確率と統計（0）
「計算機による乱数の発生」

- **担当教員：　杉山　将（計算工学専攻）**
- **居室：**　W8E-505
- **電子メール:**　sugi@cs.titech.ac.jp

- **授業のウェブサイト：**

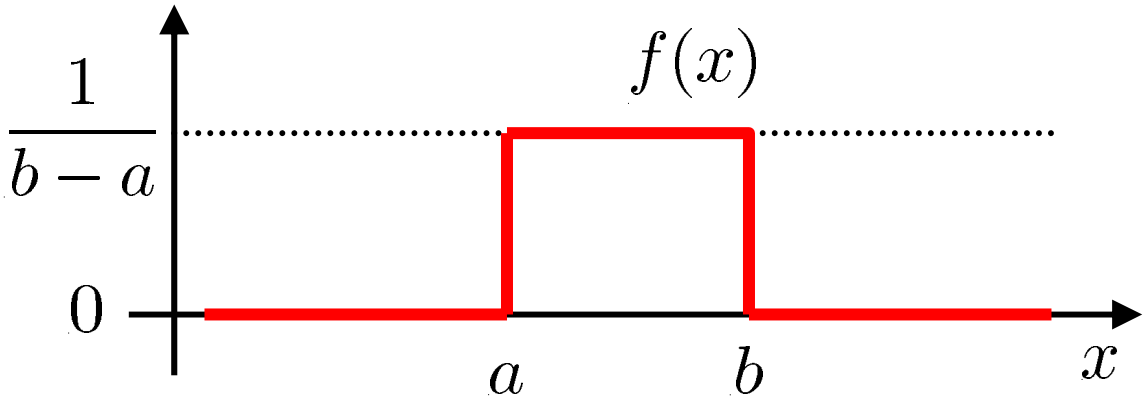
http://sugiyama-www.cs.titech.ac.jp/~sugi/

一様分布

- **連続一様分布**(uniform distribution of continuous type)：**確率密度関数が一様**

$$f(x) = \begin{cases} \frac{1}{b-a} & (\text{ for } a \leq x \leq b) \\ 0 & (otherwise) \end{cases}$$

- (a, b) **上の一様分布を** $U(a, b)$ **と表す．**



計算機による乱数の生成

- 乱数を計算機内で生成するのは非常に難しい！
- 乱数を生成するための専用のハードウェアもある
  （例）電子素子の熱雑音などの物理現象を利用
- 一般には，擬似乱数を用いることが多い
  （例）C言語のrand関数は一様擬似乱数を生成する
- 一様擬似乱数や正規擬似乱数を生成する関数は，大抵の計算機言語で用意されている
- それ以外の任意の分布に従う乱数も作りたい！

