Lab 2: Numerical Integration

Ting Lin, 1700010644

April 8, 2020

Contents

In this section we introduce two kind of numerical integration formula, Cotes formula and Gauss formula. For practical use, we introduce composite integration method and adaptive integration method. We will show their difference by numerical example

1 Numerical Integration Formula

1.1 Interpolation Type and Algebraic Precision

We mainly discuss numerical of **interpolation type**: for a given interval $[a, b]$ and $f \in C[a, b]$, the numerical interpolation respect to a node set $\Delta : x_0 < \cdots x_k \in [a, b]$ is defined as

$$I_N(f) = \int_a^b P_\Delta f$$

where $P_\Delta f \in \mathcal{P}_k$ is the polynomial interpolation of f. An equivalent definition is that $I(f) = \sum_{k=0}^n A_k f(x_k)$ satisfies that $I_N(f) = I(f) := \int_a^b f(x)$ for all polynomials with degree $\leq n$. The equivalence is proved in [CITE].

For a given node set Δ, suppose the interpolation operator $P_\Delta f = \sum_{k=0}^n l_k(x) f(x_k)$, then $A_k = \int_a^b l_k(x)dx$. To describe the error of a given numerical formula $\tilde{I}$, we introduce the definition of algebraic precision.

Definition 1. *If $\tilde{I}(f) = I(f)$ for all polynomials f with degree $\leq m$, but fails for higher degree, we say that the numerical formula is of m**-th order algebraic precision**, or briefly m-th order if there is no ambiguity.*

Clearly if I_n is of interpolation type, then it must be at least n-th order, the converse is also true from the equivalent definition. The further result is statemented below.

Proposition 1. *For $0 \leq k \leq n+1$, $I_N(f)$ is $k+d$-th order if and only if*

1. *I_N is of interpolation type, respect to some node set $x_0 < \cdots, x_n \in [a,b]$.*
2. *$\omega(x) = \prod_{i=0}^{n}(x - x_i)$ is orthogonal to $\mathcal{P}_{k-1}$.*

1.2 Newton–Cotes formula and Gauss formula

We introduce two types of integration formula, under the aforementioned framework. If we choose the node set being $\Delta^{NC}: a, a + \frac{h}{n-1}, a + \frac{2h}{n-1}, \cdots a + h = b$, where $h = b - a$, then we derive the Newton–Cotes formula:

$$I_N^{NC}(f) = \sum_{k=0}^{n} A_k f(x_k),$$

where $x_k = a + \frac{kh}{n-1}$. Direct observation on $\int_a^b \omega(x)$ we have the following proposition.

Proposition 2. *If n is odd, then I_N^{NC} is n-th order. If n is even, the it is $n+1$-th order.*

To seek a formula with higher order, one should recall the Legendre polynomial $L_n(a,b)$, which is orthogonal to $\mathcal{P}$ automatically. Hence if we choose the node set Δ being the roots of L_n, by the proposition we will obtain at least $(n-1) + n = 2n - 1$-th order. (Notice that $L_n(a,b)$ has only n distinct real roots). We derive the Gauss formula:

$$I_N^{G}(f) = \sum_{k=1}^{n} A_k f(x_k),$$

where x_k are the roots of $L_n(a,b)$. Since $I(L_k^2) \neq 0 = I_N^G(L_k^2)$, the Gauss formula is exactly $(2n-1)$-th order.

By considering an affine transform $[a,b] \to [0,1]$, we can easily represent the formula discussed above as

$$I_N(f) = (b-a) \sum_{k=1}^{n} w_k f(a + (b-a)c_k).$$

Here w_k and c_k are independent to $[a,b]$. In practical implementation, we use two arrays to store the value of w_k and c_k, yielding the following algorithm.

Algorithm 1 Simple-Int

```
Input: $f, w, c, a, b$
Output: $I$
1 $I = 0$,  $h = b - a$ for $i = 0 : length(c)$ do
2     $I = I + h * w[i] f(a + h * c[i])$
3 end
4 return I
```

1.3 Composite Integration and Adaptive Integration

Composite integration is to use the simple integration formula, introduce in the previous subsection, into the sub-interval, and sum the result up.

Algorithm 2 Composite-Int

```
Input: $f, a, b, M, w, c$                    /* $M$ is the number of sub-interval */
Output: $I$
$I = 0$
$h = (b-a)/M$ for $i = 0 : M-1$ do
    Calculate $I_i =$ Simple-Int$(f, w, c, a + ih, a + (i+1)h)$
    $I = I + I_i$
end
return $I$
```

For the composite integration formula, we have the following estimates:

Proposition 3. *If the integration formula is of k, then we have*

$$|I_N(f) - I(f)| \leq Ch^{k+1}.$$

A more wiser choice is to use the adaptive integration, for a given tolerance tol(typically $1e-12$ or $1e-16$), we introduce the following algorithm.

