5.20 モンテカルロシミュレーション　Composition 法　　ver.0.1

ここでは、若干テクニカルになるけれども、分布を分割して乱数を発生させ、それらを合成する方法である composition method を見ていこう。まずは、逆関数法のところで扱った exponential distribution の分布と逆関数の関係を思い出してみよう。

$F(x)=1-\exp(-\lambda x)$　　$\lambda>0$　ただし$x\geqq 0$

$F^{-1}(x)=-\ln(1-x)/\lambda$　　$\lambda>0$　　$0<x<1$

exponential の逆関数は上のように０から１までの定義域についてなっているから、０から１までの一様乱数をｘの中に代入すればこのパラメーターがλのときの exponential 分布にしたがう乱数を発生させることができた。

対称ラプラス分布 Symmetric Laplace Distribution(double exponential)

逆関数がわかっている exponential はｘが非負の値であるときのみに定義されている分布である。すなわち、パラメータλにたいして分布がｘが大きくなるにしたがって指数的に減衰するような分布であった。これをｘ＝０を対称軸に左右に振り分けたものが対称 Laplace 分布である。exponential が２つということで double exponential とも呼ばれる。今､簡単化のため下のような pdf が次のように定義される分布にしたがう乱数を発生させてみよう｡アイディアは左右対称なので確率 0.5 でｘが負の領域の折り返した分布が発生し、残りの確率 0.5 でｘが歳の領域の分布が発生するということである。

$f(x)=0.5e^{x}$　　$x<0$

$f(x)=0.5e^{-x}$　　$x\geqq 0$

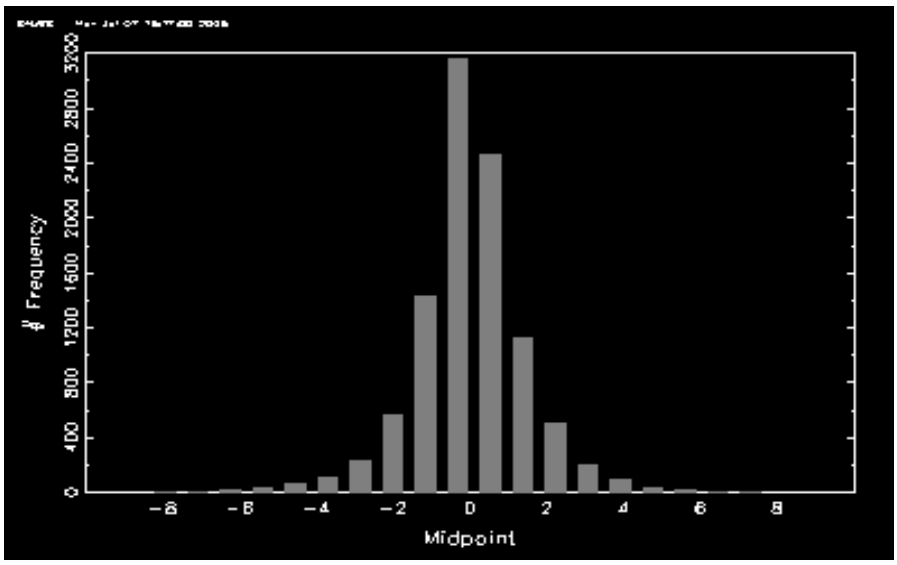
プログラム

```
new; cls;
n=10000;
call compositionf(n);

proc compositionf(n);
    local U,X,j;
    U=rndu(n,1);
    X=zeros(n,1);
    j=1;
    do while j<=n;
        if rndu(1,1)<0.5;
            X[j]=ln(1-U[j]);
        else;
            X[j]=-ln(1-U[j]);
```

```
        endif;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

グラフ表示



ただし、Laplace 分布の pdf の形はパラメーターλに対して、

$$f(x)=\frac{\lambda}{2}e^{-\lambda|x|},\quad -\infty<x<\infty$$

という pdf となる。上の例は、λ＝1 の特殊なケースである。λが 1 ではないときのプログラムは、以下のようになる。

プログラム

