$\boxed{= \frac{1}{\sqrt{x}} + \frac{1}{2\sqrt{x^3}}}$

h) $\overset{a}{\tan(z)}\overset{b}{(\sin(z) - 5)}$

diff $a' = \frac{1}{\cos^2(x)}$ $b' = \cos(z)$

product rule $y' = \cos(z)\tan(z) + \frac{\sin(z) - 5}{\cos^2(x)}$

simplify $\boxed{y' = \sin(z) + \frac{\sin(z) - 5}{\cos^2(z)}}$

i) $\dfrac{\sin(x)}{x^2}$ a b $a' = \cos(x)$ $b' = 2x$

$y' = \dfrac{x^2\cos(x) - 2x\sin(x)}{(x^2)^2}$

$\boxed{= \dfrac{x^2\cos(x) - 2x\sin(x)}{x^4}}$

(j) $x^2 e^x$

$a' = 2x$

$b' = e^x$

$y' = 2xe^x + x^2e^x$

$= (2x + x^2)e^x$

(k) $y = \cos(x) + e^x$

$\boxed{y' = -\sin(x) + e^x}$

(l) $\sin(x) / \cos(x)$ ok... I'll do the quotient rule

$a' = \cos(x)$ $b' = -\sin(x)$

$y' = \frac{\cos(x)\cos(x) + \sin(x)\sin(x)}{\cos^2(x)}$

$y' = \frac{\cos^2(x) + \sin^2(x)}{\cos^2(x)} \rightarrow$ trig. $= 1$

$y' = \frac{1}{\cos^2(x)}$

(m) $e^x \ln(x)$

$a' = e^x$ $b' = 1/x$

$\boxed{y' = e^x \ln(x) + e^x/x = y + \frac{e^x}{x}}$

(n) $\frac{2x}{10 + \sin(x)}$

$a' = 2$ $b' = \cos(x)$

$\boxed{y' = \frac{2(10 + \sin(x)) - 2x\cos(x)}{(10 + \sin(x))^2}}$

(o) $\sin(e^x \cos(x))$

$a_1' = e^x$ $a_2' = -\sin(x)$

$b' = \cos(a)$

$y' = \cos(e^x\cos(x)) \cdot a'$

$a' = e^x\cos(x) - e^x\sin(x)$

$\boxed{y' = \cos(e^x\cos(x)) \cdot (e^x\cos(x) - e^x\sin(x))}$

(p) $6e^{\cos(t)} / 5\sqrt[3]{t}$

$a_1' = -\sin(t)$ $a_2' = a_1 6e^{a_1}$

$a' = 6e^{\cos(t)} \cdot -\sin(t)$

$b' = \frac{5}{3}\frac{1}{\sqrt[3]{t^2}}$

$y' = \frac{(-\sin(t)6e^{\cos(t)} \cdot 5\sqrt[3]{t}) - (6e^{\cos(t)} \cdot \frac{5}{3\sqrt[3]{t^2}})}{25t^{2/3}}$

(q) $\ln(x^2 + xe^x)$

$a' = 1/b$

$b' = 2x + xe^x$

$\boxed{y' = \frac{2x + xe^x}{x^2 + xe^x} =}$

(r) $\frac{5x^2 + \ln(x)}{7\sqrt{x} + 5}$

$a' = 10x + 1/x$

$b' = \frac{7}{2\sqrt{x}}$

$\boxed{y' = \frac{(10x + 1/x)(7\sqrt{x} + 5) - (5x^2 + \ln(x))(\frac{7}{2\sqrt{x}})}{(7\sqrt{x} + 5)^2}}$

eh $= \frac{(70\sqrt{x^3} + \frac{7\sqrt{x}}{x} + 50x + 5/x) - (\frac{35x^2}{2\sqrt{x}} + \frac{7\ln(x)}{2\sqrt{x}})}{49x + 70\sqrt{x} + 25}$

not interesting :/

$= 70\sqrt{x^3} + 50x + \frac{7\sqrt{x} + 5}{x} - \frac{35x^2 + 7\ln(x)}{2\sqrt{x}}$

2. Suppose f and g are functions and that we are given

$$f(2) = 3, \qquad g(2) = 4, \qquad g(3) = 2,$$
$$f'(2) = 2, \qquad g'(2) = -1, \qquad g'(3) = 17.$$