乱数の生成法1：逆関数法

- $F(x)$ ：乱数を生成したい分布の累積分布関数

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du$$

- $\{u_i\}_{i=1}^{n}$：$(0,1)$ 上の一様分布に従う n 個の乱数

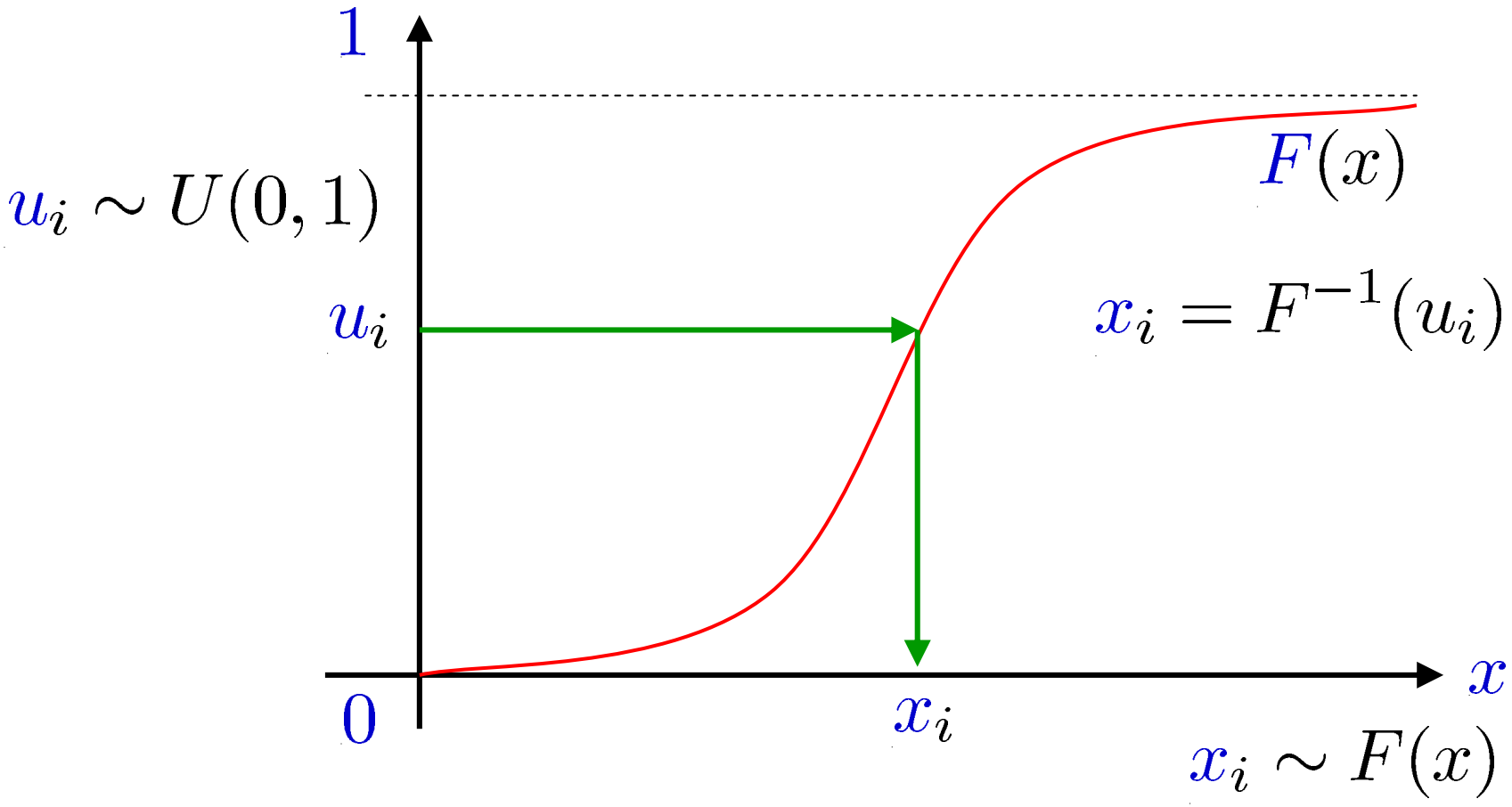
$$u_i \sim U(0,1)$$

- $F^{-1}(u)$：$F(x)$ の逆関数

$$u = F(x)$$
$$x = F^{-1}(u)$$

- $x_i = F^{-1}(u_i)$ と変換する
- そうすると，$\{x_i\}_{i=1}^{n}$ は $F(x)$ に従う

逆関数法 (続き)



逆関数法 (証明)

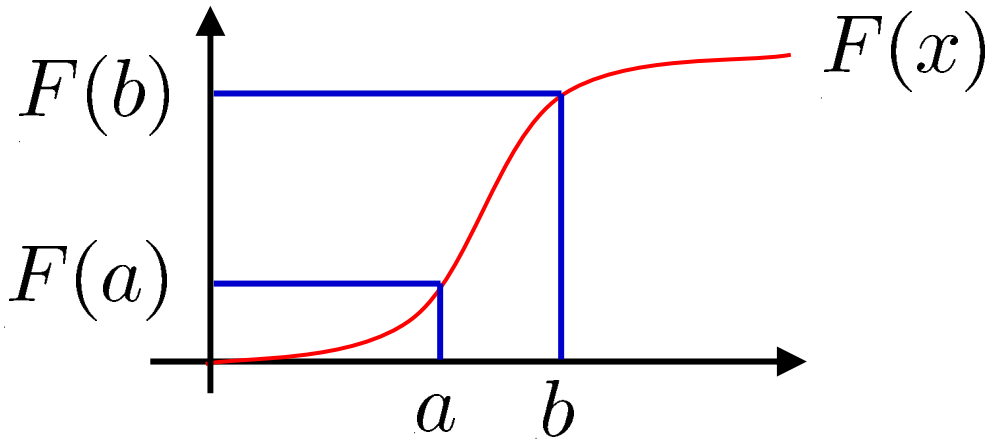
- $P(x_i \leq c) = F(c), \forall c$ **を示す**

$$
\begin{aligned}
P(x_i \leq c) &= P(x_i < c) \\
&= P(F^{-1}(u_i) < c) \\
&= P(u_i \leq F(c)) \\
&= P(0 \leq u_i \leq F(c)) \\
&= F(c)
\end{aligned}
$$

