

数理工学実験
関数の補間と数値積分

滝本 亘（学籍番号 1029-33-1175）

2022 年 12 月 5 日

1 課題 4.2

1.1 はじめに

区間 $[a, b]$ ($a < b$) における $n + 1$ 個の分点を以下で定める:

$$x_i = a + \frac{b-a}{n}i \quad (i = 0, 1, \dots, n) \quad (1.1)$$

(1)

以下に示す 3 つの関数 $f(x)$ に対して、分点 x_i における値を求めよ。 $n = 2, 4, 8, 16$ として Lagrange 補間で n 次多項式 $P(x)$ を求め、グラフを図示せよ。

$$(a) f_1(x) = \ln(x) \quad x \in [1, 2] \quad (\ln = \log_e) \quad (1.2)$$

$$(b) f_2(x) = 1/x^3 \quad x \in [0.1, 2] \quad (1.3)$$

$$(c) f_3(x) = 1/(1 + 25x^2) \quad x \in [-1, 1] \quad (1.4)$$

ヒント: いくつかのグラフを一つの図に表示するなど比較しやすいよう工夫せよ。 $P(x)$ を得る方法は次の 2 通りである。

- $P(x)$ は理論的に得られるので、gnuplot などコードを書く。
- 分点 $\{x_i\}$ よりさらに細かい分点 $\{y_j\}$ に対して $P(y_j)$ を計算する。

(2)

区間 $x \in [a, b]$ ($a < b$) で定義された関数 $f(x)$ の定積分

$$I(f) := \int_a^b f(x) dx \quad (1.5)$$

を考える。以下に示す 4 つの関数と積分区間に対して、理論値 $I(f_i)$ ($i = 4, 5, 6, 7$) を求めよ。

$$(a) f_4(x) = 1/x \quad x \in [1, 2] \quad (1.6)$$

$$(b) f_5(x) = e^{5x} \quad x \in [-1, 1] \quad (1.7)$$

$$(c) f_6(x) = 1 + \sin x \quad x \in [0, \pi] \quad (1.8)$$

$$(d) f_7(x) = 1 + \sin x \quad x \in [0, 2\pi] \quad (1.9)$$

また、定積分を分割数 $n = 2^i$ ($i = 0, 1, \dots, 10$) の (複合) 中点公式、台形公式、Simpson 公式により求めよ。分割数 n のとき数値積分で得た積分値を $I_n(f_i)$ をする。 $I_n(f_i)$ が $n \rightarrow \infty$ で理論値 $I(f_i)$ に収束するか否か、アルゴリズム間で比較できるように図にまとめよ。 $n = 2^i$ となっていることを考慮し、 n 軸のスケールを線形にするか対数にするか考えよ。

(3)

課題 4.2 の 2 において、関数 f に対する数値計算結果と理論値との誤差を

$$E_n(f) = |I_n(f) - I(f)| \quad (1.10)$$

とする（絶対値に注意）。関数 $f_i (i = 4, 5, 6, 7)$ の定積分に対して、 $n = 2^i (i = 0, 1, \dots, 10)$ に対する（複合）中点公式、台形公式、Simpson 公式による誤差を求め、 $E_n(f)$ の n への依存性（線形関数？指数関数？べき関数？）が明確になるよう図示せよ。それぞれの公式に対して指数（ e^{an} や n^a のときの a ）も求めよ。（ f_7 は特殊なので注意すること）

1.2 原理・方法の詳細

1.2.1 Lagrange 補間

相異なる n 個の点 $x_1 < \dots < x_n$ において関数 f の値 $f(x_i)$, ($i = 1, \dots, n$) が与えられたとき、

$$P(x_i) = f(x_i) \quad (i = 1, \dots, n) \quad (1.11)$$

を満たす $n - 1$ 次多項式 $P(x)$ を作り、 $x \in [x_1, x_n]$ における $f(x)$ の値を $P(x)$ と推定することを Lagrange 補間と呼ぶ。このような $n - 1$ 次の補間多項式 $P(x)$ は一意に存在する。実際、 $P(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$ とおくと、与えられた条件は

$$a_0 + a_1x_i + \dots + a_{n-1}x_i^{n-1} = f(x_i) \quad (i = 1, \dots, n) \quad (1.12)$$

となり、この式をまとめて書くと

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_n) \end{pmatrix} \quad (1.13)$$

となる。係数行列の行列式は Vandermonde の行列式と呼ばれ

$$\begin{vmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{n-1} \end{vmatrix} = \prod_{i>j} (x_i - x_j) \neq 0 \quad (1.14)$$

となるため、線形方程式 (1.13) の解は一意に存在する。

式 (1.11) を満たす $n - 1$ 次の補間多項式 $P(x)$ は

$$P(x) = \sum_{i=1}^n f(x_i) l_i(x) \quad (1.15)$$

$$l_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \quad (1.16)$$

で与えられる。実際、 $l_i(x_j) = \delta_{ij}$ より $P(x_i) = f(x_i)$ となる。また $P(x)$ は $n - 1$ 次多項式 $l_i(x)$ の線形結合であるため、高々 $n - 1$ 次の多項式である。これらの事実と上で示した補間多項式の一意性より、式 (1.15) は $f(x)$ の $n - 1$ 次補間多項式である。

1.2.2 Newton-Cotes 公式

定積分

$$I(f) = \int_{\tilde{a}}^{\tilde{b}} f(x) dx \quad (1.17)$$

を近似的に計算するために、1.2.1 節で述べた Lagrange 補間で関数 f を近似する方法を考える。 \tilde{x}_i を n 個の等間隔分点

$$\tilde{x}_i = \tilde{x}_1 + (i-1)h \quad (1 \leq i \leq n, h > 0) \quad (1.18)$$

とし、この分点における関数 $f(x)$ の Lagrange 補間 (1.15) を用いると、

$$\begin{aligned} I(f) &= \int_{\tilde{a}}^{\tilde{b}} f(x) dx \\ &\approx \int_{\tilde{a}}^{\tilde{b}} P(x) dx \\ &= \int_{\tilde{a}}^{\tilde{b}} \sum_{i=1}^n f(\tilde{x}_i) l_i(x) dx \\ &= \sum_{i=1}^n c_i f(\tilde{x}_i) \end{aligned} \quad (1.19)$$

となる。ここで、

$$c_i = \int_{\tilde{a}}^{\tilde{b}} l_i(x) dx \quad (1.20)$$

である。この数値積分公式が n 点 Newton-Cotes 公式である。

1.2.3 中点公式

区間 (\tilde{a}, \tilde{b}) において、

$$\tilde{x}_1 = \frac{\tilde{a} + \tilde{b}}{2} \quad (1.21)$$

の 1 点 Newton-Cotes 公式を考える。 $l_1(x) \equiv 1$ に注意して、式 (1.20) より

$$c_1 = \int_{\tilde{a}}^{\tilde{b}} dx = \tilde{b} - \tilde{a} \quad (1.22)$$

となる。したがって以下の数値積分公式を得る。

$$\int_{\tilde{a}}^{\tilde{b}} f(x) dx \approx (\tilde{b} - \tilde{a}) f(\tilde{x}_1) \quad (1.23)$$

これが中点公式である。

しかしこの公式は、被積分関数を定数で近似して得られる数値積分公式であるため、複雑な形状の被積分関数 $f(x)$ に対して、積分区間全体にこの公式をそのまま適用してもまともな精度は得られない。そこで実際の計算では積分区間 $[a, b]$ を等分して小区間を作り、各区間に対して上の数値積分公式を適用する。

$h = (b - a)/n$, $x_i = a + ih$, ($0 \leq i \leq n$) とおき、積分区間 (a, b) を n 個の小区間 (x_i, x_{i+1}) ($0 \leq i \leq n - 1$) に分割し、各々の小区間に対して中点公式 (1.23) を適用すると、

$$\int_a^b f(x)dx \approx h \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) \quad (1.24)$$

を得る。これが複合中点公式である。

1.2.4 台形公式

区間 (\tilde{a}, \tilde{b}) において

$$\tilde{x}_1 = \tilde{a}, \quad \tilde{x}_2 = \tilde{b} \quad (1.25)$$

の 2 点 Newton-Cotes 公式を考える。分点の間隔は $\tilde{h} = \tilde{x}_2 - \tilde{x}_1 = b - a$ であり、また式 (1.20) より、 $c_1 = c_2 = h/2$ 。ゆえに以下の数値積分公式を得る。

$$\int_{\tilde{a}}^{\tilde{b}} f(x)dx \approx \frac{\tilde{h}}{2} \{f(\tilde{x}_1) + f(\tilde{x}_2)\} \quad (1.26)$$

これが台形公式である。

中点公式と同様にこの公式は、被積分関数を一次関数で近似して得られる数値積分公式であるため、積分区間 $[a, b]$ を等分して小区間を作り、各区間に対して上の数値積分公式を適用する。

$h = (b - a)/n$, $x_i = a + ih$, ($0 \leq i \leq n$) とおき、 n 個の小区間 (x_i, x_{i+1}) に台形公式 (1.26) を適用すると、

$$\int_a^b f(x)dx \approx \frac{h}{2} \left(f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right) \quad (1.27)$$

を得る。これが複合台形公式である。

1.2.5 Simpson 公式

区間 (\tilde{a}, \tilde{b}) において

$$\tilde{x}_i = \tilde{a} + (i - 1)\tilde{h}, \quad (i = 1, 2, 3) \quad (1.28)$$

$$\tilde{h} = \frac{\tilde{b} - \tilde{a}}{2} \quad (1.29)$$

の 3 点 Newton-Cotes 公式を考えると、式 (1.20) より

$$c_1 = \frac{\tilde{h}}{3}, \quad c_2 = \frac{4}{3}\tilde{h}, \quad c_3 = \frac{\tilde{h}}{3} \quad (1.30)$$

ゆえに以下の数値積分公式を得る。

$$\int_{\tilde{a}}^{\tilde{b}} f(x)dx \approx \frac{\tilde{h}}{3} \{f(\tilde{x}_1) + 4f(\tilde{x}_2) + f(\tilde{x}_3)\} \quad (1.31)$$

これが Simpson の公式である。

中点公式、台形公式と同様にこの公式は、被積分関数を二次関数で近似して得られる数値積分公式であるため、積分区間 $[a, b]$ を等分して小区間を作り、各区間に対して上の数値積分公式を適用する。

$h = (b - a)/n$, $x_i = a + ih$, $(0 \leq i \leq n)$ とおき、各小区間に Simpson 公式 (1.31) を適用すると、

$$\int_a^b f(x)dx \approx \frac{h}{6} \left(f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) + 4 \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) \right) \quad (1.32)$$

を得る。これが複合 Simpson 公式である。

1.3 実験方法

(1)

(a) の $n = 2$ の場合のコードを以下に示す。

コード 1: Lagrange 補間

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5  #define n 2.0
6
7  double f(double x){
8      return log(x);
9  }
10
11 double x(double i){
12     return a+(b-a)/n*i;
13 }
14
15 double l(double X, double i){
16     double l=1;
17
18     for(int j=0; j<=n; j++){
19         if(j!=i){
20             l=l*(X-x(j))/(x(i)-x(j));
21         }
22     }
23
24     return l;
25 }
26
27 double P(double X){
28     double p=0;
29
30     for(int i=0; i<=n; i++){
31         p+=f(x(i))*l(X,i);
32     }
33
34     return p;
35 }
36
37 int main(void){
38     for(int i=0; i<=n; i++){
39         printf("%.16f\n", f(x(i)));
40     }
41
42     FILE *fp;
43     fp = fopen("a2", "w");
44
```

```

45     for(double k=a; k<b; k+=(b-a)/100){
46         fprintf(fp, "%.2f_%.16f\n", k, P(k));
47     }
48     fprintf(fp, "%.2f_%.16f\n", b, P(b));
49
50     fclose(fp);
51 }

```

(2)

(a) のコードを以下に示す。

コード 2: 中点公式

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5
6  double f(double x){
7      return 1/x;
8  }
9
10 double x(double i, double n){
11     double h=(b-a)/n;
12     return a+i*h;
13 }
14
15 double I(double n){
16     double u=0;
17
18     for(int i=0; i<n; i++){
19         u+=f((x(i,n)+x(i+1,n))/2);
20     }
21
22     double h=(b-a)/n;
23     double I=h*u;
24
25     return I;
26 }
27
28 int main(void){
29     FILE *fp;
30     fp = fopen("a-1", "w");
31
32     for(int k=0; k<=10; k++){
33         double n=pow(2,k);
34         fprintf(fp, "%.16f_%.16f\n", n, I(n));
35     }
36 }