Algorithm 3 Adaptive-Int

```
Input: $f, a, b, w, c$
Output: $I$
$m = \frac{a+b}{2}$
if $|$Simple-Int$(f, w, c, a, m) +$ Simple-Int$(f, w, c, m, b) -$ Simple-Int$(f, w, c, a, b)| \leq (b-a)tol$ then
    return Simple-Int$(f, w, c, a, b)$
else
    return Adaptive-Int$(f, w, c, a, m) +$ Adaptive-Int$(f, w, c, m, b)$
end
```

2 Question

In this report we focus on the integral

$$F = \int_0^\infty \frac{x^3}{e^x - 1} dx.$$

Clearly, F is finite, but it is integrated on a infinite interval. Hence we need to do some preprocessing. (Notice if we want to obtain a more accuracy one, we will not adopt the Gauss-Laguerre interpolation)

2.1 Preprocessing

First, rather than computing F, we compute

$$G = \int_0^\infty \frac{x^3}{(e^x-1)e^x}dx$$

instead. Since

$$F - G = \int_0^\infty x^3 e^{-x} dx = \Gamma(3) = 6.$$

A reason that we prefer computing G is that G is decaying more rapidly than F, thus we can use a small M to truncate, that is, using

$$\tilde{G} = \int_0^{30} \frac{x^3}{(e^x-1)e^x}dx$$

instead of G. We first show the error estimates.

Proposition 4.

$$\int_{30}^\infty \frac{x^3}{(e^x-1)e^x}dx < 10^{-16}$$

Proof.

$$\begin{aligned}\int_{30}^\infty \frac{x^3}{(e^x-1)e^x}dx &< \frac{1}{e^{30}-1}\int_{30}^\infty x^3 e^{-x}dx \\ &< \frac{1}{e^{30}-1} * \int_{30}^\infty \frac{(12)!x^3}{x^{12}}dx \\ &< \frac{1}{e^{30}-1}\frac{(12)!}{8(30)^8} \\ &< 10^{-16}\end{aligned}$$

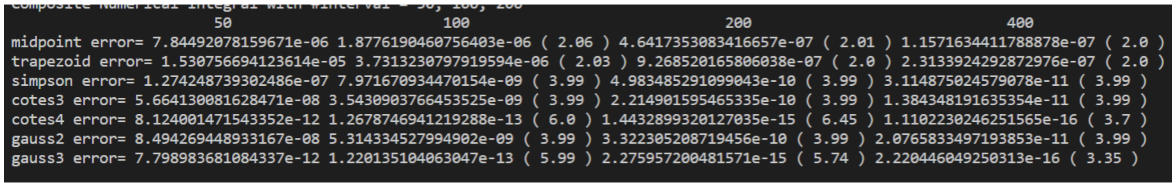
□

2.2 Numerical Result and Discussion

We first test the composite integration and adaptive integration (with $tol = 1e-16$).

	50	100	order	200	order	adaptive node	adaptive result
Trapezoid(Cotes1)	1.53E-03	9.98E-05	3.93	6.31E-06	3.98	-	-
Simpson(Cotes2)	3.77E-04	2.49E-05	3.92	1.57E-06	3.98	9224	0.49393940226683
Cotes3	1.69E-04	1.11E-05	3.93	7.00E-07	3.98	7658	0.493939402266829
Cotes4	1.39E-06	2.24E-08	5.95	3.52E-10	5.98	446	0.4939394022668297
Midpoint(Gauss1)	1.33E-03	8.72E-05	3.93	5.52E-06	3.98	-	-
Gauss2	2.50E-04	1.66E-05	3.91	1.05E-06	3.98	8429	0.4939394022668284
Gauss3	1.33E-06	2.15E-08	5.95	3.38E-10	5.99	443	0.4939394022668287