$x_i = F^{-1}(u_i)$

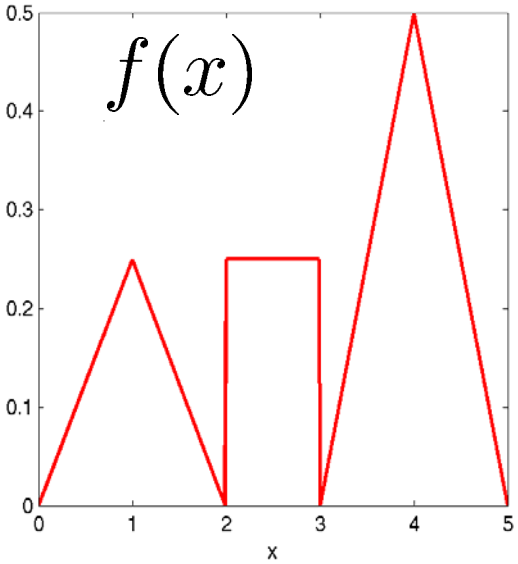
$a < b \Rightarrow F(a) \leq F(b)$

$u_i \sim U(0,1)$

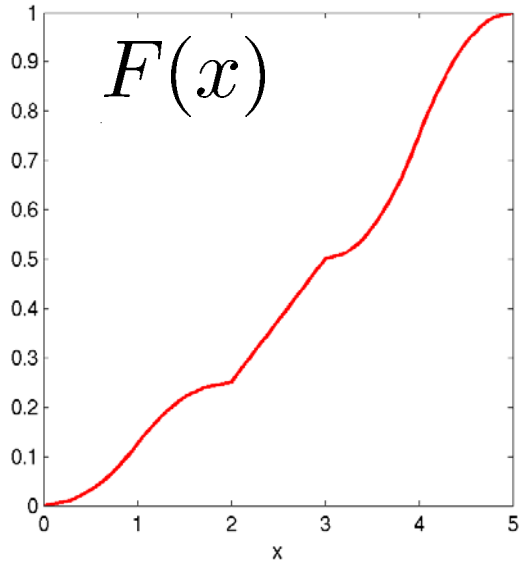


逆関数法による乱数生成の例

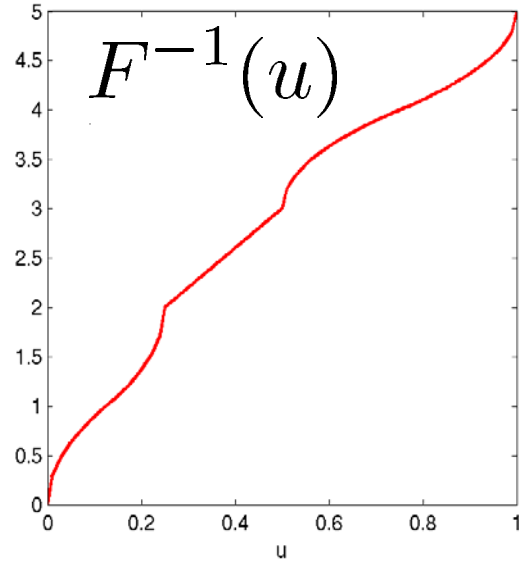
$$x = F^{-1}(u)$$



確率密度関数

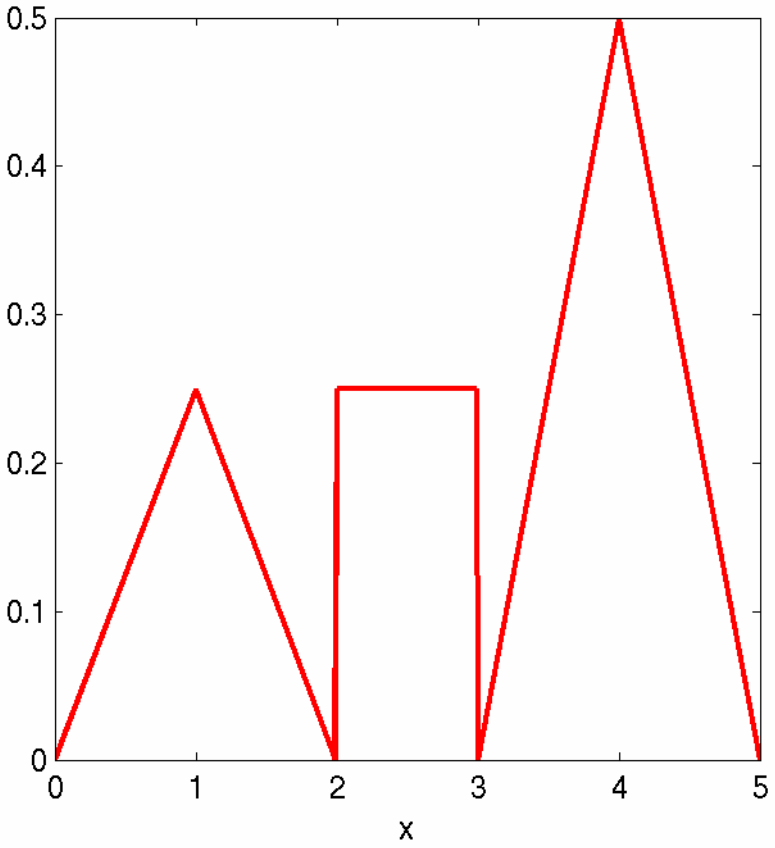