```

コード 3: 台形公式

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5
6  double f(double x){
7      return 1/x;
8  }
9
10 double x(double i, double n){
11     double h=(b-a)/n;
12     return a+i*h;

```

```

13     }
14
15     double I(double n){
16         double u=0;
17
18         for(int i=1; i<n; i++){
19             u+=f(x(i,n));
20         }
21
22         double h=(b-a)/n;
23         double I=h/2*(f(x(0,n))+2*u+f(x(n,n)));
24
25         return I;
26     }
27
28     int main(void){
29         FILE *fp;
30         fp = fopen("a-2","w");
31
32         for(int k=0; k<=10; k++){
33             double n=pow(2,k);
34             fprintf(fp,"% .16f_%.16f\n", n, I(n));
35         }
36     }

```

コード 4: Simpson 公式

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5
6  double f(double x){
7      return 1/x;
8  }
9
10 double x(double i, double n){
11     double h=(b-a)/n;
12     return a+i*h;
13 }
14
15 double I(double n){
16     double u1=0;
17     double u2=0;
18
19     for(int i=1; i<n; i++){
20         u1+=f(x(i,n));
21     }
22
23     for(int i=0; i<n; i++){
24         u2+=f((x(i,n)+x(i+1,n))/2);
25     }
26
27
28     double h=(b-a)/n;
29     double I=h/6*(f(x(0,n))+2*u1+f(x(n,n))+4*u2);
30
31     return I;
32 }
33
34 int main(void){
35     FILE *fp;
36     fp = fopen("a-3","w");
37
38     for(int k=0; k<=10; k++){
39         double n=pow(2,k);
40
41         fprintf(fp,"% .16f_%.16f\n", n, I(n));

```


$\begin{matrix} 42 \\ 43 \end{matrix} \quad \left. \begin{matrix} \\ \end{matrix} \right\}$

1.4 結果

(1)

$n = k$ のときの分点 x_i の集合を $X(k)$ とすると、

$$X(2) \subset X(4) \subset X(8) \subset X(16) \quad (1.33)$$

であるため、分点 X_i における値は、 $n = 16$ の場合のみ示す。

(a)

表 1: $f_1(x_i)$

i	x_i
0	0.0000000000000000
1	0.0606246218164348
2	0.1177830356563835
3	0.1718502569266592
4	0.2231435513142098
5	0.2719337154836418
6	0.3184537311185346
7	0.3629054936893685
8	0.4054651081081644
9	0.4462871026284195
10	0.4855078157817008
11	0.5232481437645479
12	0.5596157879354227
13	0.5947071077466928
14	0.6286086594223741
15	0.6613984822453650
16	0.6931471805599453

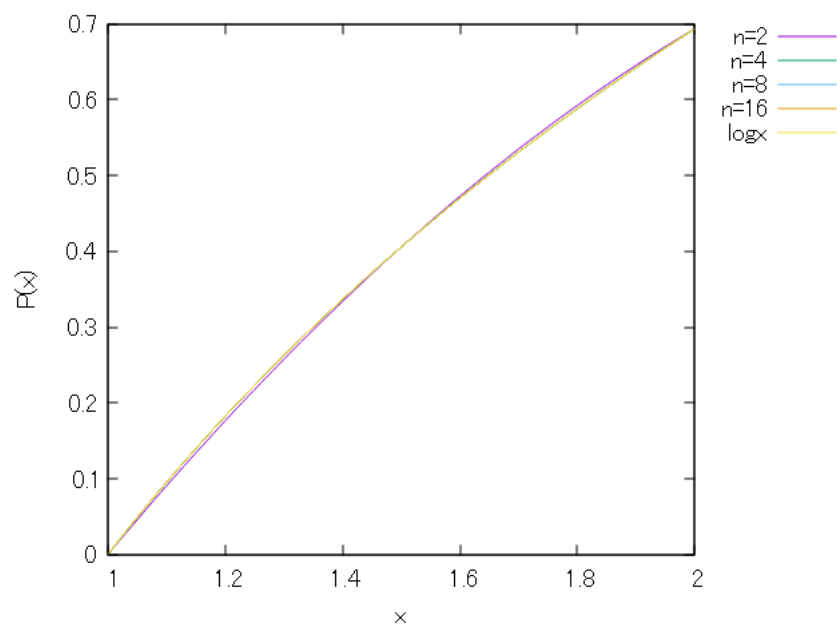


图 1: (a) $f_1(x) = \ln(x)$ $x \in [1, 2]$

(b)

表 2: $f_2(x_i)$

i	x_i
0	999.999999999997700
1	95.5335276967930010
2	26.0122948737489170
7	1.2382294890277936
8	0.8638375985314760
9	0.6263760277819790
10	0.4685525295888176
11	0.3595939643347051
12	0.2819619263286355
13	0.2251607582398476
14	0.1826470335374913
15	0.1501967243399842
16	0.1250000000000000

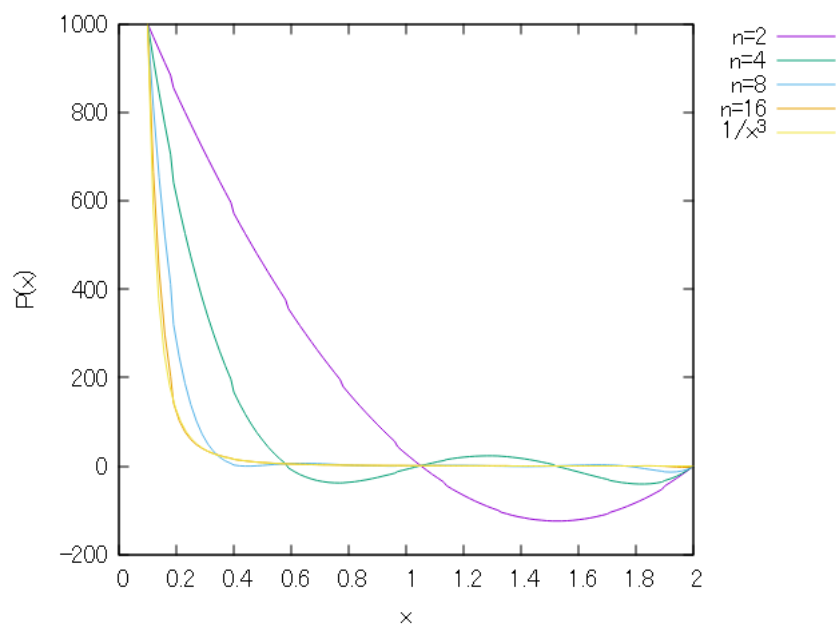


图 2: (b) $f_2(x) = 1/x^3 \quad x \in [0.1, 2]$

(c)

表 3: $f_3(x_i)$

i	x_i
0	0.0384615384615385
1	0.0496508921644686
2	0.0663900414937759
3	0.0928882438316401
4	0.1379310344827586
5	0.2214532871972318
6	0.3902439024390244
7	0.7191011235955056
8	1.0000000000000000
9	0.7191011235955056
10	0.3902439024390244
11	0.2214532871972318
12	0.1379310344827586
13	0.0928882438316401
14	0.0663900414937759
15	0.0496508921644686
16	0.0384615384615385

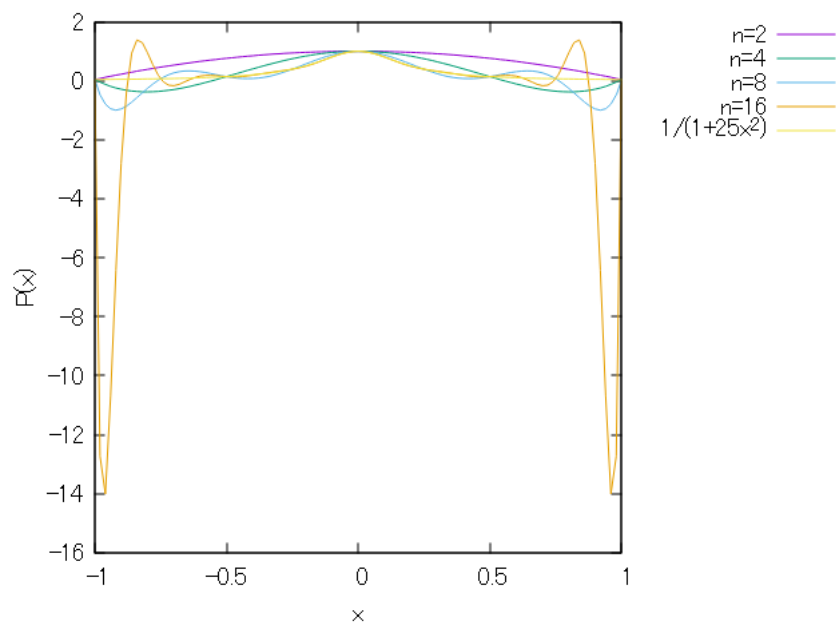


図 3: (c) $f_3(x) = 1/(1 + 25x^2)$ $x \in [-1, 1]$

(2)

各積分の理論値を求める。

(a)

$$\begin{aligned}\int_1^2 f_4(x) dx &= \int_1^2 \frac{1}{x} dx \\ &= [\log x]_1^2 \\ &= \log 2\end{aligned}$$

(b)

$$\begin{aligned}\int_{-1}^1 f_5(x) dx &= \int_{-1}^1 e^{5x} dx \\ &= \left[\frac{e^{5x}}{5} \right]_{-1}^1 \\ &= \frac{e^5 - e^{-5}}{2} \\ &= \frac{2 \sinh 5}{5}\end{aligned}$$

(c)

$$\begin{aligned}\int_0^\pi f_6(x) dx &= \int_0^\pi (1 + \sin x) dx \\ &= [x - \cos x]_0^\pi \\ &= \pi + 2\end{aligned}$$

(d)

$$\begin{aligned}\int_0^{2\pi} f_7(x)dx &= \int_0^{2\pi} (1 + \sin x)dx \\ &= [x - \cos x]_0^{2\pi} \\ &= 2\pi\end{aligned}$$