Evaluate the derivative of each of the following functions at $t = 2$:

a) $f(t) + g(t)$

b) $5f(t) - 2g(t)$

c) $f(t)g(t)$

d) $\dfrac{f(t)}{g(t)}$

e) $g(f(t))$

f) $\sqrt{g(t)}$

g) $t^2 f(t)$

h) $(f(t))^2 + (g(t))^2$

i) $\dfrac{1}{f(t)}$

j) $f(3t - (g(1+t))^2)$

k) What additional piece of information would you need to calculate the derivative of $f(g(t))$ at $t = 2$?

l) Estimate the value of $f(t)/g(t)$ at $t = 1.95$

a) $y' = f'(t) + g'(t)$
$= f'(2) + g'(2)$
$= 2 + -1$
$\boxed{= 1}$

b) $5f(t) - 2g(t)$
$y' = 5f'(2) - 2g'(t)$
$= 10 - (-2)$
$\boxed{= 12}$

c) $f(t)g(t)$
$y' = f'(t)g(t) + f(t)g'(t)$
$= (2 \cdot 4) + (3 \cdot -1)$
$= 8 + -3 \quad \boxed{= 5}$

d) $f(t)/g(t)$
$y' = \dfrac{f'(t)g(t) - f(t)g'(t)}{g(2)^2}$
$= \dfrac{(2 \cdot 4) - (3 \cdot -1)}{4^2}$
$= \dfrac{8+3}{16} \boxed{= \dfrac{11}{16}}$

e) $g(f(t))$
$y' = f'(2) \cdot g'(f(2))$
$= 2 \cdot g'(3)$
$= 2 \cdot 17$
$\boxed{= 34}$

f) $\sqrt{g(t)}$
$y' = \dfrac{g'(2)}{2\sqrt{g(2)}} = \dfrac{-1}{2\sqrt{4}}$
$\boxed{= \dfrac{-1}{4}}$

g) $t^2 f(t)$
$y' = 2t f(t) + t^2 f'(t)$
$= (4 \cdot 3) + (4 \cdot 2)$
$= 12 + 8 \boxed{= 20}$

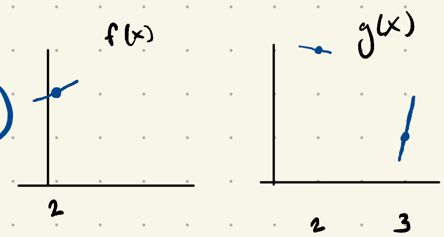
h) $f(t)^2 + g(t)^2$
$y' = 2f(t) \cdot f'(t) + 2g(t) \cdot g'(t)$
$= (2 \cdot 3 \cdot 2) + (2 \cdot 4 \cdot -1)$
$= 12 + (-8) \quad \boxed{= 4}$

i) $1/f(t) = f(t)^{-1}$
$y = -f(2)^{-2} \cdot f'(2)$
$= -\dfrac{1}{3^2} \cdot 2$
$\boxed{= -\dfrac{2}{9}}$

j) $f(3t - (g(1+t))^2)$
$y' = f'(3t - (g(1+t))^2) \cdot (3 - (2(g(1+t)) \cdot (g'(1+t) \cdot 1))$
$y' = f'(6 - g(3)^2) \cdot (3 - 2g(3)) \cdot (g'(3))$
$= f'(6 - 4) \cdot ((3 - 4) \cdot 17)$
$= f'(2) \cdot -17$
$= 2 \cdot -17$
$\boxed{= -34}$

k) $f(g(t))$
$y' = f'(g(t)) \cdot g'(t)$
$f'(g(2)) \cdot g'(2) =$
$\boxed{f'(4)} \cdot -1$
$f'(4)$ is needed to answer the question

l)



- $f(1.95)$ is likely slightly smaller and $g(1.95)$ is likely larger, so $f(t)/g(1.95)$ is likely between 1/2 and 3/4

Algebraic Derivative Solver

1. define inputs + outputs
   - input str, regex?
   - Output: float / int
2. define basic derivatives
   - $mx+b$
   - x^n
   - $\sin(x)$,
   - $\cos(x)$
   - e^x
   - $\ln(x)$

   dict
3. Solving functions for basic derivatives
4. implement chain rule, product rule, multiple functions
   - Start w/ 2 functions
   - next, 3
5. mult. function recognition via regex?

Pseu o

Octo er 24, 2024

0.1 Pseudocode for lgebraic Derivative Solver in Python

I thought bout this for while, nd identified few steps to implement bottom-up ppro ch to building this progr m:

1. **Define b sic deriv tives nd solving functions** For this step, I would implement n individu l function for e ch of the "b sic" deriv tive functions s outlined on p ge 276, ch pter 5.1 of C lculus in Context. Depending on whether or not I decide to implement the deriv tive finder s cl ss, I might h ve the form s cl ss ttribute th t determines wh t solving function to use.