累積分布関数

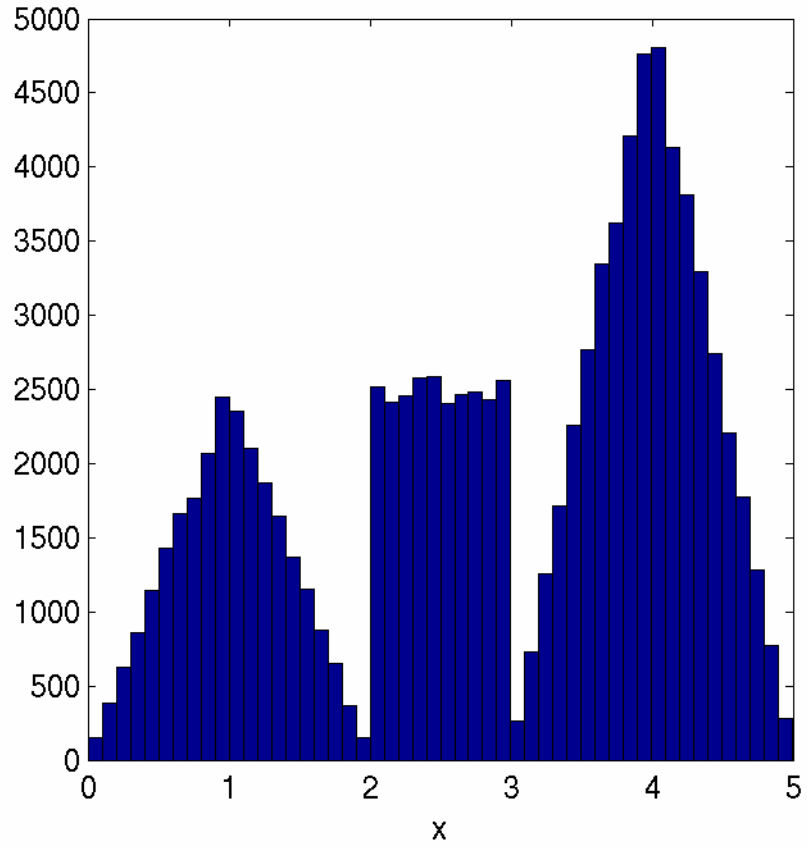


累積分布関数の逆

逆関数法による乱数生成の例（続き）



確率密度関数



生成した乱数の
ヒストグラム

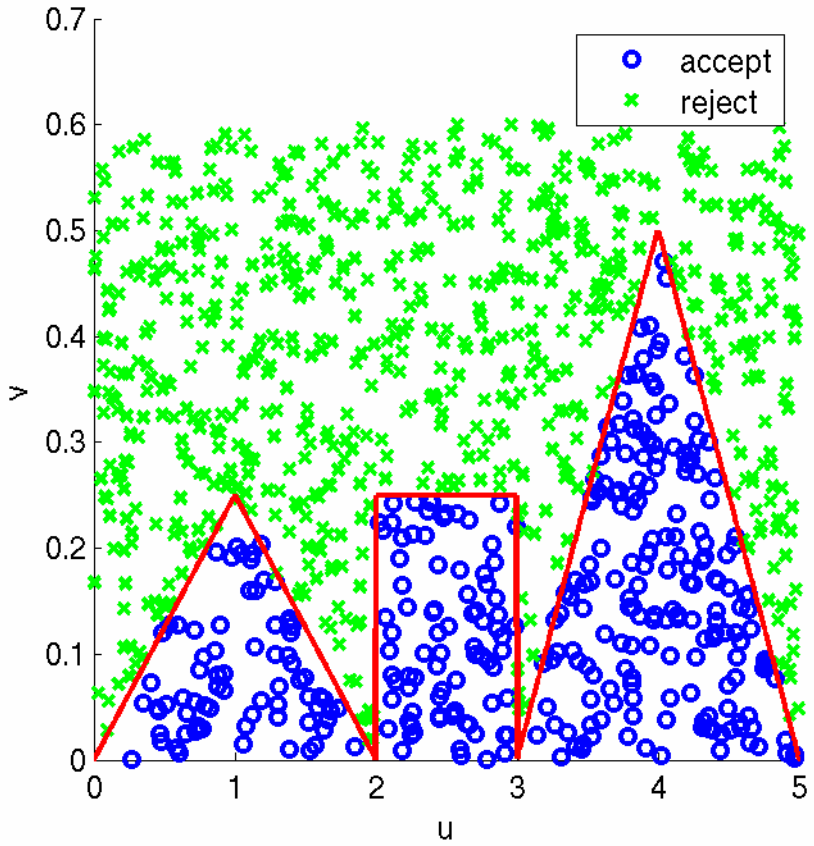
乱数の生成法2：棄却法

$[a, b]$ 上の確率密度関数 $f(x)$ に従う乱数を生成する

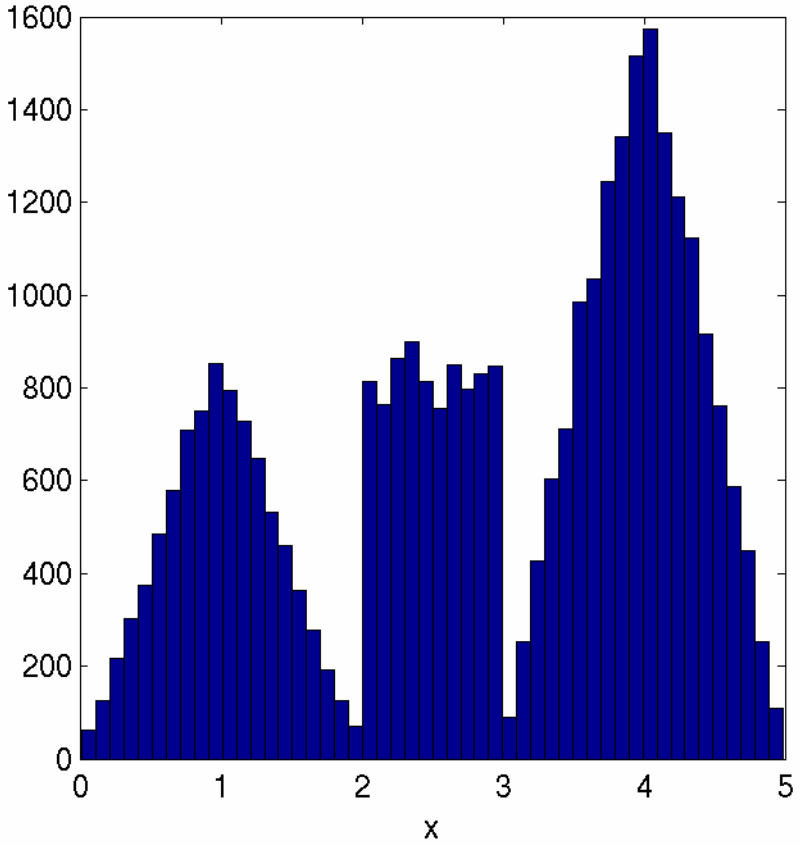
- u：$[a, b]$ 上の一様乱数
- v：$[0, \max_x f(x)]$ 上の一様乱数