$n = 2^i$ となっているので、 n 軸のスケールを対数にした片対数グラフを採用している。

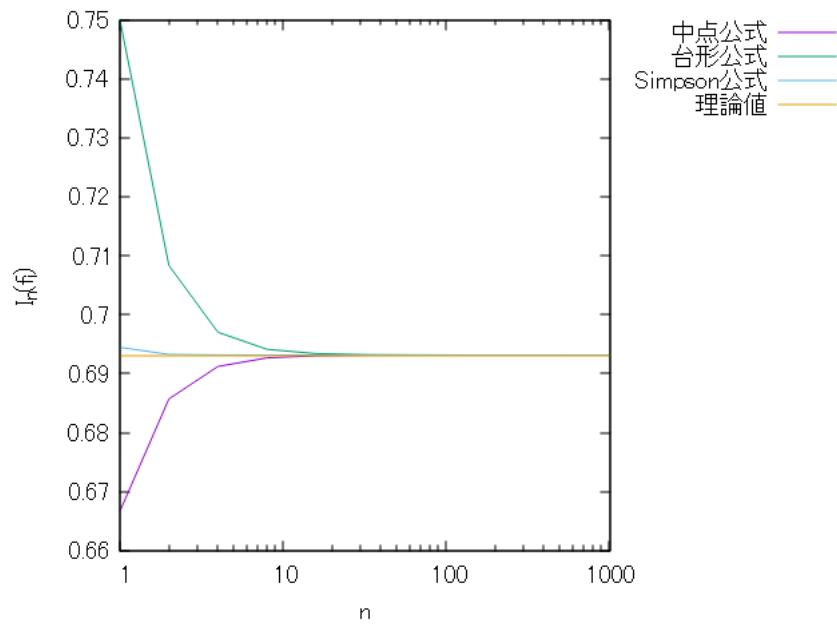


図 4: (a) $f_4(x) = 1/x$ $x \in [1, 2]$

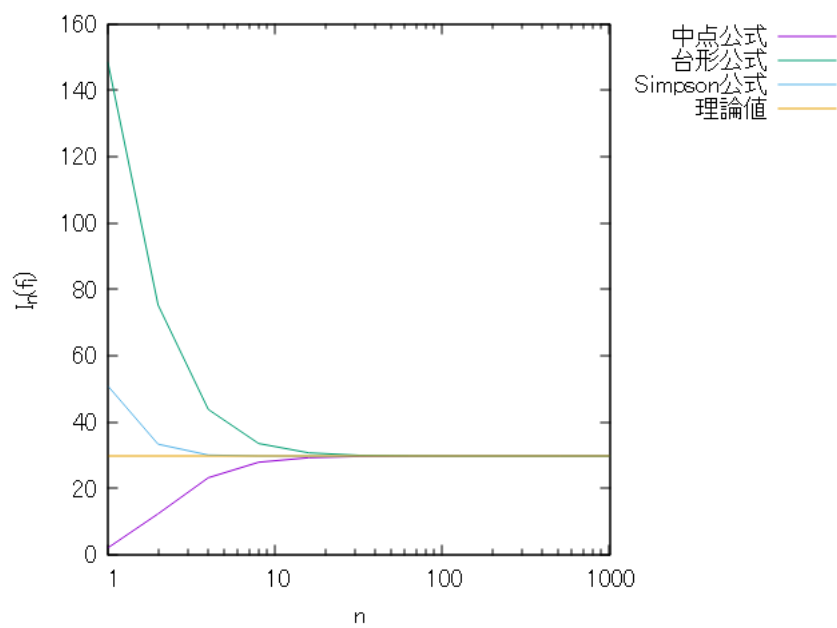


图 5: (b) $f_5(x) = e^{5x} \quad x \in [-1, 1]$

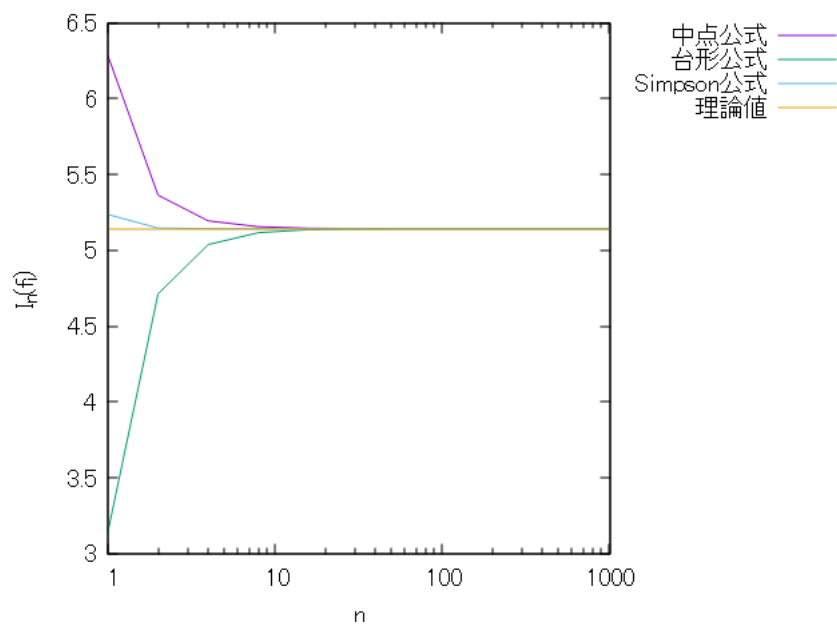


图 6: (c) $f_6(x) = 1 + \sin x \quad x \in [0, \pi]$

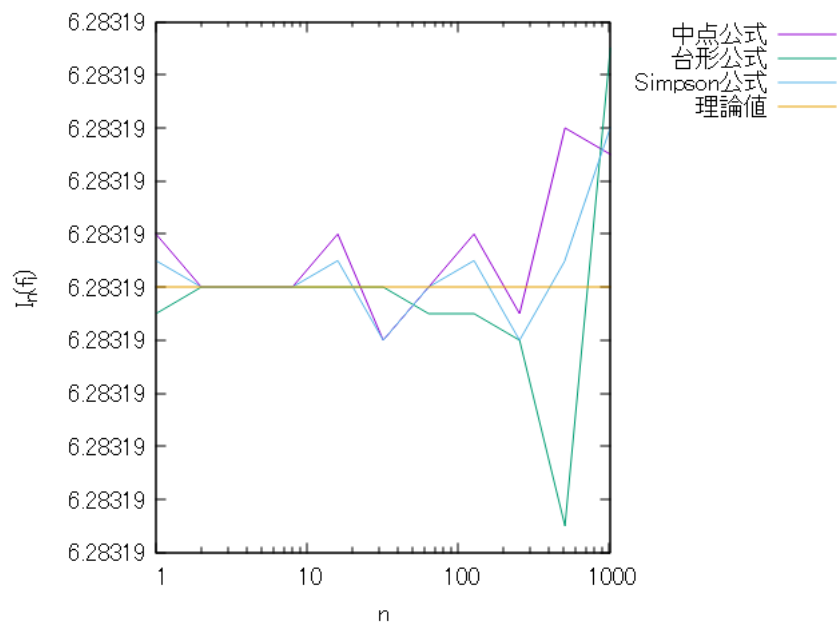


图 7: (d) $f_7(x) = 1 + \sin x \quad x \in [0, 2\pi]$

(3)

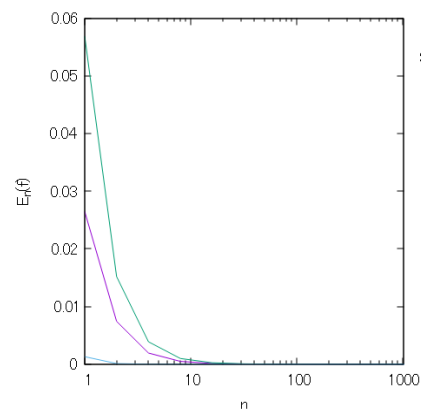


图 8: (a) 片对数

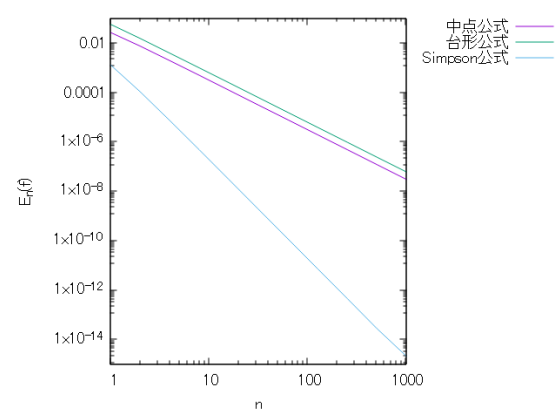


图 9: (a) 两对数

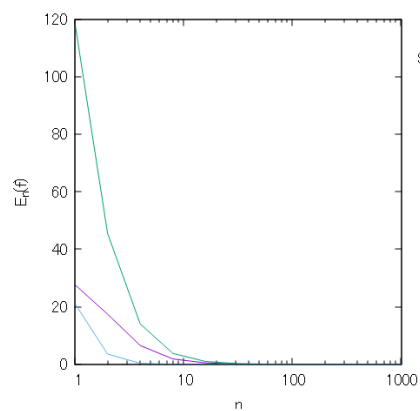


图 10: (b) 片对数

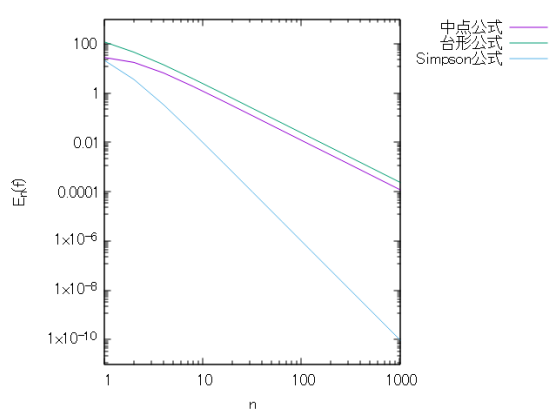


图 11: (b) 两对数

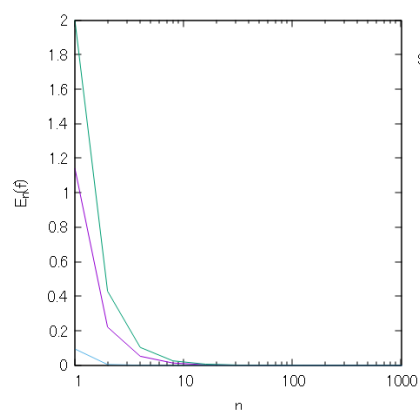


图 12: (c) 片对数

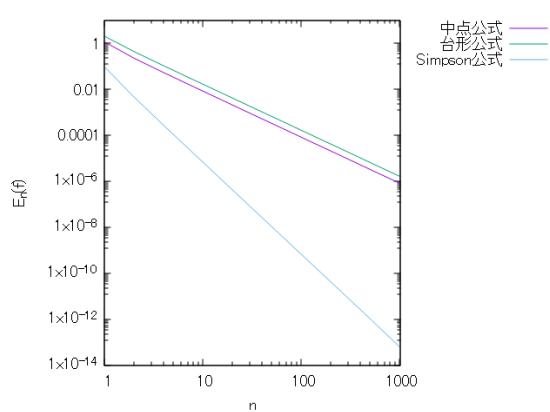


图 13: (c) 两对数

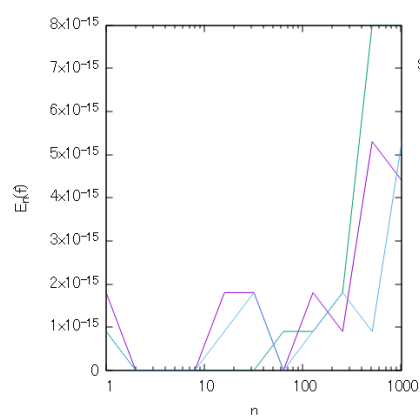


图 14: (d) 片对数

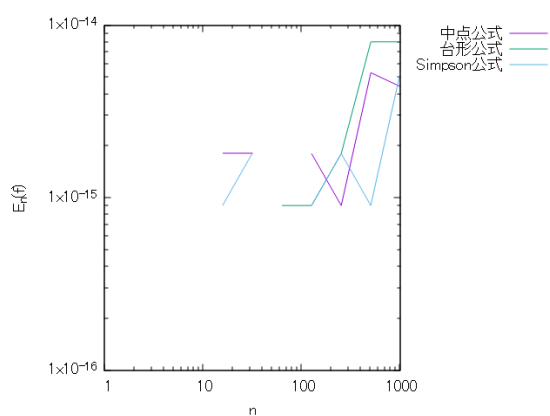


图 15: (d) 两对数

1.5 考察

(1)

- (a) n の値にかかわらず、高精度の近似ができています。
- (b) n の値が大きくなるにつれて、近似の精度も増している。これより、さらに大きい n も用いると、さらに高精度の近似が得られると考えられる。
- (c) $n = 2, 4, 8$ では近似ができていように見えるが、 $n = 16$ のときには絶対値が大きくなると誤差が大きくなっており、近似はできていないことがわかる。

$f_3(x) = 1/(1 + 25x^2)$ の $x = 0$ まわりでのテイラー展開は、

$$f_3(x) = 1 - 25x^2 + 625x^4 + O(x^6) \quad (1.34)$$

となる。係数は

$$a_{2k} = (-25)^k \quad (1.35)$$

となるため、収束半径 r は

$$|(-25r)^2| = 1 \quad (1.36)$$

$$r = 0.2 \quad (1.37)$$

である。よって、 $f_3(x)$ は $|x| < 0.2$ で近似可能と予測できる。実際、この範囲に着目すると、 n の値が大きくなるにつれて、近似の精度も増している。

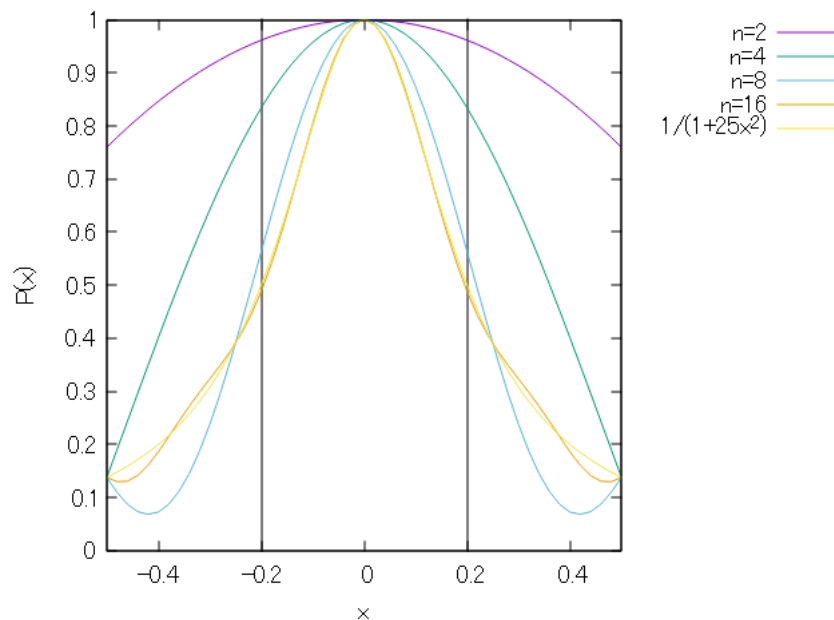


図 16: (c) $f_3(x) = 1/(1 + 25x^2)$ $x \in [-1, 1]$

(2)