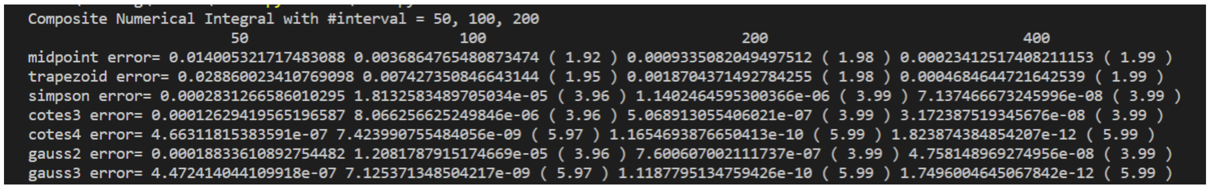
We also test There are two orders violates the theoretical result, the Trapezoid and Midpoint. First we should notice that it does not hold in general. For example, if we try to calculate $\int_0^4 \frac{x^3}{e^x-1}dx$ we obtain the following result, which shows the theoretical result is still convincing.



```
                          50                                           100                                                    200                                                  400
midpoint error= 7.84492078159671e-06 1.8776190460756403e-06 ( 2.06 ) 4.6417353083416657e-07 ( 2.01 ) 1.1571634411788878e-07 ( 2.0 )
trapezoid error= 1.530756694123614e-05 3.7313230797919594e-06 ( 2.03 ) 9.268520165806038e-07 ( 2.0 ) 2.3133924292872976e-07 ( 2.0 )
simpson error= 1.274248739302486e-07 7.971670934470154e-09 ( 3.99 ) 4.983485291099043e-10 ( 3.99 ) 3.114875024579078e-11 ( 3.99 )
cotes3 error= 5.664130081628471e-08 3.5430903766453525e-09 ( 3.99 ) 2.214901595465335e-10 ( 3.99 ) 1.384348191635354e-11 ( 3.99 )
cotes4 error= 8.124001471543352e-12 1.2678746941219288e-13 ( 6.0 ) 1.4432899320127035e-15 ( 6.45 ) 1.1102230246251565e-16 ( 3.7 )
gauss2 error= 8.494269448933167e-08 5.314334527994902e-09 ( 3.99 ) 3.322305208719456e-10 ( 3.99 ) 2.0765833497193853e-11 ( 3.99 )
gauss3 error= 7.798983681084337e-12 1.220135104063047e-13 ( 5.99 ) 2.275957200481571e-15 ( 5.74 ) 2.220446049250313e-16 ( 3.35 )
```

We try to change x^3 to x^2, obtaining the following result.

$$I = \int_0^{30} \frac{x^2}{e^x - 1} dx$$



```
Composite Numerical Integral with #interval = 50, 100, 200
                          50                                           100                                                    200                                                  400
midpoint error= 0.014005321717483088 0.0036864765480873474 ( 1.92 ) 0.0009335082049497512 ( 1.98 ) 0.00023412517408211153 ( 1.99 )
trapezoid error= 0.028860023410769098 0.007427350846643144 ( 1.95 ) 0.0018704371492784255 ( 1.98 ) 0.0004684644721642539 ( 1.99 )
simpson error= 0.0002831266586010295 1.8132583489705034e-05 ( 3.96 ) 1.1402464595300366e-06 ( 3.99 ) 7.137466673245996e-08 ( 3.99 )
cotes3 error= 0.0001262941956519 6587 8.066256625249846e-06 ( 3.96 ) 5.068913055406021e-07 ( 3.99 ) 3.172387519345676e-08 ( 3.99 )
cotes4 error= 4.66311815383591e-07 7.423990755484056e-09 ( 5.97 ) 1.1654693876650413e-10 ( 5.99 ) 1.823874384854207e-12 ( 5.99 )
gauss2 error= 0.00018833610892754482 1.2081787915174669e-05 ( 3.96 ) 7.600607002111737e-07 ( 3.99 ) 4.758148969274956e-08 ( 3.99 )
gauss3 error= 4.472414044109918e-07 7.125371348504217e-09 ( 5.97 ) 1.1187795134759426e-10 ( 5.99 ) 1.7496004645067842e-12 ( 5.99 )
```

We give a proper explanation for this phenomenon.

Proposition 5. *For composite integration method with midpoint (trapezoid) integrator as an inner routine, it holds*

$$|I_N^h(f) - I(f)| = \mathcal{O}(h^4) \tag{1}$$

if $\int_a^b f''(x)\,dx = 0$.

Proof. Assume $b - a = nh$. For midpoint rule, denote the middle point of each sub-interval as $c_j = a + (j + \frac{1}{2})h$. By Taylor expansion at b_j,

$$f(x) = f(c_j) + (x - c_j)f'(c_j) + \frac{(x - c_j)^2}{2} f''(c_j) + \frac{(x - c_j)^3}{6} f'''(c_j) + \mathcal{O}(h^4), \tag{2}$$

$$\begin{aligned} \int_{a+jh}^{a+(j+1)h} f(x)\,dx &= hf(c_j) + \frac{h^3}{24} f''(c_j) + \mathcal{O}(h^4) \\ &= hf(c_j) + \frac{h^2}{24} \int_{a+jh}^{a+(j+1)h} f''(x)\,dx + \mathcal{O}(h^4). \end{aligned} \tag{3}$$

Thus the error estimate of composite method $I_N(f)$ reads

$$\begin{aligned} I(f) - I_N(f) &= \int_a^b f(x)\,dx - \sum_{j=0}^{n-1} hf(c_j) \\ &= \int_a^b f''(x)\,dx + \mathcal{O}(h^4) \\ &= \mathcal{O}(h^4). \end{aligned} \tag{4}$$