- もし，$v \leq f(u)$ ならば，u を採択し，そうでなければ棄却する．
- 必要な数だけ標本が集まるまで，これを繰り返す．

棄却法による乱数生成の例



確率密度関数



生成した乱数の
ヒストグラム

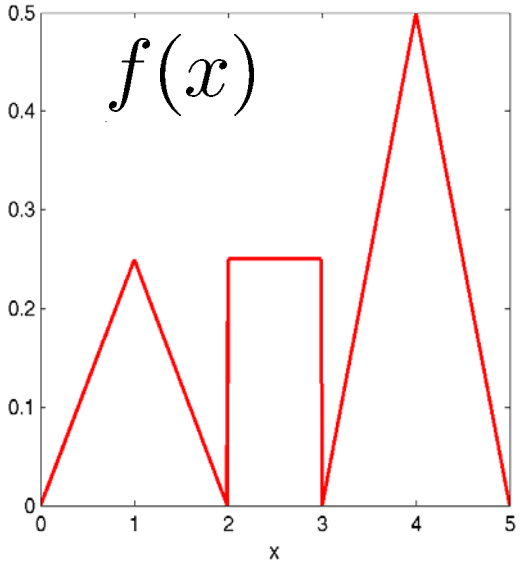
逆関数法と棄却法の問題点

- **逆関数法**:
  - 逆関数がきれいな形で求まらないことがある.
- **棄却法**:
  - 定義域が有限の乱数しか発生させることができない
  - 棄却域が大きい場合，たくさんの乱数を発生させるのに時間がかかる