- (a) $n = 20$ 辺りから収束している。
- (b) $n = 30$ 辺りから収束している。
- (c) $n = 20$ 辺りから収束している。
- (d) 理論値まわりで振動している。

(3)

(a)(b)(c)

区間 (\tilde{a}, \tilde{b}) における単一区間の中点公式

$$\int_{\tilde{a}}^{\tilde{b}} f(x) dx \approx (\tilde{b} - \tilde{a}) f(\tilde{x}_1) \quad (1.38)$$

の左辺において $f(x)$ を $x = \tilde{x}_1 := (a + b)/2$ 周りで Taylor 展開すると、

$$\begin{aligned} \int_{\tilde{a}}^{\tilde{b}} f(x) dx &\approx \int_{\tilde{a}}^{\tilde{b}} \left(f(\tilde{x}_1) + f'(\tilde{x}_1)(x - \tilde{x}_1) + \frac{1}{2} f''(\tilde{x}_1)(x - \tilde{x}_1)^2 \right) dx \\ &= (\tilde{b} - \tilde{a}) f(\tilde{x}_1) + \frac{(\tilde{b} - \tilde{a})^3}{24} f''(\tilde{x}_1) \end{aligned} \quad (1.39)$$

となる。したがって積分公式 (1.38) の誤差は

$$\int_a^b f(x) dx - (b - a) f(x_1) \approx \frac{(b - a)^3}{24} f''(x_1) \quad (1.40)$$

となる。これを用いると n 点複合中点公式

$$\int_a^b f(x) dx \approx h \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) \quad (1.41)$$

の誤差は

$$\begin{aligned} \int_a^b f(x) dx - h \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) &= \sum_{i=0}^{n-1} \left\{ \int_{x_i}^{x_{i+1}} f(x) dx - h f\left(\frac{x_i + x_{i+1}}{2}\right) \right\} \\ &= \sum_{i=0}^{n-1} \frac{(x_{i+1} - x_i)^3}{24} f''\left(\frac{x_i + x_{i+1}}{2}\right) \\ &= \frac{n}{24} f''(\eta) h^3 \\ &= \frac{(b - a)}{24} f''(\eta) h^2 \end{aligned} \quad (1.42)$$

となる。ただし η は次式を満たす。

$$\sum_{i=0}^{n-1} f'' \frac{x_i + x_{i+1}}{2} = n f''(\eta) \quad (1.43)$$

これより、固定した積分区間 (a, b) における積分に対する複合中点公式の誤差は、分割幅 h に対して h^2 程度のオーダーであると予測される。

複合台形公式、複合 Simpson 公式についても同様の評価を得ることができ、それぞれ

$$-\frac{b-a}{12}f''(\eta)h^2 \quad -\frac{b-a}{2880}f^{(4)}(\eta)h^4$$

となり、それぞれ h^2, h^4 程度のオーダーであると予測される。

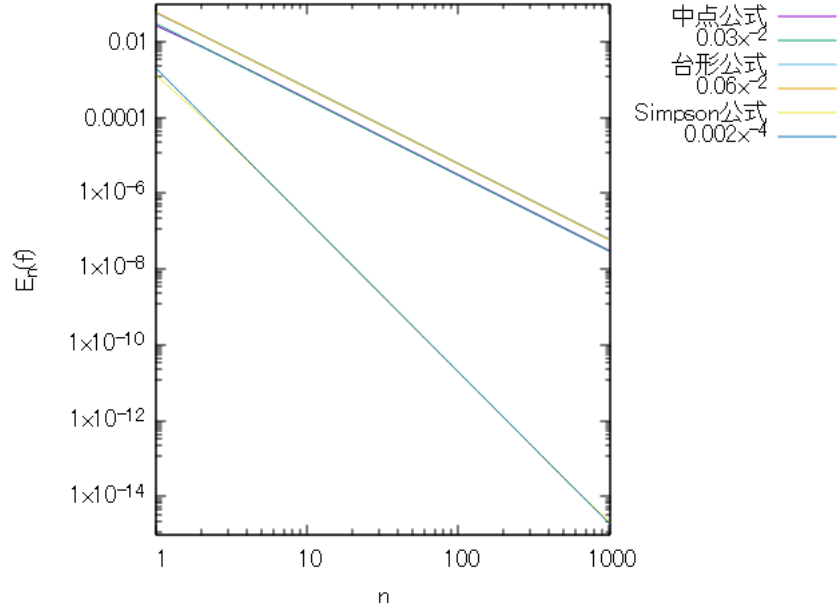


図 17: (a) $f_4(x) = 1/x \quad x \in [1, 2]$

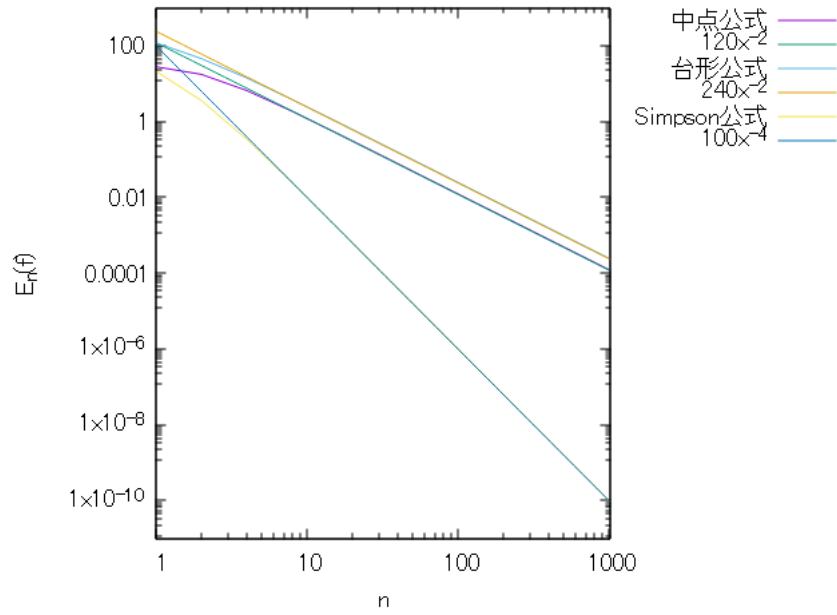


図 18: (b) $f_5(x) = e^{5x} \quad x \in [-1, 1]$

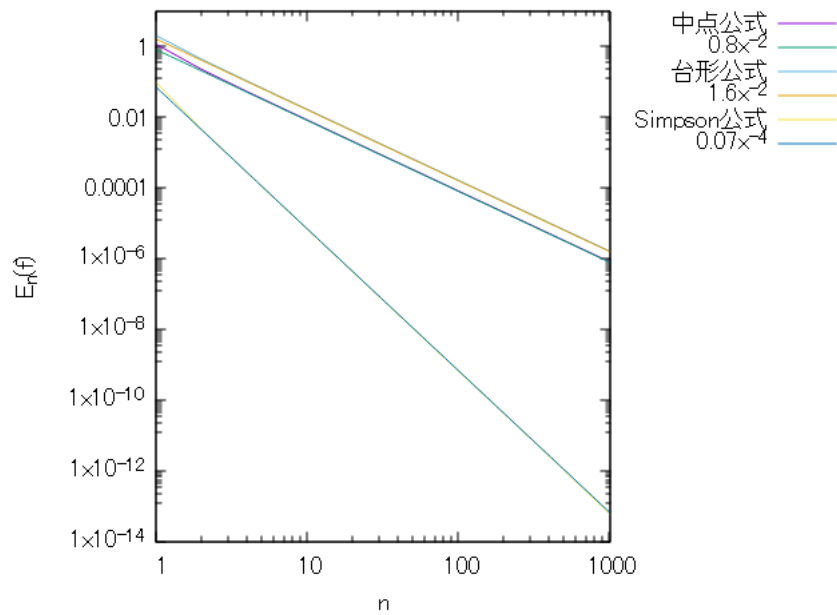


図 19: (c) $f_6(x) = 1 + \sin x \quad x \in [0, \pi]$

実際、中点公式、台形公式の指数は $a = 2$ 、Simpson 公式の指数は $a = 4$ である。また、いずれも精度は台形公式、中点公式、Simpson 公式の順に高くなっている。

- (d) $y = f_7(x)$ のグラフは、点 $(\pi, 1)$ について対称である。よって、 $[x_i, x_{i+1}]$ における誤差と $[x_{n-i-1}, x_{n-i}]$ における誤差は絶対値が同じで符号が相異なる。したがって、分割数 n によらず、積分値は理論値と一致している。また、振動しているように見えるのは、計算機により誤差が生じるためだと考えられる。

2 課題 4.3

2.1 はじめに

(1)

課題 4.2 の 2 における 4 つの関数 $f_i (i = 4, 5, 6, 7)$ の定積分を考える。分割数 $n = 2^i (i = 0, 1, \dots, 10)$ の (複合) M 次 Gauss 型積分公式による数値積分値を $I_n(f_i)$ とし、真値との誤差を $E_n(f_i) = |I_n(f_i) - I(f_i)|$ と定義する。課題 4.2 の 3 と同様に、 $E_n(f_i)$ の n への依存性が明確となるよう図示すると共に、指数の M 依存性を調べよ。ただし M とは分割した小区間内における分点の数である。表 4 に従い、 $M = 2$ および $M = 3$ の Gauss 型積分公式を用いよ。

表 4: Gauss 型積分公式. (a) $M = 2$. (b) $M = 3$.

(a)	m	1	2	(b)	m	1	2	3
	y_m	$-1/\sqrt{3}$	$1/\sqrt{3}$		m	$-\sqrt{3/5}$	0	$-\sqrt{3/5}$
	w_m	1	1		w_m	5/9	8/9	5/9

(2)

以下の定積分を考える

$$A = \int_0^1 \frac{e^{-x}}{\sqrt{x}} dx \simeq 1.49364826562 \quad (2.1)$$

以下では積分区間を $n = 2^i (i = 0, 1, \dots, 10)$ 等分して、 M 次 Gauss 型積分公式を用いる。得られた数値積分値を A_n とし、真値との誤差を $E_n = |A_n - A|$ とする。 $M = 2$ に対して、次に示す 3 通りの方法における精度の違いとその理由について議論せよ。台形公式や Simpson 公式が適用できるかについても議論せよ。

(方法 1) 積分表式 $\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx$ を用いる。

(方法 2) 積分表式 $\int_0^1 \frac{e^{-x}}{\sqrt{x}} dx$ から $t = \sqrt{x}$ と変数変換した積分表式を用いる。

(方法 3) 定積分 A を

$$A = \int_0^1 \frac{1}{\sqrt{x}} dx + \int_0^1 \frac{e^{-x} - 1}{\sqrt{x}} dx \quad (2.2)$$

と変形する。右辺第 1 項の積分を理論的に求め、第 2 項を数値積分で求める。

2.2 原理・方法の詳細

2.2.1 直交多項式

$w(x) > 0$ を区間 (a, b) で定義された連続な関数とし、実係数多項式 $p(x), q(x)$ に対して

$$(p, q)_w = \int_a^b w(x) p(x) q(x) dx \quad (2.3)$$

とおく。このとき、 $(\cdot, \cdot)_w$ は内積を与える。すなわち次の性質を満たすことがわかる。

正値性 $(p, p)_w \leq 0$ で、 $(p, p)_w = 0 \Rightarrow p \equiv 0$

対称性 $(p, q)_w = (q, p)_w$

線形性 多項式 $p(x), q(x), r(x)$ と実数 α, β に対して $(\alpha p + \beta q, r)_w = \alpha(p, r)_w + \beta(q, r)_w$

