Numerical Analysis
homework 10: Numerical Integration

Due on Tuesday, May 9, 2017

102061149 Fu-En Wang

1 Introduction

In this assignment, we will implement Newton-Cotes Integration to calculate the following function:

$$f(x) = e^x \tag{1}$$

$$I = \int_0^2 f(x)dx \tag{2}$$

Because the integration of e^x can be evaluated by ourself, so we have the groundtruth:

$$I^* = e^2 - e^0 \approx 6.389056 \tag{3}$$

We will divide the interval 0 to 2 into 12, 24, 96, 192, 384, 768, 1536 subintervals, respectively, and calculate the error with different order. For 12 subintervals, we will create 13-vector X and Y:

$$h = \frac{2}{12}$$
$$X[k] = k * h$$
$$Y[k] = e^{X[k]}$$

X, Y and order will be our input parameter of function.

2 Implementation

Algorithm 1 Newton-Cotes Integration

```
Input is order, VEC X, VEC Y
len = size of X
W is weight of the order.
sum = 0
for each i ∈ {0, order, 2∗order, 3∗order, ...., len-2} do
    for each k ∈ {0, 1, 2, ..., order} do
        sum += W[k] * Y[i+k]
    end for
end for
return sum * (X[1] - X[0])
```

3 Discussion

From Section 1, we have already known that:

$$I^* \approx 6.389056$$

As a result, we will calculate the error with different subintervals and order in this section.

3.1 Experiment Result

For convinient purpose, I will refer I_k^n and E_k^n as my answer and error with k subintervals and order n, respectively.

The following tables show the experiment result.

Table 1: Experiment Result

(a) Result of 12 subintervals

I_{12}^1	6.403839	E_{12}^1	1.48E-02
I_{12}^2	6.389083	E_{12}^2	2.73E-05
I_{12}^3	6.389117	E_{12}^3	6.12E-05
I_{12}^4	6.389056	E_{12}^4	2.86E-07
I_{12}^6	6.389056	E_{12}^6	3.94E-09

(b) Result of 24 subintervals

I_{24}^1	6.392753	E_{24}^1	3.70E-03
I_{24}^2	6.389058	E_{24}^2	1.71E-06
I_{24}^3	6.38906	E_{24}^3	3.85E-06
I_{24}^4	6.389056	E_{24}^4	4.51E-09
I_{24}^6	6.389056	E_{24}^6	1.58E-11

(c) Result of 48 subintervals

I_{48}^1	6.38998	E_{48}^1	9.24E-04
I_{48}^2	6.389056	E_{48}^2	1.07E-07
I_{48}^3	6.389056	E_{48}^3	2.41E-07
I_{48}^4	6.389056	E_{48}^4	7.07E-11
I_{48}^6	6.389056	E_{48}^6	6.04E-14

(d) Result of 96 subintervals

I_{96}^1	6.389287	E_{96}^1	2.31E-04
I_{96}^2	6.389056	E_{96}^2	6.69E-09
I_{96}^3	6.389056	E_{96}^3	1.50E-08
I_{96}^4	6.389056	E_{96}^4	1.10E-12
I_{96}^6	6.389056	E_{96}^6	0

(e) Result of 192 subintervals

I_{192}^1	6.389114	E_{192}^1	5.78E-05
I_{192}^2	6.389056	E_{192}^2	4.18E-10
I_{192}^3	6.389056	E_{192}^3	9.40E-10
I_{192}^4	6.389056	E_{192}^4	1.69E-14
I_{192}^6	6.389056	E_{192}^6	1.78E-15

(f) Result of 384 subintervals

I_{384}^1	6.389071	E_{384}^1	1.44E-05
I_{384}^2	6.389056	E_{384}^2	2.61E-11
I_{384}^3	6.389056	E_{384}^3	5.88E-11
I_{384}^4	6.389056	E_{384}^4	3.55E-15
I_{384}^6	6.389056	E_{384}^6	8.88E-16

(g) Result of 768 subintervals

I_{768}^1	6.38906	E_{768}^1	3.61E-06
I_{768}^2	6.389056	E_{768}^2	1.64E-12
I_{768}^3	6.389056	E_{768}^3	3.66E-12
I_{768}^4	6.389056	E_{768}^4	0
I_{768}^6	6.389056	E_{768}^6	1.78E-15

(h) Result of 1536 subintervals

I_{1536}^1	6.389057	E_{1536}^1	9.03E-07
I_{1536}^2	6.389056	E_{1536}^2	1.07E-13
I_{1536}^3	6.389056	E_{1536}^3	2.27E-13
I_{1536}^4	6.389056	E_{1536}^4	8.88E-16
I_{1536}^6	6.389056	E_{1536}^6	7.99E-15

From Table 1, we can plot Error vs Different Order as shown in Figure 1.

