

第8章 演習課題

課題 PDF ダウンロード

課題 1

サンプルプログラムをコンパイル・実行して動作を確認せよ。さらに、適宜修正してその実行結果を確認せよ。

課題 2

2 次方程式 $ax^2 + bx + c = 0$ の解は以下の解の公式を用いて求めることが出来る。

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

しかしこの公式を単純に用いた場合には $b^2 \gg 4ac$ の時には、桁落ちによってどちらか一方の解の精度が悪くなってしまう。このことを実際に確認し、式変形によってその精度を改善せよ。

例えば $a = 1, b = -10^9, c = 1$ とした時の解は

$$x \simeq \underbrace{1.000000000000000}_{0 \text{ が } 15 \text{ 個}} \times 10^9, \quad \underbrace{1.000000000000000}_{0 \text{ が } 15 \text{ 個}} \times 10^{-9}$$

であるが、解の公式を用いた場合と式変形によって桁落ち対策をした場合で数値解を比較せよ。(桁落ち対策が必要なのはどちらか一方の解のみである。)

課題 3

Newton 法では関数の微分値を解析的に与えて用いるが、割線法 (Secant method) と呼ばれる手法では連続した 2 つの近似解を用いて、微分を差分で近似する。即ち n 番目の近似解を x_n と書いたときに、Newton 法の公式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

において、微分値を以下のように近似する。

$$f'(x_n) \simeq \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

(この手法では微分値の計算をする必要がないため、反復 1 回あたりの計算量が少なくなる可能性があるが、収束は Newton 法よりも少し遅くなる。)

この割線法を実装し、以下の方程式

$$f(x) = \frac{1-x}{\sqrt{M}} - \exp(x)$$

を数値的に解くことで、その収束の速さを二分法および Newton 法と比較せよ。ただし $M = 1836$ とする。なお解は $x \simeq -2.5$ なので二分法ではこの前後に 2 つの初期値を指定すればよい。割線法も 2 つ初期値が必要になるが、(この関数に関しては) 初期値には敏感ではないので、例えば 0 と 1 を初期値として用いればよい。ここでは収束の速さを比較すればよいので、各反復ごとの収束判定はせずに 20 回程度の反復を行い誤差の減少の様子を gnuplot を用いて図示せよ。ただし連続する 2 つの近似解の差を誤差と定義する。

結果は以下の例ようになるであろう。

課題 4

$f(x) = \frac{4}{\pi} \frac{1}{1+x^2}$ および $f(x) = (n+1)x^n$ ($n = 0, \dots, 5$) のそれぞれについて

$$\int_0^1 f(x) dx$$

の積分を台形公式および Simpson の公式で数値的に行い、誤差の分割数に対する依存性を gnuplot を用いて図示せよ。例えば分割数を 2^n ($n = 1, 2, \dots, 16$) の範囲で変えて依存性を調べれば良い。

結果は以下の例ようになるであろう。

このような依存性となるのはなぜか？

課題 5

一様乱数を用いてある確率分布に従う乱数を発生させるために逆関数を用いる方法 (変換法や逆関数法などと呼ばれる) を考える。区間 $[0, 1)$ での一様乱数 x およびその確率分布 $f(x)$ 、また必要な乱数 y とその確率分布 $g(y)$ とする。ただし乱数の値域は $a \leq y < b$ とする。 y を x から何らかの変換で $y = P(x)$ のように表すとき、確率密度の保存から

$$g(y)dy = f(x)dx = dx$$

が成り立つ。(最後の等式は一様乱数であることから自明である。) この両辺を積分して

$$x = \frac{\int_a^y g(y') dy'}{\int_a^b g(y') dy'} \equiv G(y)$$

によって関数 $G(y)$ を定義する。ただし $G(a) = 0$ 、 $G(b) = 1$ となるように規格化した。これより

$$y = G^{-1}(x)$$

を得る。即ち、 $G^{-1}(x)$ を解析的に求めることができれば、一様乱数 x を用いて必要な確率分布 $g(y)$ に従う乱数を作ることができる。(以下の図を参照)