計算機実験：例

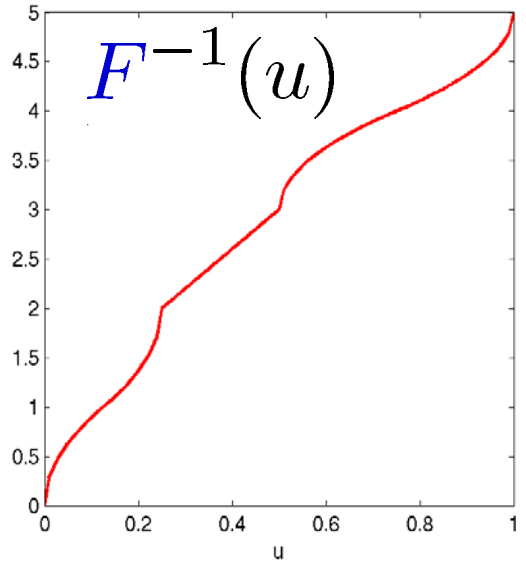
- 逆関数法と棄却法で，次の確率密度関数を持つ確率分布に従う確率変数を生成せよ．

$$f(x) = \begin{cases} \frac{1}{4}x & 0 \le x < 1 \\ \frac{1}{2} - \frac{1}{4}x & 1 \le x < 2 \\ \frac{1}{4} & 2 \le x < 3 \\ -\frac{3}{2} + \frac{1}{2}x & 3 \le x < 4 \\ \frac{5}{2} - \frac{1}{2}x & 4 \le x \le 5 \end{cases}$$



計算機実験：例 (続き)

$$F^{-1}(u) = \begin{cases} \sqrt{8u} & 0 \le u < \frac{1}{8} \\ 2 - \sqrt{2 - 8u} & \frac{1}{8} \le u < \frac{1}{4} \\ 1 + 4u & \frac{1}{4} \le u < \frac{1}{2} \\ 3 + \sqrt{-2 + 4u} & \frac{1}{2} \le u < \frac{3}{4} \\ 5 - \sqrt{4 - 4u} & \frac{3}{4} \le u \le 1 \end{cases}$$



実験環境

- 使用するプログラム言語は何でもよいが, CやJavaなどの低級言語よりも, Octave, Scilab, Rなどのフリーの数値計算ソフトを使うと簡単であろう.
- 以下では, Octaveを使った場合の例を示す
- Octaveに関する質問は以下のアドレスまで

  kitamura@sg.cs.titech.ac.jp

件名は「課題の質問」とし、本文中に名前と学籍番号を明記すること。

参考サイト:

Octave: http://adlib.rsch.tuis.ac.jp/~akira/unix/octave/index-j.html

Scilab: http://www.geocities.jp/rui_hirokawa/scilab/

Octaveとは？

- **フリーの数値計算ソフト。様々なグラフの出力が可能。**
- **ダウンロード先**

http://www.gnu.org/software/octave/

棄却法のメインプログラム

```
1. clear;
2. n=10000;
3. u=5*rand(n,1);
4. v=0.6*rand(n,1);
5. flag=(v<=f(u));
6. clg();
7. axis('auto');
8. gset xlabel "x"
9. gset nokey
10. hist(u(flag==1),50);
11. gset terminal postscript eps color
12. gset output "hist_rej.eps"
13. replot;
```

- 実際には10行目、11行目、12行目はそれぞれ一行で書くこと。

確率密度関数を表示するプログラム

```
1.  clear;
2.  n=10000;
3.  x=linspace(0,5,n);
4.  y=f(x);
5.  u=5*rand(n,1);
6.  v=0.6*rand(n,1);
7.  flag=(v<=f(u));
8.  clg();
9.  hold on
10. axis([0 5 0 0.75]);
11. gset xlabel "u"
12. gset ylabel "v"
13. gset key 4,0.7
14. gset key box
15. plot(u(flag==1),v(flag==
    1), 'bo ;accept;');
16. plot(u(flag~=1),v(flag~=
    1),'gx ;reject;');
17. plot(x,y,'r-');

18. gset terminal postscript
    eps color
19. gset output
    "rejection.eps"
20. replot;
```