とくに $(p, q)_w = 0$ のとき、多項式 p, q は重み関数 w に関して直交するという。さらに $\phi_n(x)$, $(n = 0, 1, 2, \dots)$ が n 次多項式 ($\phi_0(x) \neq 0$) で $(\phi_m, \phi_n)_w = 0$, $(m \neq n)$ が満たすとき、 $\{\phi_n\}$ を直交多項式系という。

直交多項式系は以下の性質を満たす。

直交多項式系の存在 任意の内積に対して直交多項式系は存在する。。

直交性 $\phi_n(x)$ は $n - 1$ 次以下の任意の多項式と直交する。

一意性 二つの直交多項式系 $\{\phi_n(x)\}, \{\tilde{\phi}_n(x)\}$ に対して、ある定数 $c_n \neq 0$ が存在して、 $\phi_n(x) = c_n \tilde{\phi}_n(x)$ 。すなわち直交多項式系は定数倍を除いて一意に定まる。

よく用いられる直交多項式として、内積 (2.3) を

$$(p, q) = \int_{-1}^1 p(x)q(x)dx \quad (2.4)$$

(すなわち、重み関数 $w \equiv 1$ 、区間 $(a, b) = (-1, 1)$) とすると得られる直交多項式系 $p_n(x)$ は Legendre 多項式と呼ばれる。

2.2.2 Gauss 型積分公式

Newton-Cotes 積分公式は、 n 個の等間隔分点 x_i が与えられたときに、 n 個の重み w_i を適切に設定することで、少なくとも $n - 1$ 次までの多項式に対して誤差なく積分を行える積分公式と捉えることができる。そして、重み w_i だけでなく分点 x_i も適切に決めること作れる、より高次の多項式に対しても誤差の無い積分公式が Gauss 型積分公式である。ここではこのアイデアに基づき、Gauss 型積分公式を構成的に得る方法を示す。

n 個の重み w_i と分点 x_i , $(i = 1, \dots, n)$ に対して積分公式

$$\int_{-1}^1 f(x)dx = \sum_{i=1}^n w_i f(x_i) \quad (2.5)$$

が m 次までの多項式に対して正確であるとする (簡単のため積分範囲は $(-1, 1)$ とする)。このとき分点 x_i を零点とする n 次多項式を $G(x) = \prod_{i=1}^n (x - x_i)$ 、 r 次の Legendre 多項式を P_r とすると、 $0 \leq r \leq m - n$ に対して

$$\int_{-1}^1 G(x)P_r(x)dx = \sum_{i=1}^n w_i G(x_i)P_r(x_i) = 0 \quad (2.6)$$

が成り立つ ($\because G(x)P_r(x)$ は高々 m 次多項式であるから)。つまり $G(x)$ と $P_r(x)$ ($0 \leq r \leq m - n$) は直行する。これと Legendre 多項式の直交性より $G(x)$ は $m - n + 1$ 次以上の多項式でなければならない (\because もし $G(x)$ が $m - n$ 次以下の多項式ならば、 $G(x)$ は $P_r(x)$ ($0 \leq r \leq m - n$) の線形結合で表されるため、いずれかの $P_r(x)$ と直交しないから)。すなわち

$$\begin{aligned} n &\geq m - n + 1 \\ \Rightarrow m &\leq 2n - 1 \end{aligned} \quad (2.7)$$

積分公式 (2.5) は m 次までの多項式に対して正確であるとしたので、 m を最大化するために $m = 2n - 1$ とする。このとき n 次多項式 $G(x)$ に対して式 (2.6) は、 $0 \leq r \leq n - 1$ で成り立たなければならない。すなわち $G(x)$ は n 次の Legendre 多項式 (の定数倍) である必要がある。さらにこの事実と式 (2.6) より x_i は n 次 Legendre 多項式の零点となることがわかる。

数値積分公式の分点 x_i は n 次 Legendre 多項式の零点とすればよいことがわかったので、次に $m = 2n - 1$ 次以下の多項式とすると、 $f(x)$ を $P_n(x)$ で割った商を $Q(x)$ 、余りを $R(x)$ とする:

$$f(x) = P_n(x)Q(x) + R(x) \quad (2.8)$$

ただし $Q(x), R(x)$ は $n - 1$ 次以下の多項式である。 $P_n(x)$ は $n - 1$ 次以下の任意の多項式と直交するので、

$$\begin{aligned} \int_{-1}^1 f(x)dx &= \int_{-1}^1 (P_n(x)Q(x) + R(x))dx \\ &= \int_{-1}^1 R(x)dx \end{aligned} \quad (2.9)$$

となる。一方、分点 x_i は Legendre 多項式の零点であったから

$$\begin{aligned} \sum_{i=1}^n w_i f(x_i) &= \sum_{i=1}^n w_i (P_n(x_i)Q(x_i) + R(x_i)) \\ &= \sum_{i=1}^n w_i R(x_i) \end{aligned} \quad (2.10)$$

となる。以上より、数値積分公式が $n - 1$ 次以下の多項式 $R(x)$ に対して正確な値を与えるように重み w_i を定めれば、 $2n - 1$ 次以下の多項式 $f(x)$ に対しても正確な値を与えることがわかる。 $n - 1$ 次多項式 $R(x)$ の Lagrange 補間は $R(x)$ に等しいから、 $R(x) = \sum_{i=1}^n R(x_i)l_i(x)$ である。これを積分すると

$$\int_{-1}^1 R(x)dx = \sum_{i=1}^n R(x_i) \int_{-1}^1 l_i(x)dx \quad (2.11)$$

となるので、重みは

$$w_i = \int_{-1}^1 l_i(x)dx \quad (2.12)$$

で与えられる。 n 次 Legendre 多項式の零点 x_i と重み (2.12) による積分公式 (2.5) が n 次の Gauss 型積分公式である。

2.2.3 複合公式

Gauss 型積分公式も Newton-Cotes 公式と同様の理由で積分区間を小区間に分割し、それぞれの小区間に低次の積分公式を用いる複合公式を使うのが一般的である。一般に区間 (a, b) における積分を分点 $x_i = a + i(b - a)/n (i = 0, \dots, N)$ で N 等分すると

$$\int_a^b f(x)dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)dx \quad (2.13)$$

区間 (x_i, x_{i+1}) の積分は変数変換

$$y = 2 \frac{x - x_i}{x_{i+1} - x_i} - 1 \quad (2.14)$$

で $(-1, 1)$ の積分に変換できるので、 n 次 Gauss 型積分公式の積分点と重みをそれぞれ $y_m, w_m (m = 1, \dots, n)$ とすると、

$$\begin{aligned} \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx &= \sum_{i=0}^{N-1} \int_{-1}^1 f \left\{ \frac{(y+1)(x_{i+1}-x_i)}{2} + x_i \right\} \frac{x_{i+1}-x_i}{2} dy \\ &\approx \sum_{i=0}^{N-1} \sum_{m=1}^n w_m f \left\{ \frac{(y_m+1)(x_{i+1}-x_i)}{2} + x_i \right\} \frac{x_{i+1}-x_i}{2} \end{aligned} \quad (2.15)$$

となり、Gauss 型積分公式に対する複合公式を得る。

2.3 実験方法

(1)

(a) のコードを以下に示す。

コード 5: 2 次 Gauss 型積分公式

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5  #define M 2
6
7  double f(double x){
8      return 1/x;
9  }
10
11 double x(double i, double n){
12     double h=(b-a)/n;
13     return a+i*h;
14 }
15
16 double I(double n){
17     double I;
18     double u;
19
20     double y[2]={-1/sqrt(3),1/sqrt(3)};
21     double w[2]={1,1};
22
23     for(int i=0; i<n; i++){
24         for(int m=0; m<M; m++){
25             u=(y[m]+1)*(x(i+1,n)-x(i,n))/2+x(i,n);
26             I+=w[m]*f(u)*(x(i+1,n)-x(i,n))/2;
27         }
28     }
29
30     return I;
31 }
32
33 int main(void){
34     FILE *fp;
35     fp = fopen("a-2", "w");
36
37     double c=log(2);
38
39     for(int k=0; k<=10; k++){
40         double n=pow(2,k);
41         fprintf(fp,"% .16f_%.16f\n", n, fabs(I(n)-c));
42     }
43 }
```

コード 6: 3 次 Gauss 型積分公式

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 1.0
4  #define b 2.0
5  #define M 3
6
7  double f(double x){
8      return 1/x;
9  }
10
11 double x(double i, double n){
12     double h=(b-a)/n;
13     return a+i*h;
14 }
15
16 double I(double n){
17     double I;
18     double u;
19
20     double y[3]={-sqrt(3.0/5.0),0,sqrt(3.0/5.0)};
21     double w[3]={5.0/9.0,8.0/9.0,5.0/9.0};
22
23     for(int i=0; i<n; i++){
24         for(int m=0; m<M; m++){
25             u=(y[m]+1)*(x(i+1,n)-x(i,n))/2+x(i,n);
26             I+=w[m]*f(u)*(x(i+1,n)-x(i,n))/2;
27         }
28     }
29
30     return I;
31 }
32
33 int main(void){
34     FILE *fp;
35     fp = fopen("a-3", "w");
36
37     double c=log(2);
38
39     for(int k=0; k<=10; k++){
40         double n=pow(2,k);
41         fprintf(fp,"% .16f_%.16f\n", n, fabs(I(n)-c));
42     }
43 }

```

(2)

$M = 2$ の場合を示す。

コード 7: 方法 1

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 0.0
4  #define b 1.0
5  #define M 2
6
7  double f(double x){
8      return exp(-x)/sqrt(x);
9  }
10
11 double x(double i, double n){
12     double h=(b-a)/n;
13     return a+i*h;
14 }

```

```

15
16 double I(double n){
17     double I;
18     double u;
19
20     double y[2]={-1/sqrt(3),1/sqrt(3)};
21     double w[2]={1,1};
22
23     for(int i=0; i<n; i++){
24         for(int m=0; m<M; m++){
25             u=(y[m]+1)*(x(i+1,n)-x(i,n))/2+x(i,n);
26             I+=w[m]*f(u)*(x(i+1,n)-x(i,n))/2;
27         }
28     }
29
30     return I;
31 }
32
33 int main(void){
34     FILE *fp;
35     fp = fopen("a-1","w");
36
37     double c=1.49364826562;
38
39     for(int k=0; k<=10; k++){
40         double n=pow(2,k);
41         fprintf(fp,"%0.16f_%.16f\n", n, fabs(I(n)-c));
42     }
43 }

```

コード 8: 方法 2

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 0.0
4  #define b 1.0
5  #define M 2
6
7  double f(double x){
8      return 2*exp(-x*x);
9  }
10
11 double x(double i, double n){
12     double h=(b-a)/n;
13     return a+i*h;
14 }
15
16 double I(double n){
17     double I;
18     double u;
19
20     double y[2]={-1/sqrt(3),1/sqrt(3)};
21     double w[2]={1,1};
22
23     for(int i=0; i<n; i++){
24         for(int m=0; m<M; m++){
25             u=(y[m]+1)*(x(i+1,n)-x(i,n))/2+x(i,n);
26             I+=w[m]*f(u)*(x(i+1,n)-x(i,n))/2;
27         }
28     }
29
30     return I;
31 }
32
33 int main(void){
34     FILE *fp;
35     fp = fopen("a-2","w");
36

```

```

37     double c=1.49364826562;
38
39     for(int k=0; k<=10; k++){
40         double n=pow(2,k);
41         fprintf(fp,"%0.16f_%.16f\n", n, fabs(I(n)-c));
42     }
43 }

```

コード 9: 方法 3

```

1  #include<stdio.h>
2  #include<math.h>
3  #define a 0.0
4  #define b 1.0
5  #define M 2
6
7  double f(double x){
8      return (exp(-x)-1)/sqrt(x);
9  }
10
11 double x(double i, double n){
12     double h=(b-a)/n;
13     return a+i*h;
14 }
15
16 double I(double n){
17     double I;
18     double u;
19
20     double y[2]={-1/sqrt(3),1/sqrt(3)};
21     double w[2]={1,1};
22
23     for(int i=0; i<n; i++){
24         for(int m=0; m<M; m++){
25             u=(y[m]+1)*(x(i+1,n)-x(i,n))/2+x(i,n);
26             I+=w[m]*f(u)*(x(i+1,n)-x(i,n))/2;
27         }
28     }
29
30     double A=2;
31     I+=A;
32
33     return I;
34 }
35
36 int main(void){
37     FILE *fp;
38     fp = fopen("a-3","w");
39
40     double c=1.49364826562;
41
42     for(int k=0; k<=10; k++){
43         double n=pow(2,k);
44         fprintf(fp,"%0.16f_%.16f\n", n, fabs(I(n)-c));
45     }
46 }

