In [4]:

```
#4.
import torch
import math
import numpy as np
#Manual Derivative
def true_grad(x0,x1):
    return np.array([
        2*x0*np.exp(x1),
        x0*x0*np.exp(x1) + 2*np.cos(x1)*-np.sin(x1)
    ])
print(true_grad(3,3.141))
#Using Backward AD Pytorch
x = torch.tensor([3, 3.141], requires_grad=True)
print(x)
f= x[0]**2*torch.exp(x[1])+torch.cos(x[1])
f.backward()
x.grad
```

```
[138.76189369 208.14402584]
tensor([3.0000, 3.1410], requires_grad=True)
```

Out[4]:

```
tensor([138.7619, 208.1423])
```