BLG202E – ASSIGNMENT1

Student Name: Ahmet Furkan Kavraz

Student Num: 150190024

Q4)

The whole code and some result of test cases are in hw4.py.

y ^ (1/3) = a ^ (1/3) * 2 ^ (e/3)

We need to find y ^ (1/3). For this we need to find “a ^ (1/3)” and “2 ^ (e/3)”.

“a ^ (1/3)” can be found with Newton Iteration.

“2 ^ (e/3)” can be found with integer and floating point operations.

a)

If we have “a ^ (1/3)” then we should only calculate “2 ^ (e/3)”. In last we multiply with each other. We can split “2 ^ (e/3)” as a 2 ^ (IntegerPart) * 2 ^ (FloatPart). IntegerPart = e//3 and FloatPart = e%3. “2 ^ (IntegerPart)” has no flops but “2 ^ (FloatPart)” has flops according to result of “e%3”. “2 ^ (FloatPart)” has same flops with number of “e%3”. “a ^ (1/3)” * “2 ^ (IntegerPart)” has 1 flop. And lastly multiplying the result of last sentence and “2 ^ (FloatPart)” has “e%3” flop and we have “y^(1/3)”. In the result, we have maximum 3 flops.

b)

Formula of Newton iteration => X(n+1) = X(n)*(2/3) + (a/3) / (Xn ^ 2) , while abs(X(n+1) – X(n) > 2^(-52)). In each iteration we have 4 flop. As seen in the code.

1.Flop = X(n)*(2/3)

2.Flop = Xn ^ 2

3.Flop = (a/3) / (2.Flop)

4.Flop = 1.Flop + 2.Flop

c)

Initial approximation is = 0.9. “a” values are in range [0.5, 1]. “a ^ (1/3)” values are in range [0.79, 1]. So, I select the middle point of these boundaries.

The iteration number is nearly 4-5 iterations. We need to find nearly 16 digit after coma because of “2 ^ (-52)”. Newton Method is founding approximately 3 digit by 3 digit. So we need roughly 5 step. As seen in the code.