```
new; cls;
lambda=2;
n=10000;
call rndlaplace(lambda,n);

proc rndlaplace(lambda,n);
    local U,X,j;
    U=rndu(n,1);
```

```
    X=zeros(n,1);
    j=1;
    do while j<=n;
        if rndu(1,1)<0.5;
            X[j]=1/lambda*ln(1-U[j]);
        else;
            X[j]=-1/lambda*ln(1-U[j]);
        endif;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

λが大きくなるにつれて指数減衰が急になって分布の広がりが狭くなるのは、その逆関数法によるシミュレーションの方法からもわかるであろう。プラスマイナス両側の分布を確率 0.5 ずつの出具合で合成したのがこの Laplace 分布の乱数生成である。

非対称ラプラス分布 Asymmetric Laplace Distribution

株価の収益率の分布などによくあてはめられる、左右が対称でない Laplace 分布を同様にして乱数生成してみよう。ここでは簡単化のため、左右の density からなる面積が 0.5 ずつになるようなケースを考えてみよう。ただし、上の Laplace 分布のλに相当する部分が x が非負の側と負の側では異なる非対称のケースをプログラムする。

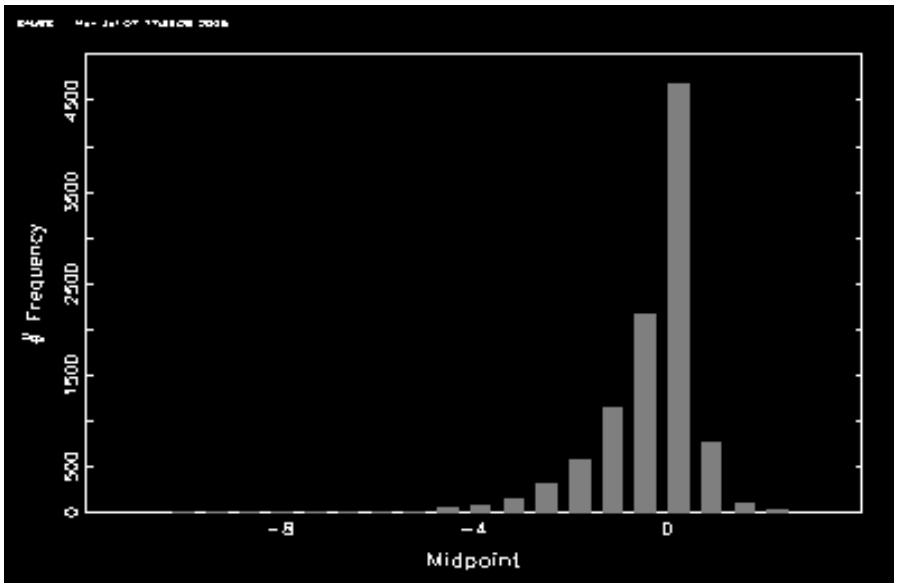
プログラム

```
new; cls;
alpha=1; beta=3;
n=10000;
call rndasymlap(alpha,beta,n);

proc rndasymlap(alpha,beta,n);
    local U,X,j;
    U=rndu(n,1);
    X=zeros(n,1);
    j=1;
```

```
    do while j<=n;
        if rndu(1,1)<0.5;
            X[j]=1/alpha*ln(1-U[j]);
        else;
            X[j]=-1/beta*ln(1-U[j]);
        endif;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

グラフ表示



上のプログラムでは、λに相当する部分をｘが負の領域の値に対してはαというパラメータとして、通常の逆関数にはマイナスがついているのをとった値にしている。非負の領域のｘに対してはλに相当するパラメータをβとしてαとは区別して通常どおりの逆関数法で乱数を発生させている。実際の分析では、標準化されたデータにあてはめ、左右の density を独立に測定するため、ｘ＝０のところの切片はともに１になるはずであるが、ここでは左右の調整は行なっていないので、切片はe^0の値に（すなわち１に）αまたはβをかけてさらに確率 0.5 をかけた値がそれぞれくる。

超指数分布 **Hyperexponential Distribution**

今度は、exponential を非負の側だけに合成する Hyperexponential を考える。まずは、２つの exponential をつなぎ合わせる簡単なケースを扱う。

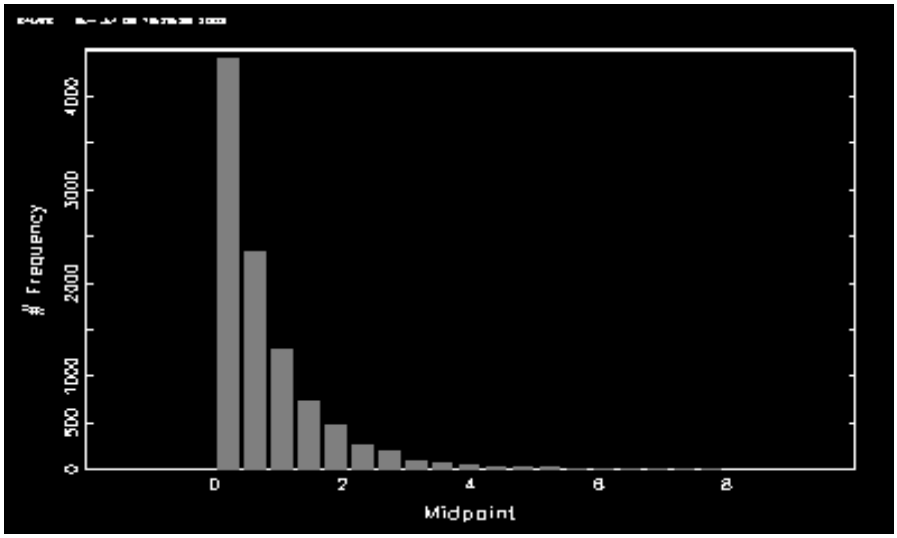
プログラム