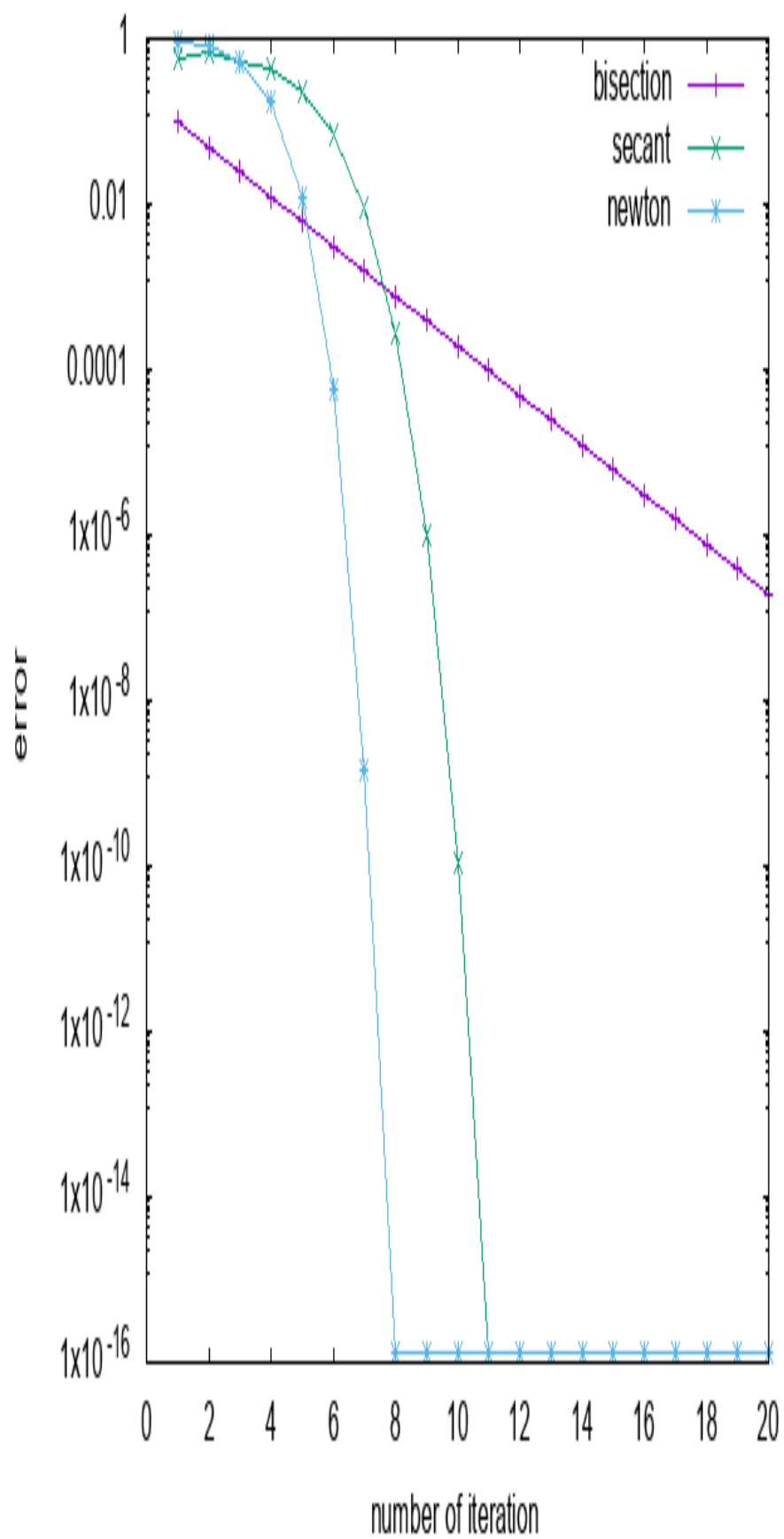


図 1: 求根法の収束比較

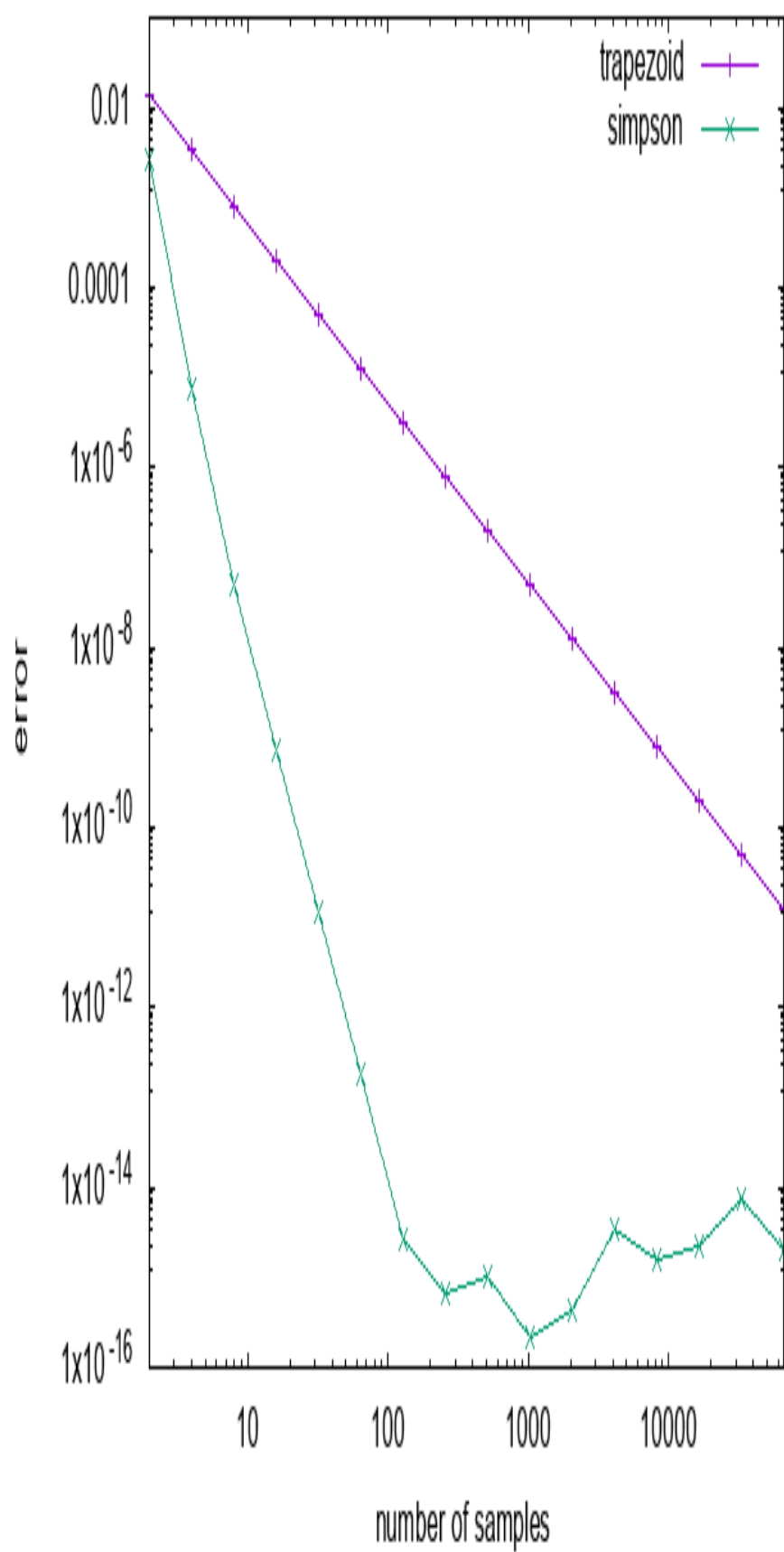


図 2: 数値積分誤差の分割数依存性

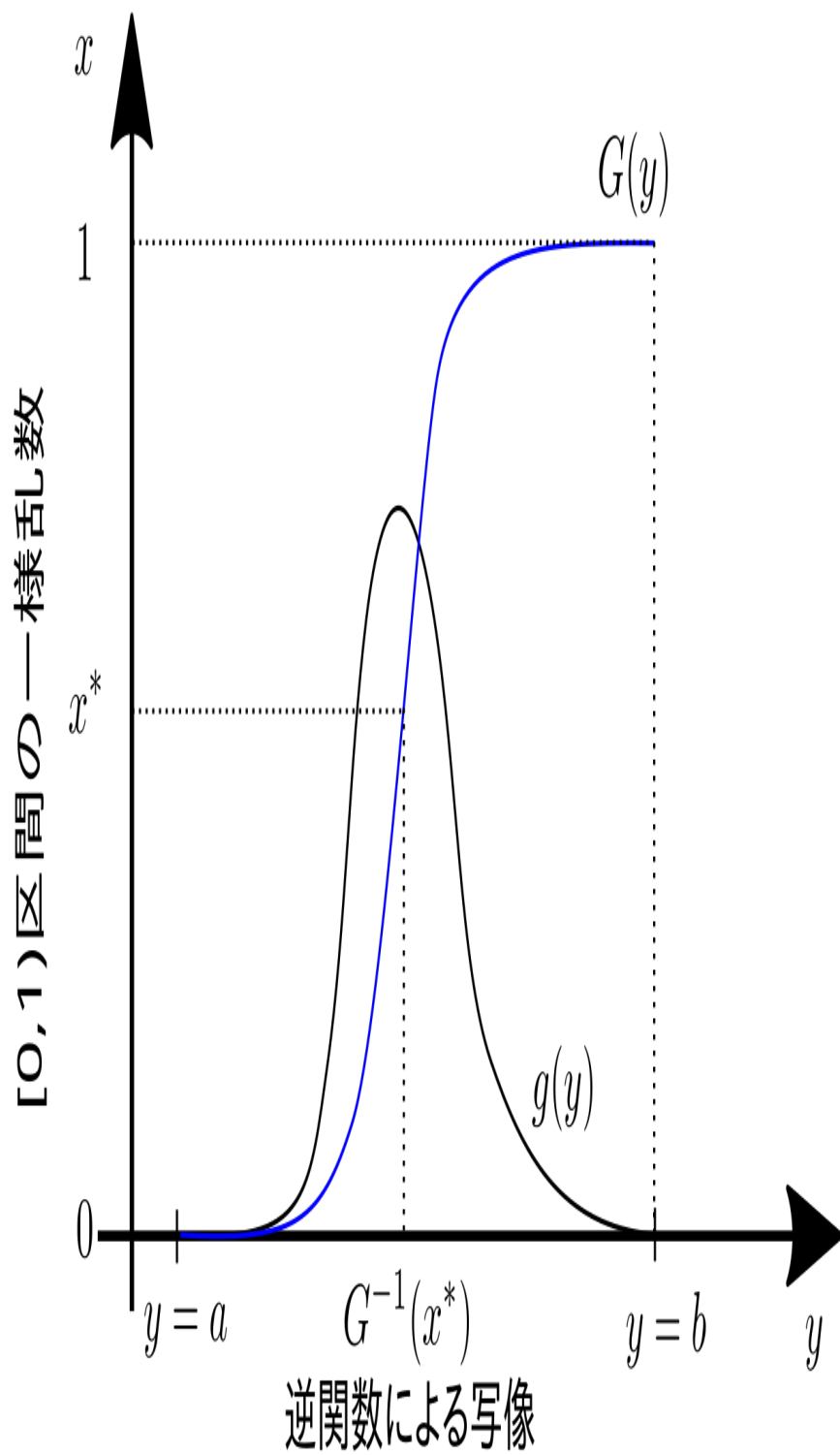


図 3: 逆関数法の概念図

このことを用いて指数分布

$$g(y; \lambda) = \lambda \exp(-\lambda y)$$

に従う乱数分布を発生させるプログラムを作成せよ。また分布のヒストグラムを作成し、gnuplot を用いてヒストグラムを解析的な分布と共に図示し、乱数の発生数を増やした時に乱数分布が真の分布に近づくことを確かめよ。

以下は 60000 個の乱数を発生させた場合のヒストグラムの例である。

課題 6 [†]

乱数を利用した数値積分法としてモンテカルロ法が知られている。これを用いて n 次元ユークリッド空間における単位超球の体積 V_n を求めるプログラムを作成せよ。

ただし n 次元超球の体積は n 次元空間の座標を x_i ($i = 1, \dots, n$) とし、

$$\sum_{i=1}^n x_i^2 \leq 1$$

なる領域の体積と定義される。ここで対称性から $0 \leq x_i \leq 1$ の領域のみを考えれば、この体積は $V_n/2^n$ となる。従って n 個の一樣乱数 $0 \leq x_i < 1$ ($i = 1, \dots, n$) を発生させ、上式の条件を満足するかどうかを調べる試行を多数回行い、その確率を求めることによって $V_n/2^n$ を推定すればよい。

