MachineEpsilon.py

```
# numpy is the fundamental package for scientific computing in python
import numpy as np


def machine_epsilon(func):
    """
    Returns:
         Machine Epsilon
    Arguments:
        func -- function converts a floating number to float32, float64
        float32 - Single Precision float
        float64 - Double Precision float
    """
    x = func(0)
    # Single Precision
    if func is np.float32:
        min_x = func(-126)  # least possible value of power, x, for a 32-bit system
    # Double Precision
    elif func is np.float64:
        min_x = func(-1023)  # least possible value of power, x, for a 64-bit system

    while func(1)+func(2**x) > func(1) and min_x <= x:
        # When loop breaks, prev_x stores the smallest power of x for which
        # 1+2**x > 1, and x stores a value such that 1+2**x  = 1
        prev_x = func(x)
        x += func(-1)

    return func(2**prev_x)


print("Machine Epsilon for single precision = ", machine_epsilon(np.float32))

print("Machine Epsilon for Double precision = ", machine_epsilon(np.float64))
```

Output:

```
Machine Epsilon for single precision =  1.19209e-07
Machine Epsilon for Double precision =  2.22044604925e-16
```

The below four points helped me in finding the machine epsilon for both single precision and double precision in python language:

- ✓ Machine epsilon for a specific computer is defined as the smallest power of 2, 2^X, such that 1 + 2^X > 1 on that machine.
- ✓ Numpy is the fundamental package for scientific computing with Python and supports a much greater variety of numerical types like float32, float64
- ✓ float32 – Single Precision float, 8 bits exponent, 23 bits mantissa
- ✓ float64 – Double Precision float, 11 bits exponent, 52 bits mantissa

SOLUTION:

$1 + 2^0 > 1$

$1 + 2^{-1} > 1$

-

-

-

-

$1 + 2^X > 1$ --------→ Machine Epsilon: 2^X

$1 + 2^{X+1} = 1$

So, I started with the X = 0 and looped until $1+2^X > 1$

When the loop breaks at X+1, where $1+ 2^{X+1} = 1$, the previous instance of X+1, which is X , is the smallest power such that **$1+2^X > 1$** - (1.1)

Therefore, machine epsilon $e_m \rightarrow 2^X$

Single Precision:

Range of exponent, X, -126 <= X <= 127

Since my initial condition is X = 0 all the values -126 <=X <= 0 are considered

Double Precision:

Range of exponent, X, -1023 <= X <= 1024

Since my initial condition is X = -1 all the values -1023 <=X <= 0 are considered

machine_epsilon (func):

A single parameter func, which is a data type, is used as a function to convert floating point numbers into float32, float64

In the function body, at each instance the value of X is decremented by 1 and the previous instance of X is stored in a variable, prev_x, such that when the while condition breaks for some variable, X+1, ($1+2^{X+1} = 1$) then prev_x stores the smallest power, X, where $1+2^X >1$ →Machine Epsilon (same as (1.1))

MY RESULTS:

Machine Epsilon for single precision = 1.19209e-07

→1.19209 * 10^{-7} →A number in decimal format can be stored up to 7 decimal digits in a 32-bit system

Machine Epsilon for Double precision = 2.22044604925e-16

→2.22044604925 * 10^{-16} →A number in decimal format can be stored up to 16 decimal digits in a 64-bit system

In a 64-bit system I can store up to 16 decimal digits

REFERENCES:

1. http://www.numpy.org/
2. https://docs.scipy.org/doc/numpy-1.13.0/user/basics.types.html