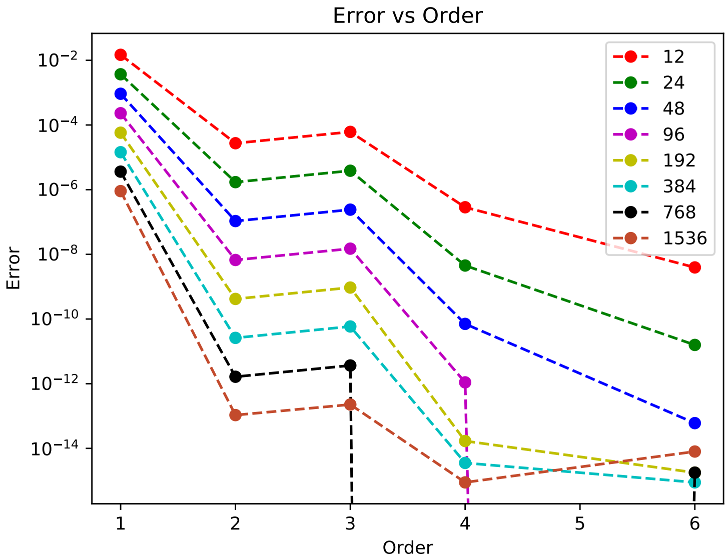

Figure 1: Error vs Order

In Figure 1, we can find that 2-order and 3-order are very close and 3-order have a higher error than 2-order. In addition, because E_{96}^{6} and E_{768}^{4} become 0, so the figure is cut at the two position. To analyze the error more clearly, we can plot Error vs number of subintervals as shown in Figure 2

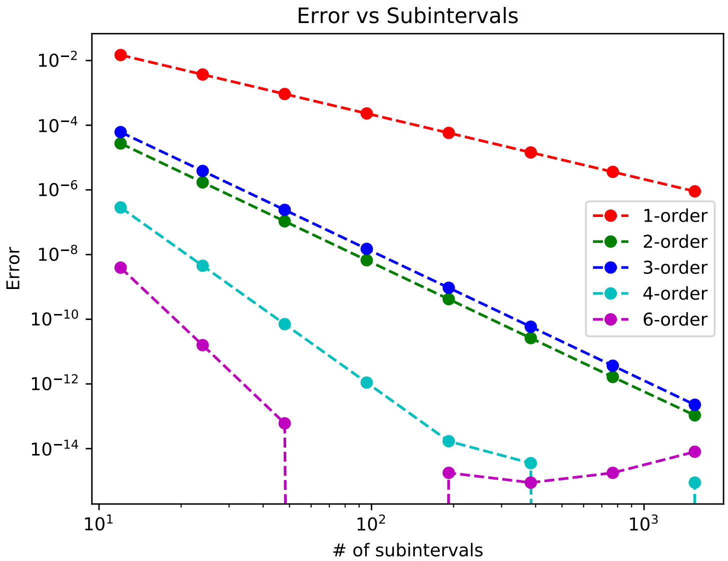

Figure 2: Error vs Subintervals

We can see that 2-order and 3-order are indeed very close. The reason lies on **Theorem 6.1.5** in class material.

$$E_{n,m}(f) = \frac{b-a}{(n+2)!}\frac{H^{n+2}}{n^{n+3}}f^{n+2}(\xi)\int_0^n t\prod_{i=0}^{n}(t-i)dt, \text{ if n is even.} \tag{4}$$

$$E_{n,m}(f) = \frac{b-a}{(n+1)!}\frac{H^{n+1}}{n^{n+2}}f^{n+1}(\xi)\int_0^n \prod_{i=0}^{n}(t-i)dt, \text{ if n is odd.} \tag{5}$$

From this Theorem, we can found that the error of pair [0-order, 1-order], [2-order, 3-order], [4-order, 5-order] and [6-order, 7-order] should be close. However, in our case we only have [2-order, 3-order] and they are close in Figure 2, which satisfies the **Theorem 6.1.5**.
Another strange phenomenon is that 6-order will get larger error after more that 384 subintervals. I think this is because we use **double type** to calculate the data, but double precision number will have **larger error** after 10^{-15}. In other words, the step size is too small for double precision. Maybe we can fix this problem if we use **long double**.