```

2.4 結果

(1)

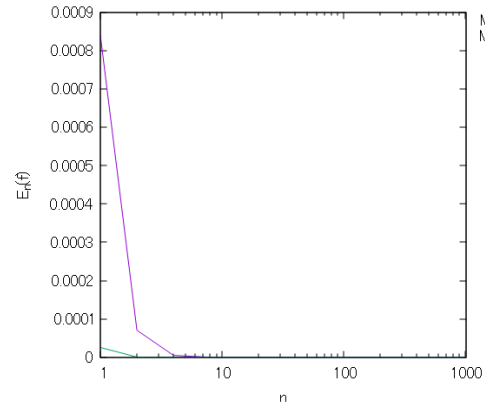


図 20: (a) 片対数

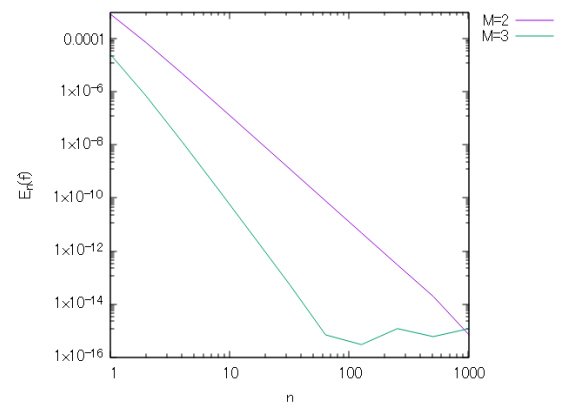


図 21: (a) 両対数

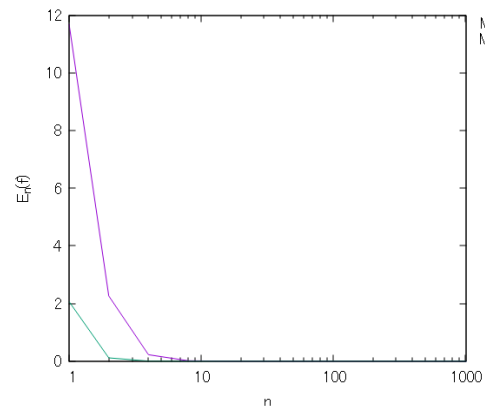


図 22: (b) 片対数

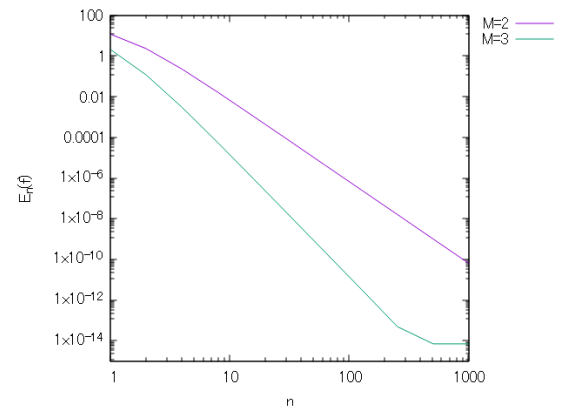


図 23: (b) 両対数

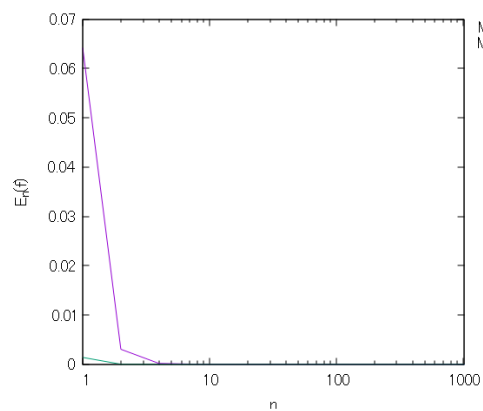


图 24: (c) 片对数

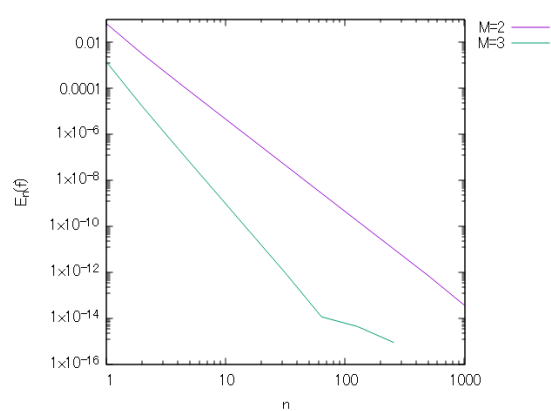


图 25: (c) 两对数

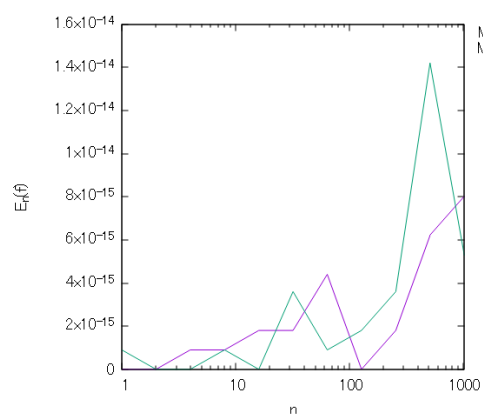


图 26: (d) 片对数

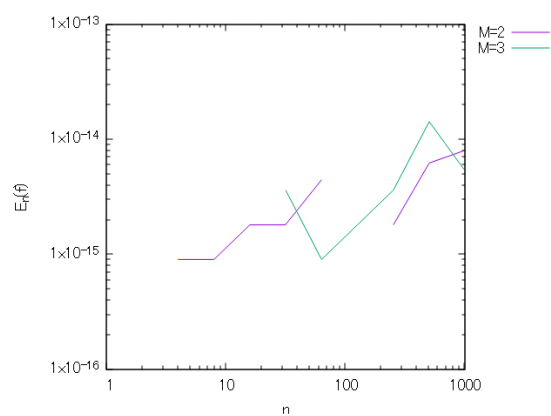


图 27: (d) 两对数

(2)

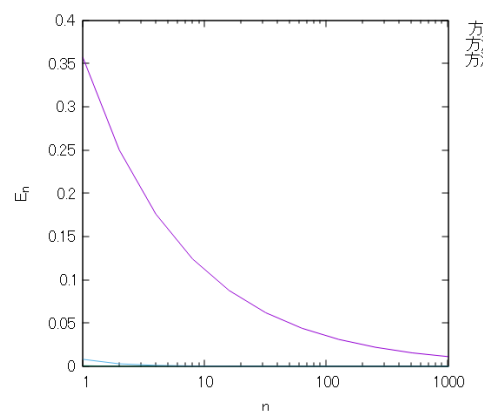


图 28: M=2 片对数

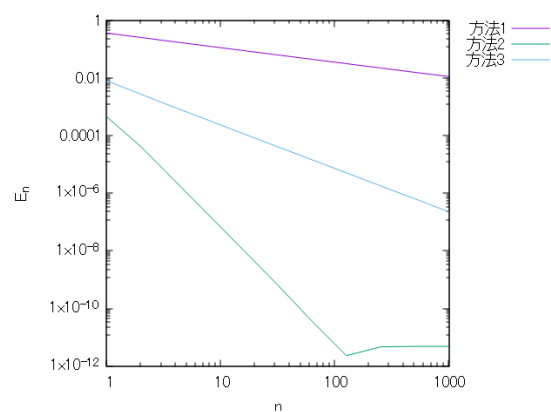


图 29: M=2 两对数

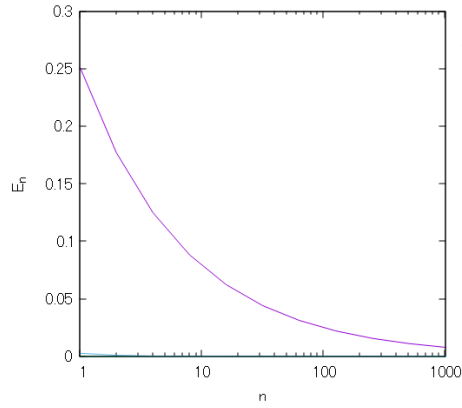


図 30: M=3 片対数

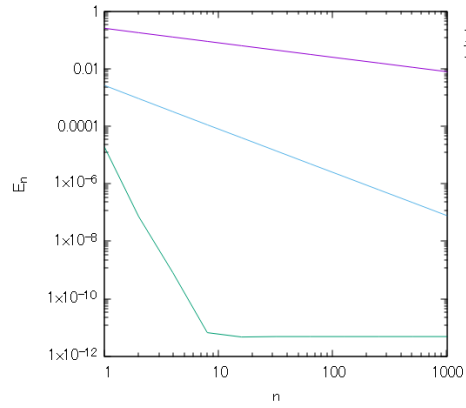


図 31: M=3 両対数

2.5 考察

(1)

(a)(b)(c)

区間 $(-1, 1)$ における積分の Gauss 型積分公式

$$\int_{-1}^1 f(x) dx = \sum_{i=1}^n w_i f(x_i) \quad (2.16)$$

の誤差はある $\xi \in (a, b)$ に対して

$$\int_{-1}^1 f(x) dx - \sum_{m=1}^n w_m f(y_m) \approx \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^1 \phi_n(x)^2 dx \quad (2.17)$$

$$\phi_n(x) = (x - y_1)^2 \cdots (x - y_m)^2 \quad (2.18)$$

となる。これを用いると複合 Gauss 型積分公式の誤差は

$$\begin{aligned} \int_a^b f(x) dx - \sum_{i=0}^{N-1} \sum_{m=1}^n w_m f(x(y_m)) \frac{x_{i+1} - x_i}{2} &= \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx - \sum_{i=0}^{N-1} \sum_{m=1}^n w_m f(x(y_m)) \frac{x_{i+1} - x_i}{2} \\ &= \sum_{i=0}^{N-1} \left\{ \int_{-1}^1 f(x(y)) dy - \sum_{m=1}^n w_m f(x(y_m)) \right\} \frac{x_{i+1} - x_i}{2} \\ &\approx \sum_{i=0}^{N-1} \frac{f^{(2n)}(\xi_i)}{(2n)!} \int_{-1}^1 \phi_n(x)^2 dx \left(\frac{x_{i+1} - x_i}{2} \right)^{2n+1} \\ &= N \frac{f^{(2n)}(\eta)}{(2n)!} \int_{-1}^1 \phi_n(x)^2 dx \left(\frac{x_{i+1} - x_i}{2} \right)^{2n+1} \end{aligned} \quad (2.19)$$

となる。したがって複合公式の刻み幅を $h = (b - a)/N = x_{i+1} - x_i$ とすると、この式は

$$\frac{(b - a) f^{(2n)}(\eta)}{(2n)! 2^{2n+1}} \int_{-1}^1 \phi_n(x)^2 dx h^{2n} \quad (2.20)$$

