「モデリングとシミュレーション特論」課題（解答例)

2019/5/21

1　乱数と MonteCarlo 法

課題 1　Consider a probability density defined by Eq. (1.1) in $[0, 1)$ (Fig. 1). Generate random numbers obeying Eq. (1.1) by transformation method. And show the histogram for generated random numbers. The reference is `randomNumbers/Exp.java`.

$$f(x) = \begin{cases} 4x & 0 \le x < 1/2 \\ 4 - 4x & 1/2 \le x < 1 \end{cases} \tag{1.1}$$

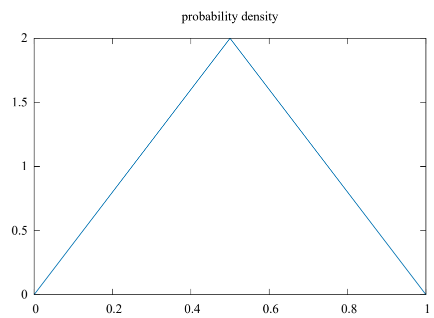


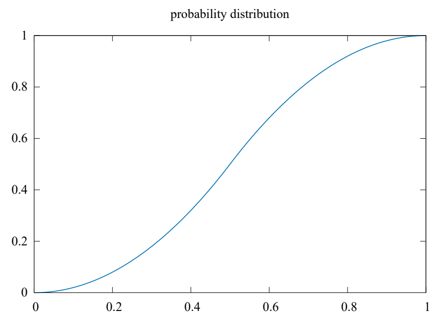
図 1　probability density

解答例　For applying transformation method to generate random numbers, we need to have the reverse of the probability distribution. For $0 \le x < 1/2$, the probability distribution $F(x)$ is

$$F(x) = \int_0^x f(y)\mathrm{d}y = \int_0^x 4y\mathrm{d}y = 2x^2,$$

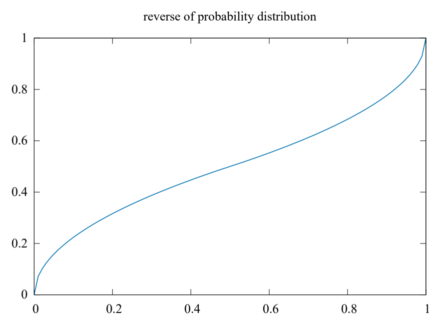
and for $1/2 \le x < 1$,

$$F(x) = \frac{1}{2} + \int_{1/2}^{x} f(y)\mathrm{d}y = \frac{1}{2} + \int_{1/2}^{x} (4 - 4y)\mathrm{d}y$$
$$= \frac{1}{2} + \left[4y - 2y^2\right]_{1/2}^{x} = -2x^2 + 4x - 1.$$



The reverse function of the distribution $F(x)$ is

$$F^{-1}(x) = \begin{cases} \left(\dfrac{x}{2}\right)^{1/2} & 0 \le x < 1/2, \\ 1 - \left(\dfrac{1-x}{2}\right)^{1/2} & \text{otherwise.} \end{cases}$$



By reference to the sample program `randomNumbers/Exp.java`, we can implement the random number generator as in Source Code 1.