```
new; cls;
n=10000;
call hyperexp(0.6,0.4,1,2,n);

proc hyperexp(alpha1,alpha2,lambda1,lambda2,n);
    local Ua,I,j,Ub,X;
/* Parameter check */
    if lambda1<=0 or lambda2<=0;
        errorlog "ERROR: Parameter lambda's must be positive number.";
        retp(".");
    elseif alpha1<0 or alpha2<0;
        errorlog "ERROR: Parameter alpha's must be non negative.";
        retp(".");
    elseif alpha1+alpha2/=1;
        errorlog "ERROR: Relation alpha1+alpha2=1 must hold.";
        retp(".");
    endif;
/* Generate I */
    Ua=rndu(n,1);
    I=zeros(n,1);
    j=1;
    do while j<=n;
      if Ua[j]<alpha1;
          I[j]=1;
      else;
          I[j]=2;
      endif;
      j=j+1;
    endo;
/* Generate X */
    Ub=rndu(n,1);
    X=zeros(n,1);
```

```
    j=1;
    do while j<=n;
        if I[j]==1;
            X[j]=-1/lambda1*ln(Ub[j]);
        else;
            X[j]=-1/lambda2*ln(Ub[j]);
        endif;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

画面表示



上の例は、２つの exponential を確率α_1とα_2の割合で composition したものである。これまでの composition では一貫して確率は半分半分であった。さらに次のような composition method の一般形

$$F(x)=\alpha_1F_1(x)+\alpha_2F_2(x)+\ \ldots\ +\alpha_nF_n(x)$$

$$\text{ここで}\alpha_1+\alpha_2+\ \ldots\ +\alpha_n=1$$

について、この hyperexponential のケースをあてはめて乱数発生させると次のようになる。

なお､分布について上のような composition が言えると一般に pdf についても同じように考えることができる。当然のことながら、pdf を積分した面積の総和は１となる。