As to trapezoid rule, applying the result derived above to the interval $[b_0, b_{n-1}]$ yields

$$
\begin{aligned}
\int_{c_0}^{c_{n-1}} f(x)\,dx &= \sum_{j=1}^{n-1} hf(a+jh) + \int_{c_0}^{c_{n-1}} f''(x)\,dx + \mathcal{O}(h^4), \\
\int_a^{c_0} f(x)\,dx + \int_{c_{n-1}}^{b} f(x)\,dx &= \frac{h}{2}(f(a)+f(b)) + \int_a^{c_0}(x-a)f'(a)\,dx + \int_{c_{n-1}}^{b}(x-b)f'(b)\,dx \\
&\quad + \frac{h^2}{24}(\int_a^{c_0} f''(x)\,dx + \int_{c_{n-1}}^{b} f''(x)\,dx) + \mathcal{O}(h^4) \\
&= \frac{h}{2}(f(a)+f(b)) + \frac{h^2}{24}(\int_a^{c_0} f''(x)\,dx + \int_{c_{n-1}}^{b} f''(x)\,dx) + \mathcal{O}(h^4), \qquad (5) \\
I(f) - I_N(f) &= \int_a^b f(x)\,dx - \sum_{j=1}^{n-1} hf(a+jh) - \frac{h}{2}(f(a)+f(b)) \\
&= \int_a^b f''(x)\,dx + \mathcal{O}(h^4) \\
&= \mathcal{O}(h^4).
\end{aligned}
$$

□

2.3 Romberg Technique

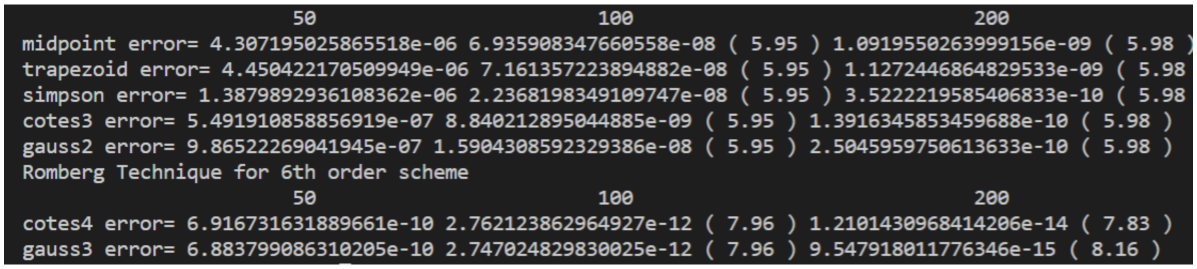
Also we can utilize the Romberg technique to obtain higher order. Suppose the method has order k, we then write

$$I_{N,h}(f) = I(f) + C_k h^k + C_{k+1} h^{k+1} + C_{k+2} h^{k+2} + \cdots$$

Then simple calculation shows that

$$I^R_{N,h}(f) = (2^k I_{N,h/2}(f) - I_{N,h}(f))/(2^k - 1)$$

is a approximation of at least $k+1$ order. Moreover, if $C_{k+1} = 0$ then the scheme is of $k+2$ order. We test the idea in numeric, see the figure below, that is exactly we desired.



```
                      50                         100                             200
midpoint error= 4.307195025865518e-06 6.935908347660558e-08 ( 5.95 ) 1.0919550263999156e-09 ( 5.98 )
trapezoid error= 4.450422170509949e-06 7.161357223894882e-08 ( 5.95 ) 1.1272446864829533e-09 ( 5.98
simpson error= 1.3879892936108362e-06 2.2368198349109747e-08 ( 5.95 ) 3.5222219585406833e-10 ( 5.98
cotes3 error= 5.491910858856919e-07 8.840212895044885e-09 ( 5.95 ) 1.3916345853459688e-10 ( 5.98 )
gauss2 error= 9.86522269041945e-07 1.5904308592329386e-08 ( 5.95 ) 2.5045959750613633e-10 ( 5.98 )
Romberg Technique for 6th order scheme
                      50                         100                             200
cotes4 error= 6.916731631889661e-10 2.762123862964927e-12 ( 7.96 ) 1.2101430968414206e-14 ( 7.83 )
gauss3 error= 6.883799086310205e-10 2.747024829830025e-12 ( 7.96 ) 9.547918011776346e-15 ( 8.16 )
```

2.4 Conclusion

In this lab we test several integration formula in solving a simple question. We show the composite integration, Romberg technique and adaptive integration. Rather than believing the conservative estimation, we show the error and order numerically. Also we observe that in some special cases, the order will be lifted in 2-orders (like what we see in trapezoid and midpoint scheme). Also, we found that the adaptive integration is practical.