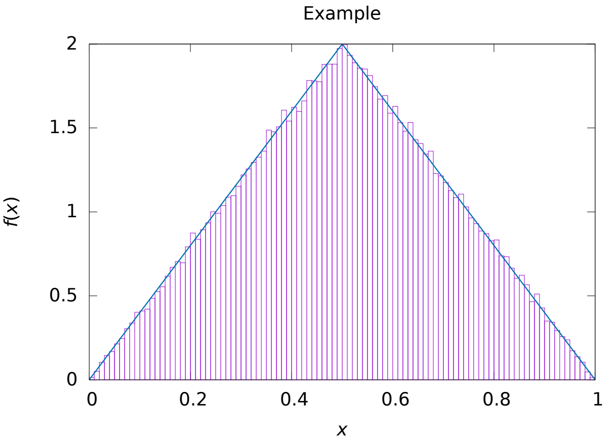
Source Code 1 ExampleRandom.java

```
package exercise;

import histogram.Histogram;
import java.awt.geom.Point2D;
import java.io.BufferedWriter;
import java.io.IOException;
```

```
import java.util.List;
import java.util.function.DoubleFunction;
import myLib.utils.FileIO;
import randomNumbers.AbstractRandom;
import randomNumbers.Exp;
import randomNumbers.Transform;

/**
 *
 * @author tadaki
 */
public class ExampleRandom {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {
        DoubleFunction<Double> invProDist = x -> {
            if (x < .5) {
                return Math.sqrt(x / 2.);
            }
            return 1. - Math.sqrt((1.-x)/2.);
        };

        //変換法による乱数生成のインスタンス
        AbstractRandom aRandom = new Transform(invProDist);

        double min = 0.;//下限
        double max = 1.;//上限
        int numBin = 100;//binの数
        int numSamples = 100000;//乱数の総数
        //ヒストグラムを生成
        Histogram histogram = new Histogram(min, max, numBin);
        for (int i = 0; i < numSamples; i++) {
            double x = aRandom.getNext();
            histogram.put(x);
        }
        //ヒストグラムを出力
        List<Point2D.Double> plist = histogram.calculateFrequency();
        String filename = ExampleRandom.class.getSimpleName() + "-output.txt";
        try (BufferedWriter out = FileIO.openWriter(filename)) {
            for (Point2D.Double p : plist) {
                FileIO.writeSSV(out, p.x, p.y);
            }
        }
    }
```

```
}
```

We obtain the histgram for this random numbers. The result shows that Source code 1 generates random numbers defined by Eq. (1.1).



課題 2 Derive the average μ and variance σ^2 for the probability density (Eq. 1.1) analytically.

解答例

$$
\begin{aligned}
\mu = \langle x \rangle &= \int_0^1 x f(x) \mathrm{d}x = \int_0^{1/2} 4x^2 \mathrm{d}x + \int_{1/2}^1 x\left(4 - 4x\right)\mathrm{d}x \\
&= \left[\frac{4}{3}x^3\right]_0^{1/2} + \left[2x^2 - \frac{4}{3}x^3\right]_{1/2}^1 = \frac{1}{2}, \\
\left\langle x^2 \right\rangle &= \int_0^1 x^2 f(x) \mathrm{d}x = \int_0^{1/2} 4x^3 \mathrm{d}x + \int_{1/2}^1 4x^2\left(1 - x\right)\mathrm{d}x \\
&= \left[x^4\right]_0^{1/2} + \left[\frac{3}{4}x^4 - x^4\right]_{1/2}^1 = \frac{5}{24}, \\
\sigma^2 &= \left\langle x^2 \right\rangle - \mu^2 = \frac{1}{24}.
\end{aligned}
$$

課題 3 At the lecture, we observe the law of large numbers for uniform random numbers. Observe the law for the case defined by Eq. (1.1). The reference is `LawOfLargeNumbers/LargeNumbers.java`.

解答例 By the reference to the sample program `LawOfLargeNumbers/LargeNumbers.java`, we construct the simulation program as Source Code 2

Source Code 2 LargeNumbersExercise.java

```java
package exercise;

import LawOfLargeNumbers.*;
import randomNumbers.AbstractRandom;
import java.io.BufferedWriter;
import java.io.IOException;
import java.util.List;
import java.util.function.DoubleFunction;
import myLib.utils.FileIO;
import randomNumbers.Transform;

/**
 * 大数の法則を確認する。
 *
 * @author tadaki
 */
public class LargeNumbersExercise {

    public static void main(String args[]) throws IOException {
        int nSamples = 1000;//各サイズの標本数
        int num = 16;//標本サイズ初期値
        int numMax = 16384;//標本サイズ最大値
        DoubleFunction<Double> invProDist = x -> {
            if (x < .5) {
                return Math.sqrt(x / 2.);
            }
            return 1. - Math.sqrt((1.-x)/2.);
        };

        //変換法による乱数生成のインスタンス
        AbstractRandom myRandom = new Transform(invProDist);

        LargeNumbers ln = new LargeNumbers(myRandom);
        List<Result> plist = ln.observeSizeDependence(num, numMax, nSamples);

        //結果出力
        String filename = LargeNumbersExercise.class.getSimpleName() + ".txt";
        try (BufferedWriter out = FileIO.openWriter(filename)) {
            for (Result p : plist) {
                out.write(p.toString());
                out.newLine();
            }
        }
```

```
    }
}
```

We find that the averages of samples are that of the population. And they converges to the population mean as $\sigma n^{-1/2}$.