となり、誤差は h^{2n} 程度となると予測できる。

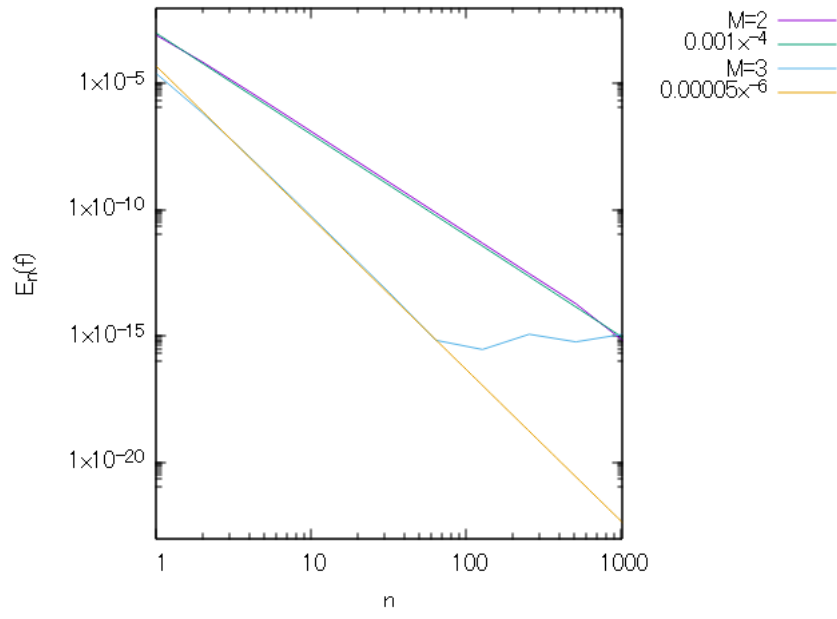


图 32: (a) $f_4(x) = 1/x \quad x \in [1, 2]$

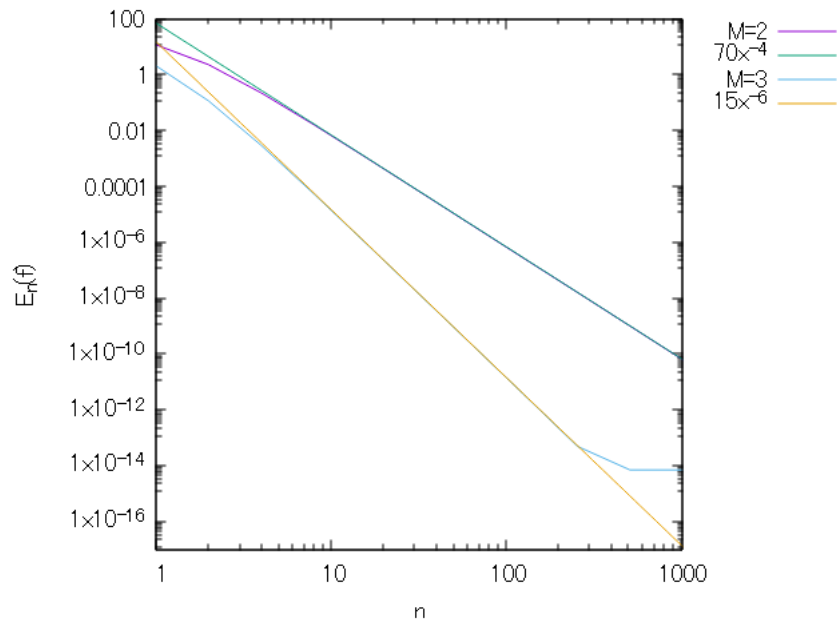


图 33: (b) $f_5(x) = e^{5x} \quad x \in [-1, 1]$

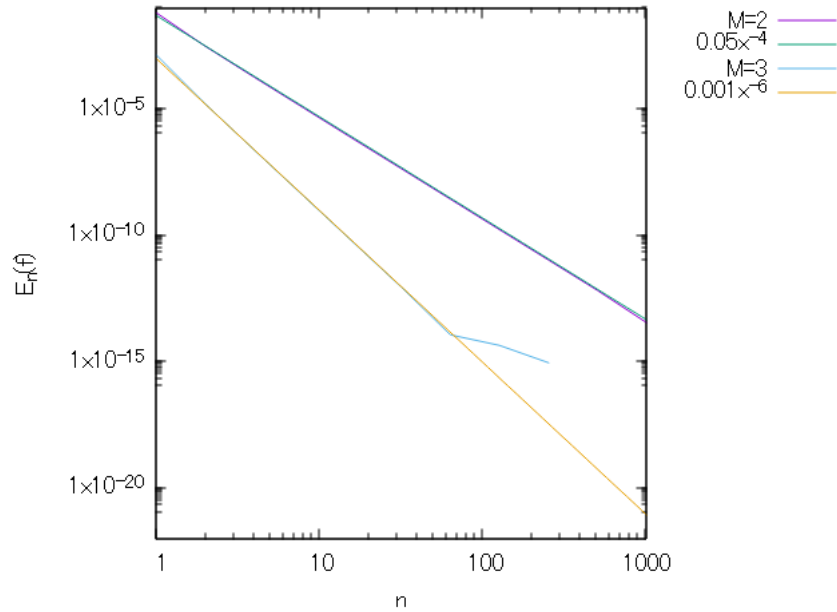


図 34: (c) $f_6(x) = 1 + \sin x \quad x \in [0, \pi]$

実際、2 次 Gauss 型積分公式の指数は $a = 4$ 、3 次 Gauss 型積分公式の指数は $a = 6$ である。
また、いずれも 3 次 Gauss 型積分公式の方が精度は高い。

(d) 課題 4.2 の 3 と同様のことが言える。

(2)

各方法の精度について議論する。 $f(x)$ が積分区間で発散するとき、きわめて大きな値が現れるので、大きな誤差が生じうると考えられる。

(方法 1) $x \rightarrow +0$ で $e^{-x}/\sqrt{x} \rightarrow \infty$
これより、精度は低いと予測できる。

(方法 2) 変数変換すると、

$$\begin{aligned} A &= \int_0^1 \frac{e^{-t^2}}{t} 2t dt \\ &= \int_0^1 2e^{-t^2} dt \end{aligned} \quad (2.21)$$

となる。 $x \rightarrow +0$ のとき、 $t \rightarrow +0$ であるため、 $2e^{-t^2} \rightarrow 1$
これより、精度は高い予測できる。

(方法 3) $x \rightarrow +0$ で $(e^{-x} - 1)/\sqrt{x} \rightarrow -0$ 収束はしているものの、極限を求める際に $0/0$ の不定形が現れている。そのため精度は、方法 1 よりは高いが、方法 2 よりは低いと考えられる。

実際、 $M = 2, 3$ のいずれにおいても精度は

$$\text{方法 1} < \text{方法 3} < \text{方法 2} \quad (2.22)$$

の順になっている。

また、台形公式、Simpson 公式についても $x = 0$ では関数が定義されていないので、数値計算は不可能である。

中点公式では、端点 ($x = 0$) における $f(x)$ は用いないので、数値計算は可能である。。

3 課題 4.4

3.1 はじめに

1 次元 Sturm-Liouville 型境界値問題

$$\frac{d}{dx} \left(e^{-x^2} \frac{d\tilde{u}}{dx}(x) \right) - 6e^{-x^2} \tilde{u}(x) = 0, \quad (0 < x < 1) \quad (3.1)$$

$$\tilde{u}(0) = 0, \quad \tilde{u}(1) = -4 \quad (3.2)$$

を考える。区間 $[0, 1]$ を $x_i (i = 0, \dots, n)$; $x_0 = 0, x_n = 1$ で n 等分し、有限要素法で解くコードを作成しよう。

(1)

$\tilde{u}(x) = 8x^3 - 12x$ が解であることを確かめよ。

(2)

$u = \tilde{u} + 4x$ と変数変換し、境界値問題 (1) を次の形に変形せよ。

$$-\frac{d}{dx} \left(p(x) \frac{du}{dx}(x) \right) + q(x)u(x) = f(x), \quad (0 < x < 1) \quad (3.3)$$

$$u(0) = u(1) = 0 \quad (3.4)$$

$p(x), q(x), f(x)$ を明記すること。

(3)

関数 $t_i(x) (i = 1, \dots, n-1)$ を

$$t_i(x) = \begin{cases} \frac{x-x_{i-1}}{h_i} & (x_{i-1} \leq x < x_i) \\ \frac{x_{i+1}-x}{h_{i+1}} & (x_i \leq x < x_{i+1}) \\ 0 & (\text{その他}) \end{cases} \quad h_i = x_i - x_{i-1} \quad (3.5)$$

で定義する。 $\frac{dt_i}{dx}(x)$ を求めよ。

(4)

関数がなすベクトル空間の部分空間 $\text{Span} t_1, \dots, t_{n-1}$ において境界値問題 (2) の弱解 u を求める。 u を t_i の線形結合で

$$u(x) = \sum_{i=1}^{n-1} c_i t_i(x) \quad (3.6)$$

と書く。 $c = (c_1, \dots, c_{n-1})^T$ (T は転置) を決定する方程式は

$$Ac = b \quad (3.7)$$

であり、サイズ $n-1$ の正方行列 $A = (A_{ij})$ とベクトル $b = b(b_i)$ は

$$A_{ij} = \int_0^1 \left[p(x) \frac{dt_i}{dx} \frac{dt_j}{dx} + q(x) t_i t_j \right] dx, \quad (i, j = 1, \dots, n-1) \quad (3.8)$$

$$b_i = \int_0^1 f(x) t_i(x) dx \quad (i = 1, \dots, n-1) \quad (3.9)$$

と定義される。行列 A は対称三重対角行列 (*) であることを確認せよ。行列 A とベクトル b の要素を求める際、積分区間を $[0,1]$ から縮小できることを議論せよ。

(*) 三重対角行列とは、対角成分とその上下に隣接する対角線上の成分以外は零なる行列

(5)

有限要素法のコードを作り理解を求めよ。 $M=3$ の Gauss 型積分公式を用いよ。得られた解を、分割数 n への依存性に注意しながら厳密解 $u = 8x(x^2 - 1)$ と比較・議論せよ。
ヒント: 方程式 $Ac = b$ は Gauss の消去表などで解けばよい。

3.2 原理・方法の詳細

3.2.1 有限要素法

有限要素法とは、変分問題に基づく数値解法であり、差分法と並んで代表的な偏微分方程式の数値解法である。

3.2.2 Sturm-Liouville 型微分方程式と弱形式

Sturm-Liouville 型の 2 点境界値問題

$$-\frac{d}{dx} \left(p(x) \frac{du}{dx}(x) \right) + q(x)u(x) = f(x), \quad a < x < b \quad (3.10)$$

$$u(a) = u(b) = 0 \quad (3.11)$$

を考える。ただし、 $p(x), q(x), f(x)$ は与えられた滑らかな関数とする。

$v(a) = v(b) = 0$ を満たす関数 v を式 (3.10) の両辺にかけ、区間 (a, b) で積分すると、左辺第一項は

$$\begin{aligned} \int_a^b v(x) \left\{ -\frac{d}{dx} \left(p(x) \frac{du}{dx}(x) \right) \right\} dx &= - \left[v(x) \left(p(x) \frac{du}{dx}(x) \right) \right]_a^b + \int_a^b \frac{dv}{dx}(x) p(x) \frac{du}{dx}(x) dx \\ &= \int_a^b p(x) \frac{dv}{dx}(x) \frac{du}{dx}(x) dx \end{aligned} \quad (3.12)$$

となり、以下の式を得る:

$$\begin{aligned} a(u, v) &:= \int_a^b \left(p(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) + q(x)u(x)v(x) \right) dx \\ &= \int_a^b f(x)v(x) dx \end{aligned} \quad (3.13)$$

式 (3.13) が弱形式である。有限要素法では問題 (3.10), (3.11) を解く代わりに、任意の関数 v に対して弱形式 (3.13) を満たす関数 u を求めることで解を求める。

3.2.3 弱形式の離散化

任意の関数 v に対して、弱形式 (3.13) を満たす関数 u を求めたいが、このままでは関数 u, v の取りうる自由度が無限にある。そこで離散化を行うことで関数 u, v が属する空間を有限次元空間に制限する。

例えば関数 $u(x)$ が n 次元ベクトル空間に属する場合、 u は基底関数 $t_i(x), (i = 1, \dots, n)$ と係数 $c_i \in R$ を用いて、

$$u(x) = \sum_{i=1}^n c_i t_i(x) \quad (3.14)$$

と書ける。基底 t_i の選び方は様々なものが考えられるが、ここでは折れ線関数による近似を考える。区間 (a, b) の分割 $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ に対して、 $h_i = x_i - x_{i-1} (i = 1, \dots, n)$ とし、 $t_i(x)$ を

$$t_i(x) = \begin{cases} \frac{x-x_{i-1}}{h_i} & (x_{i-1} \leq x \leq x_i) \\ \frac{x_{i+1}-x}{h_{i+1}} & (x_i \leq x \leq x_{i+1}) \\ 0 & (\text{その他}) \end{cases} \quad (3.15)$$

とすると、式 (3.14) で表される関数 $u(x)$ は点 $u(x_i) = c_i$ を満たす折れ線関数になる。

関数 $u(x)$ に式 (3.14) で表される折れ線関数を代入し、さらに $v = t_i$ を代入すると、式 (3.13) の左辺は

$$\begin{aligned} a \left(\sum_{j=1}^n u_j t_j, t_i \right) &= \int_a^b \left\{ p(x) \frac{d}{dx} \left(\sum_{j=1}^n u_j t_j(x) \right) \frac{dt_i}{dx}(x) + q(x) \left(\sum_{j=1}^n u_j t_j(x) \right) t_i(x) \right\} dx \\ &= \sum_{j=1}^n u_j \int_a^b \left(p(x) \frac{dt_j}{dx}(x) \frac{dt_i}{dx}(x) + q(x) t_i(x) t_j(x) \right) dx \end{aligned} \quad (3.16)$$

