03 Exercises

September 13, 2020

0.1 Exercise 03.1

Compare the computed values of

$$d_0 = a \cdot b + a \cdot c$$

and

$$d_1 = a \cdot (b + c)$$

when $a = 100$, $b = 0.1$ and $c = 0.2$. Store d_0 in the variable `d0` and d_1 in the variable `d1`.

Try checking for equality, e.g. `print(d0 == d1)`.

[0]:
```python
a = 100
b = 0.1
c = 0.2

d0 = (a*b)+(a*c)
d1 = (a*(b+c))

print(d0, d1, d0 == d1)
#raise NotImplementedError()
```

```
30.0 30.000000000000004 False
```

[0]:
```python
assert d0 == 30.0
assert d1 != 30.0
assert d0 != d1
```

0.2 Exercise 03.2

For the polynomial

$$\begin{aligned} f(x, y) &= (x + y)^6 && (1) \\ &= x^6 + 6x^5 y + 15x^4 y^2 + 20x^3 y^3 + 15x^2 y^4 + 6xy^5 + y^6 && (2) \end{aligned}$$

compute f using: (i) the compact form $(x + y)^6$; and (ii) the expanded form for:

(a) $x = 10$ and $y = 10.1$

(b) $x = 10$ and $y = -10.1$

and compare the number of significant digits for which the answers are the same. Store the answer for the compact version using the variable `f0`, and using the variable `f1` for the expanded version.

For case (b), compare the computed and analytical solutions and consider the relative error. Which approach would you recommend for computing this expression?

(a) $x = 10$ **and** $y = 10.1$

[0]:
```
x = 10.0
y = 10.1

f0=(x+y)**6
f1=(x**6)+(6*x**5*y)+(15*x**4*y**2)+(20*x**3*y**3)+(15*x**2*y**4)+(6*x*y**5)+(y**6)

print(f0, f1)

#raise NotImplementedError()
```

```
65944160.60120103 65944160.601201
```

[0]:
```
import math
assert math.isclose(f0, 65944160.60120103, rel_tol=1e-10)
assert math.isclose(f1, 65944160.601201, rel_tol=1e-10)
```

(b) $x = 10$ **and** $y = -10.1$

[0]:
```
x = 10.0
y = -10.1

f0=(x+y)**6
f1=(x**6)+(6*x**5*y)+(15*x**4*y**2)+(20*x**3*y**3)+(15*x**2*y**4)+(6*x*y**5)+(y**6)

print(f0, f1)
#raise NotImplementedError()
```

```
9.999999999999788e-07 9.958166629076004e-07
```

[0]:
```
import math
assert math.isclose(f0, 1.0e-6, rel_tol=1e-10)
assert math.isclose(f1, 1.0e-6, rel_tol=1e-2)
```

0.3 Exercise 03.3

Consider the expression

$$f = \frac{1}{\sqrt{x^2-1}-x}$$

When x is very large, the denominator approaches zero, which can cause problems.

Try rephrasing the problem and eliminating the fraction by multiplying the numerator and denominator by $\sqrt{x^2-1}+x$ and evaluate the two versions of the expression when:

(a) $x = 1 \times 10^7$

(b) $x = 1 \times 10^9$ (You may get a Python error for this case. Why?)

(a) $x = 1 \times 10^7$

[3]:
```
import math
x = 1.0e7
f = 1/((math.sqrt(x**2-1)-x))
print(f)
```

```
-19884107.85185185
```

(b) $x = 1 \times 10^9$

[4]:
```
import math
x = 1.0e9
f = 1/((math.sqrt(x**2-1)-x))
print(f)
#raise NotImplementedError()
```

```
␣
↪---------------------------------------------------------------------------

    ZeroDivisionError                         Traceback (most recent call␣
↪last)

    <ipython-input-4-aa90f0e4703a> in <module>()
      1 import math
      2 x = 1.0e9
----> 3 f = 1/((math.sqrt(x**2-1)-x))
      4 print(f)
      5 #raise NotImplementedError()


    ZeroDivisionError: float division by zero
```

[5]:
```
import math
x = 1
while x <= 1.0e9:
```

```
    f = 1/((math.sqrt(x**2-1)-x))
    x *=10
    print(x, f, math.sqrt(x**2-1)-x)

#raise NotImplementedError()
```

```
10 -1.0 -0.05012562893380057
100 -19.94987437106615 -0.005000125006247913
1000 -199.99499987509284 -0.0005000001250436981
10000 -1999.9994998253328 -5.000000055588316e-05
100000 -19999.99977764674 -4.999994416721165e-06
1000000 -200000.22333140278 -5.00003807246685e-07
10000000 -1999984.77112922 -5.029141902923584e-08
100000000 -19884107.85185185 0.0
```

```
       ␣
↪---------------------------------------------------------------------------

       ZeroDivisionError                         Traceback (most recent call␣
↪last)

       <ipython-input-5-b320edc48f80> in <module>()
         2 x = 1
         3 while x <= 1.0e9:
   ----> 4     f = 1/((math.sqrt(x**2-1)-x))
         5     x *=10
         6     print(x, f, math.sqrt(x**2-1)-x)


       ZeroDivisionError: float division by zero
```