なお真の値は

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)}$$

によって与えられる。

例えば次元数と試行回数を標準入力から与える形式のプログラムの実行結果は以下のようになる。この例のように誤差は試行回数 m に対して、 $1/\sqrt{m}$ に比例して減少する。

```
$ ./a.out
2 1000    # キーボード入力 (2 次元, 試行回数 1000 回)
approximation =      0.32240000E+01
exact value    =      0.31415927E+01
relative error =      0.26231044E-01
$ ./a.out
2 100000  # キーボード入力 (2 次元, 試行回数 100000 回)
approximation =      0.31430400E+01
exact value    =      0.31415927E+01
relative error =      0.46067683E-03
```

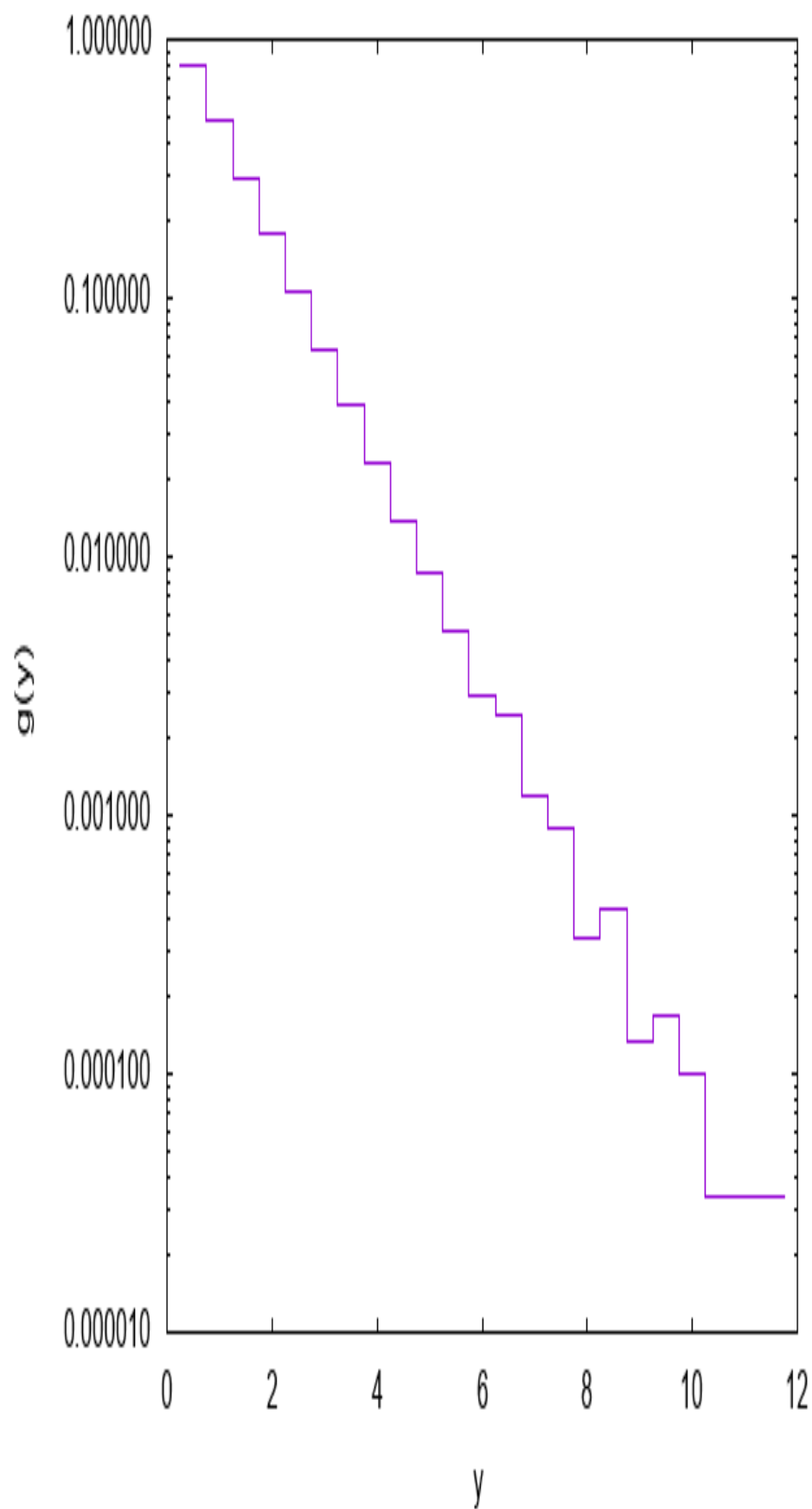


図 4: 指数分布のヒストグラム