2. **Implement functions for ch in rule, product rule, quotient rule for 2-3 functions** This step will use the b sic functions defined bove to solve more complex equ tions comprised of 2 or more functions. For simplicity it will likely be just 2 or 3 to begin with.

3. **Bre k Input Down:** This step is the one I h ve the most trepid tion round, s I m not sure th t I will be ble to consistently sep r te the regul r expression or string input of l rge equ tion into the constituent b sic deriv tives nd how they re rel ted.

4. **Wr pper Function:** This l st function will m ke sure th t ll of the functions identified in the input re n lyzed nd put b ck together. This function will build b ck up by c lcul ting the v lue for b sic inner functions first, nd working outw rds until ll identified function st tes re solved

Pseudocode for e ch step is s follows:

[3]:
```python
#Defines basic derivatives and relevant parameters
#for constants "c" is a constant multiplying the whole function and "a" multiplies x
#implemented basic chain rule for constant a, where something like y=sin(2x) evaluates to y'=2*cos(2x)

#dictionary will be used to classify functions and identify different constants.

#basic_dict = {"linear": [m, b], "polynomial":[r, c], "sine": [c, a], "cosine": [c, a], "exp":[c, a], "log":[c, a]}

def line r_deriv(m, b, x):
    return m

def poly_deriv(x, r, c = 1):
```

```
    return (r*c) * (x**(r-1))

def sin_deriv(x, c,   = 1):
    return c *   * np.cos(  * x)

def cos_deriv(x, c,  =1):
    c = -c
    return c *   * np.sin(  * x)

def exp_deriv(c,  , x):
    return c *   * np.exp(  * x)

def log_deriv(c,  , x):
    return (c *  )/(  * x)

# lternatively, as classes, these could be written as :
cl ss Line r:
    def __init__(self, expr):
        self.const nts = [1, 1]
    def get_const nts(self, expr):
        #code to get constants from regex or string
        m = 1
        b = 1
        self.const nts = [m, b]

    def get_deriv tive(self):
        #this function would likely return a second function.
        def lin_deriv:
            return self.const nts[0]
        return lin_deriv

#and so on for other types of basic function
```

[ ]:
```
# Next would be the functions for product, quotient, chain rule:

def product_deriv(f, g, x):
    #f and g are already classified as basic function types and have associated
   derivative functions
    #first, calculate already defined terms for f(x) and g(X)
    f_v l = f(x)
    g_v l = g(x)

    #get derivative functions (get derivative may need to be recursive?)
    f_der = f.get_deriv tive()
    g_der = g.get_deriv tive()
```

```
    #calculate f'(x) and g'(x)
    f2_v l = f_der(x)
    g2_v l = g_der(x)

    #finally return products:
    return (f2_v l * g_v l) + (f_v l * g2_v l)

def quotient_deriv(f, g, x):
    #f and g are already classified as basic function types and have associated
   derivative functions
    #first, calculate already defined terms for f(x) and g(X)
    f_v l = f(x)
    g_v l = g(x)

    #get derivative functions
    f_der = f.get_deriv tive()
    g_der = g.get_deriv tive()

    #calculate f'(x) and g'(x)
    f2_v l = f_der(x)
    g2_v l = g_der(x)

    return ((f2_v l * g_v l) - (f_v l * g2_v l))/(g_v l**2)


def ch in_deriv(f, g, x):
    #get value for inner func first
    g_v l = g(x)

    g_der = g.get_deriv tive()
    f_der = f.get_deriv tive()

    f2_v l = f_der(g_v l)
    g2_v l = g_der(x)

    return f2_v l * g2_v l
```

The next p rts of the code would be working with regul r expressions or strings, which could be used to sep r te the input expression on p rentheses, oper tions, or something simil r to isol te individu l expressions.

I'm not 100% sure wh t this would look like, but my thoughts re something like this:

def get_functions(expression/string): - split functions on oper tors nd/or p rentheses - check to m ke sure they fit specific form t of b sic expression - cre te list of functions sep r ted by oper tors - list is sh pe (# of dded/subtr cted terms or terms sep r te by + or - x # of functions from inner to outer) - work through list until functions c n be consolid ted into numbers nd dded or subtr cted return fin l v lues