- 実際には15行目、16行目、18行目、19行目はそれぞれ一行で書くこと。

確率密度関数のプログラム

以下は12ページの確率密度関数のサンプルプログラム。これをf.mという名前で保存する。

```
1.  function y=f(x)
2.    y=zeros(size(x));
3.    flag=(0<=x & x<1);
4.    y(flag)=0.25*x(flag);
5.    flag=(1<=x & x<2);
6.    y(flag)=-
      0.25*x(flag)+0.5;
7.    flag=(2<=x & x<3);
8.  y(flag)=0.25*ones(size
      (x(flag)))
9.  flag=(3<=x & x<4);
10.   y(flag)=0.5*x(flag)-
      1.5;
11.     flag=(4<=x & x<=5);
12.   y(flag)=
      -0.5*x(flag)+2.5;
13. endfunction
```

- 実際には6行目、8行目、10行目、12行目はそれぞれ一行で書くこと。

逆関数法のメインプログラム

```
1. clear;
2. n=10000;
3. u=rand(1,n);
4. x=Finv(u);
5. clg();
6. axis('auto');
7. gset xlabel "x"
8. gset nokey
9. hist(x,50);
10. gset terminal
    postscript eps color
11. gset output
    "inverse.eps"
12. replot;
```

- 実際には10行目、11行目はそれぞれ一行で書くこと。

累積分布関数の逆関数を表示するプログラム

1. clear;
2. n=10000;
3. u=linspace(0,1,n);
4. x=Finv(u);
5. clg();
6. axis('auto');
7. gset nokey
8. plot(u,x);
9. gset terminal postscript eps color
10. gset output "Finverse.eps"
11. replot;

- 実際には9行目、10行目はそれぞれ一行で書くこと。

累積分布関数の逆関数のプログラム

以下は13ページの累積分布関数の逆関数のサンプルプログラム。

これをFinv.mという名前で保存する。

```
1.  function x=Finv(u)
2.  x=zeros(size(u));

3.    flag=(0<=u & u<1/8);
4.    x(flag)=sqrt(8*u(flag));

5.    flag=(1/8<=u & u<1/4);
6.    x(flag)=2-sqrt(2-
      8*u(flag));

7.    flag=(1/4<=u & u<1/2);
8.    x(flag)=1+4*u(flag);

9.    flag=(1/2<=u & u<3/4);
10.   x(flag)=3+sqrt(4*u(flag)-
      2);

9.    flag=(3/4<=u & u<=1);
10.   x(flag)=5-sqrt(4-
      4*u(flag));

11. endfunction
```

プログラムの解説（1）

- size(x):行列xの大きさを返す。例えば、xが(3×2)の行列ならsize(x)は(3,2)を返す。

- zeros(size(x)): xと同じ大きさの0行列を返す。上の例だと、全ての要素が0の（3×2）の行列を返す。

- clg:グラフから全てのオブジェクトを消去する。グラフの初期化。

プログラムの解説（2）

- rand(n , 1):**各々の要素が一様分布に従って選ばれるn行1列の行列を返す。**
- gset xlabel "u" : **横軸のラベルをuにする。**
- axis('auto'):**グラフの表示範囲を自動で調整する。**
- gset terminal postscript eps color

:**グラフをカラーでepsファイルで出力する。**

プログラムの解説（3）

- flag；

例えば、

```
flag=(0<=x & x<1);
y(flag)=0.25*x(flag);

flag=(1<=x & x<2);
y(flag)=-0.25*x(flag)+0.5;
```

の表記だとxが0以上1未満だと、

```
y(flag)=0.25*x(flag);
```

が実行され、xが1以上2未満だと、

```
y(flag)=-0.25*x(flag)+0.5;
```

が実行される。

プログラムの解説（4）

- plot:

グラフを描画する。例えば、17ページのソースを見ると、

```
flag=(v<=f(u));
plot(u(flag==1),v(flag==1), 'bo ;accept;');
```