右辺は

$$\int_a^b f(x) t_i(x) dx \quad (i = 1, \dots, n) \quad (3.17)$$

となるので、これら n 本の式をまとめて行列表示で書き下すと

$$Au = b \quad (3.18)$$

となる。ここに n 次正方行列 A の ij 成分、および n 次ベクトル u, b の i 成分は、それぞれ

$$(A)_{ij} = \int_a^b \left(p \frac{dt_i}{dx} \frac{dt_j}{dx} + q t_i t_j \right) dx \quad (3.19)$$

$$(u)_i = u_i \quad (3.20)$$

$$(b)_i = \int_a^b f(x) t_i(x) dx \quad (3.21)$$

である。

3.2.4 Gauss の消去法

サイズ n の対称三重対角行列 A を

$$A = \begin{pmatrix} a_0 & b_0 & 0 & \cdots & 0 & 0 \\ b_0 & a_1 & b_1 & \ddots & 0 & 0 \\ a_1 & b_1 & a_2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & a_{n-2} & b_{n-2} \\ 0 & 0 & 0 & \cdots & b_{n-2} & a_{n-1} \end{pmatrix} \quad (3.22)$$

とする。このとき、 $Ax = d$ を解くアルゴリズムは以下の通りである。ただし $x = (x_0, \dots, x_{n-1})^T$, $d = (d_0, \dots, d_{n-1})^T$ である。

Step1

$$b'_0 = \frac{b_0}{a_0} \quad (3.23)$$

$$b'_i = \frac{b_i}{a_i - b_{i-1}b'_{i-1}} \quad (i = 1, \dots, n-2) \quad (3.24)$$

Step2

$$d'_0 = \frac{d_0}{a_0} \quad (3.25)$$

$$d'_i = \frac{d_i - b_{i-1}d'_{i-1}}{a_i - b_{i-1}b'_{i-1}} \quad (i = 1, \dots, n-2) \quad (3.26)$$

Step3

$$x_{n-1} = d'_{n-1} \quad (3.27)$$

$$x_i = d'_i - b_{i-1}b'_{i-1}x_{i-1} \quad (i = 1, \dots, n-2) \quad (3.28)$$

3.3 実験方法

$n = 7$ の場合のコードを以下に示す。

コード 10: 有限要素法

```
1  #include<stdio.h>
2  #include<math.h>
3  #define a 0.0
4  #define b 1.0
5  #define M 3
6  #define N 7
7
8  double X(double i, int n){
9      double h=(b-a)/n;
10     return a+i*h;
11 }
12
13 double t(double x, int i, int n){
14     if(x>=X(i-1,n) && x<X(i,n)){
```

```

15         return (x-X(i-1,n))/(X(i,n)-X(i-1,n));
16     }
17
18     else if(x>=X(i,n) && x<X(i+1,n)){
19         return (X(i+1,n)-x)/(X(i+1,n)-X(i,n));
20     }
21
22     else{
23         return 0;
24     }
25 }
26
27 double dt(double x,int i, int n){
28     if(x>=X(i-1,n) && x<X(i,n)){
29         return 1/(X(i,n)-X(i-1,n));
30     }
31
32     else if(x>=X(i,n) && x<X(i+1,n)){
33         return -1/(X(i+1,n)-X(i,n));
34     }
35
36     else{
37         return 0;
38     }
39 }
40
41 double dA(double x, int i, int j, int n){
42     return exp(-x*x)*dt(x,i,n)*dt(x,j,n)-6*exp(-x*x)*t(x,i,n)*t(x,j,n);
43 }
44
45 double dB(double x, int i, int n){
46     return -16*x*exp(-x*x)*t(x,i,n);
47 }
48
49 int main(void){
50     double y[3]={-sqrt(3.0/5.0),0,sqrt(3.0/5.0)};
51     double w[3]={5.0/9.0,8.0/9.0,5.0/9.0};
52     double u;
53     int n=pow(2,N);
54
55     double A[n-1][n-1];
56
57     for(int i=1; i<n; i++){
58         for(int j=1; j<n; j++){
59             A[i-1][j-1]=0;
60             for(int k=0; k<n; k++){
61                 for(int m=0; m<M; m++){
62                     u=(y[m]+1)*(X(k+1,n)-X(k,n))/2+X(k,n);
63                     A[i-1][j-1]+=w[m]*dA(u,i,j,n)*(X(k+1,n)-X(k,n))/2;
64                 }
65             }
66         }
67     }
68
69     double B[n-1];
70
71     for(int i=1; i<n; i++){
72         B[i-1]=0;
73         for(int k=0; k<n; k++){
74             for(int m=0; m<M; m++){
75                 u=(y[m]+1)*(X(k+1,n)-X(k,n))/2+X(k,n);
76                 B[i-1]+=w[m]*dB(u,i,n)*(X(k+1,n)-X(k,n))/2;
77             }
78         }
79     }
80
81     double B0[n-2];
82

```

```

83     B0[0]=A[0][1]/A[0][0];
84     for(int i=1; i<=n-3; i++){
85         B0[i]=A[i][i+1]/(A[i][i]-A[i-1][i]*B0[i-1]);
86     }
87
88     double D0[n-1];
89
90     D0[0]=B[0]/A[0][0];
91     for(int i=1; i<=n-2; i++){
92         D0[i]=(B[i]-A[i-1][i]*D0[i-1])/(A[i][i]-A[i-1][i]*B0[i-1]);
93     }
94
95     double C[n+1];
96
97     C[n]=0;
98     for(int i=n-1; i>=0; i--){
99         C[i]=D0[i-1]-B0[i-1]*C[i+1];
100    }
101
102    FILE *fp;
103    fp = fopen("7","w");
104
105    for(int i=0; i<=n; i++)
106    {
107        fprintf(fp,"% .16f %.16f\n", X(i,n), C[i]);
108    }
109
110    fclose(fp);
111 }

```

3.4 結果

(1)

$\tilde{u}(x) = 8x^3 - 12x$ のとき、境界値問題 (3.1) は

$$\begin{aligned}
 (\text{左辺}) &= -\frac{d}{dx} \left(e^{-x^2} \frac{d\tilde{u}}{dx}(x) \right) - 6e^{-x^2} \tilde{u}(x) \\
 &= -\frac{d}{dx} \left(e^{-x^2} \frac{d}{dx} (8x^3 - 12x) \right) - 6e^{-x^2} (8x^3 - 12x) \\
 &= -\frac{d}{dx} \left(e^{-x^2} (24x^2 - 12) \right) - 6e^{-x^2} (8x^3 - 12x) \\
 &= -\left\{ -2xe^{-x^2} (24x^2 - 12) + e^{-x^2} (48x) \right\} - 6e^{-x^2} (8x^3 - 12x) \\
 &= e^{-x^2} (2x(24x^2 - 12) - 48x - 6(8x^3 - 12x)) \\
 &= 0 \\
 &= (\text{右辺})
 \end{aligned} \tag{3.29}$$

となる。また、

$$\tilde{u}(0) = 0 \tag{3.30}$$

$$\tilde{u}(1) = -4 \tag{3.31}$$

となるため、 $\tilde{u}(x) = 8x^3 - 12x$ は境界値問題 (3.1) の解である。

(2)

$u = \tilde{u} + 4x$ のとき、 $\tilde{u} = u - 4x$ となるため、境界値問題 (3.1) は

$$\begin{aligned}
(\text{左辺}) &= -\frac{d}{dx} \left(e^{-x^2} \frac{d\tilde{u}}{dx}(x) \right) - 6e^{-x^2} \tilde{u}(x) \\
&= -\frac{d}{dx} \left(e^{-x^2} \frac{d}{dx} (u(x) - 4x) \right) - 6e^{-x^2} (u(x) - 4x) \\
&= -\frac{d}{dx} \left(e^{-x^2} \left(\frac{du}{dx}(x) - 4 \right) \right) - 6e^{-x^2} (u(x) - 4x) \\
&= -\frac{d}{dx} \left(e^{-x^2} \frac{du}{dx}(x) \right) + 8xe^{-x^2} - 6e^{-x^2} (u(x) - 4x) \\
&= -\frac{d}{dx} \left(e^{-x^2} \frac{du}{dx}(x) \right) - 6e^{-x^2} u(x) + 16xe^{-x^2}
\end{aligned} \tag{3.32}$$

となるため、

$$-\frac{d}{dx} \left(e^{-x^2} \frac{du}{dx}(x) \right) - 6e^{-x^2} u(x) = -16xe^{-x^2}, \quad (0 < x < 1) \tag{3.33}$$

$$u(0) = u(1) = 0 \tag{3.34}$$

となる。したがって、

$$\begin{cases} p(x) = e^{-x^2} \\ q(x) = -6e^{-x^2} \\ f(x) = -16xe^{-x^2} \end{cases} \tag{3.35}$$

(3)

$x_{i-1} \leq x < x_i$ のとき

$$\frac{dt_i}{dx}(x) = \frac{1}{h_i} \tag{3.36}$$

$x_i \leq x < x_{i+1}$ のとき

$$\frac{dt_i}{dx}(x) = -\frac{1}{h_{i+1}} \tag{3.37}$$

したがって

$$\frac{dt_i}{dx}(x) = \begin{cases} \frac{1}{h_i} & (x_{i-1} \leq x < x_i) \\ -\frac{1}{h_{i+1}} & (x_i \leq x < x_{i+1}) \\ 0 & (\text{その他}) \end{cases} \quad h_i = x_i - x_{i-1} \tag{3.38}$$

(4)

行列 A は

$$\begin{aligned}
A_{ji} &= \int_0^1 \left[p(x) \frac{dt_j}{dx} \frac{dt_i}{dx} + q(xt_j t_i) \right] dx \\
&= \int_0^1 \left[p(x) \frac{dt_i}{dx} \frac{dt_j}{dx} + q(xt_i t_j) \right] dx \\
&= A_{ij}
\end{aligned} \tag{3.39}$$

となるため、対称行列である。

また、 $|i - j| \geq 2$ のとき

$$t_i(x) = \frac{dt_i}{dx} = 0 \quad \text{または} \quad t_j(x) = \frac{dt_j}{dx} = 0 \quad (3.40)$$

となるため、

$$A_{ij} = 0 \quad (3.41)$$

となる。したがって、行列 A は対称三重対角行列である。

(5)

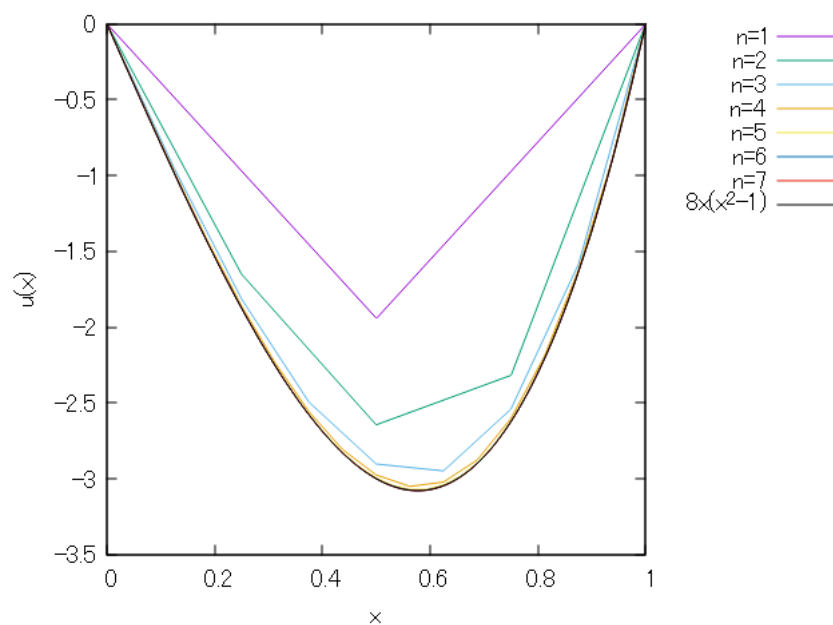


図 35: 有限要素法

3.5 考察

分割数 n が大きくなるにつれて、厳密解との誤差は小さくなっている。