上のソースでは、(v<=f(u))**の条件を満たす点を青い○で描画してその青い○を**accept**という名前にする。**

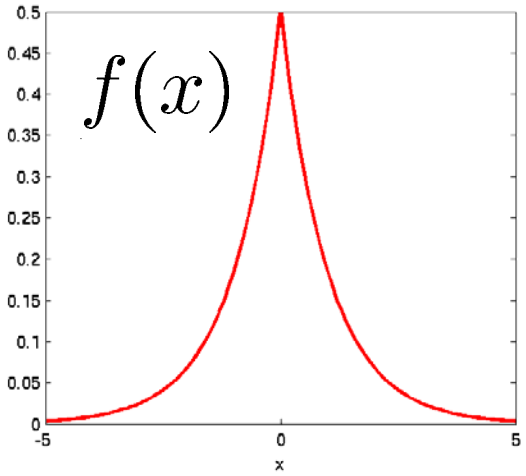
octaveのインストール方法と実行方法

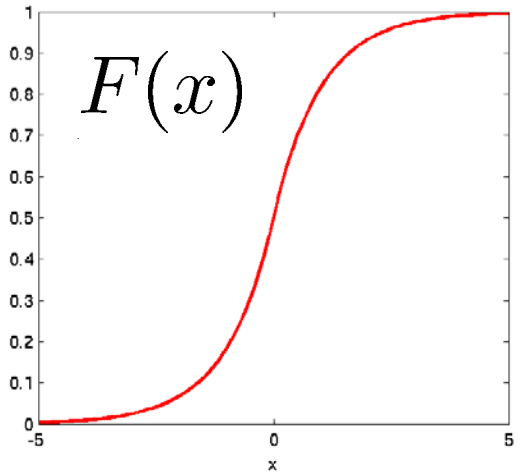
1. １５ページのURLにアクセスして、画面下のほうにあるDownloadをクリック。
2. 画面の真ん中あたりにある Octave-Forge Files という文字をクリック。
3. windowsの人は、Latest File Releases の下にある octave-forge-windows をクリックする。
4. octave-2.1.73-0-inst.exeをクリック。
5. Hostがいくつか表示されるので適当に選んで、そのホストのDownloadというところをクリック。
6. ファイル（パッケージ）を適当な場所に保存。
7. 保存したファイル（パッケージ）をダブルクリック。
8. 適当にOKを押していくとインストール完了。
9. 保存した場所にショートカットアイコンが出来ているはずなので、それをクリックすればoctaveが起動する。
10. 全てデフォルト通りにやると、ローカルディスクのProgram Filesフォルダに GNU Octave 2.1.73というフォルダが出来、さらにそのフォルダの中に、octave_filesというフォルダが出来ているはずなので、この中にプログラムファイルを作成すれば実行できる。
11. 尚、実行はファイル名を拡張子抜きで入力すればできる。

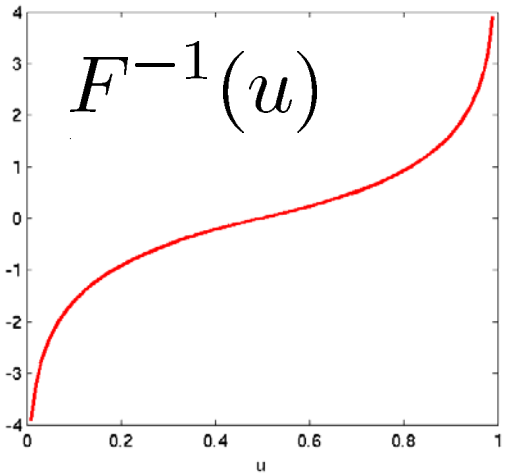
課題 (1)

- 逆関数法と棄却法で，次のラプラス分布に従う確率変数を生成せよ (棄却法では $-5 \leq x \leq 5$ でよい) .

$$f(x) = \frac{1}{2}\exp(-|x|) \qquad F(x) = \begin{cases} \frac{1}{2}\exp(x) & (x < 0) \\ 1 - \frac{1}{2}\exp(-x) & (x \geq 0) \end{cases}$$







$$F^{-1}(u) = \begin{cases} \log(2u) & (u < \frac{1}{2}) \\ -\log(2-2u) & (u \geq \frac{1}{2}) \end{cases}$$

課題 (2)

- 自分で好きな確率分布を定義し，その分布に従う確率変数を逆関数法と棄却法で生成せよ．
- 注意：
  - 確率密度関数 $f(x)$ は

$$f(x) \geq 0, \int f(x)dx = 1$$

を満たさなければならない！

課題 (3)

- 実際に逆関数法と棄却法で乱数を発生させてみた経験を元に，それぞれの手法の長所と短所を自分の言葉で述べよ．

連絡事項

- 5月19日 (金) は各自課題を行なうこと (講義は休講)
- その次の講義は5月31日 (水)
- 課題の〆切は, 5月31日 (水) の講義の最初