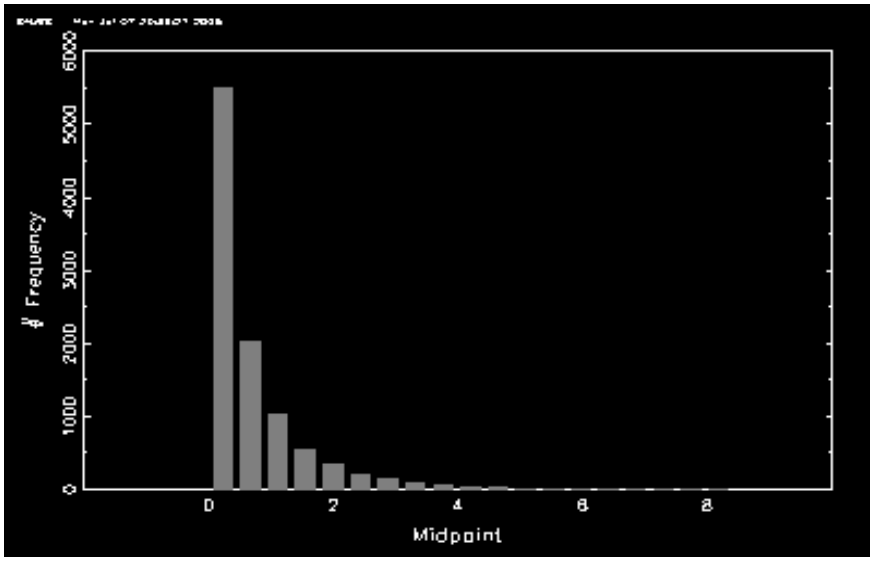
プログラム（一般形であるｎ個の exponential の composition）

```
new; cls;
alpha={0.5,0.2,0.1,0.1,0.1};
lambda={1,2,3,4,5};
n=10000;
call rndhyperexp(alpha,lambda,n);

proc rndhyperexp(alpha,lambda,n);
    local Ua,I,j,k,Ub,X;
/* Parameter check */
    if not lambda>0;
        errorlog "ERROR: Parameter lambda's must be positive number.";
        retp(".");
    elseif not alpha>=0;
        errorlog "ERROR: Parameter alpha's must be non negative.";
        retp(".");
    elseif fne(sumc(alpha),1);
        errorlog "ERROR: Relation alpha1+alpha2+...=1 must hold.";
        retp(".");
    endif;
/* Generate I */
    alpha=cumsumc(alpha);
    Ua=rndu(n,1);
    I=zeros(n,1);
    j=1;
    do while j<=n;
       k=rows(alpha);
       do while k>=1;
           if Ua[j]<alpha[k];
               I[j]=k;
           endif;
           k=k-1;
       endo;
```

```
        j=j+1;
    endo;
/* Generate X */
    Ub=rndu(n,1);
    X=zeros(n,1);
    j=1;
    do while j<=n;
        k=1;
        do while k<=rows(alpha);
            if I[j]==k;
                X[j]=-1/lambda[k]*ln(Ub[j]);
            endif;
            k=k+1;
        endo;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

グラフ表示



上では、確率パラメーターαと指数分布のパラメーターλが列ベクトルの形式で与えられ

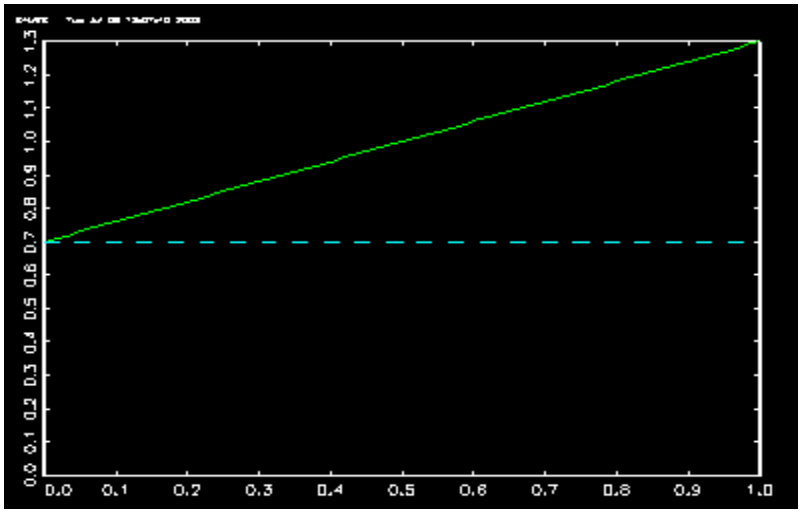
ているものとしている。パラメーターチェックのあと、そのαの累積和を計算する。パラメーターαの要素の和は１であるから、この累積和も最終的には１となる。それを後ろからその区間に入っているものは、インデックスをｎ、そして n-1,...という具合に順番に番号をつける。そうしてできあがった、1,2,3,...,n のインデックス番号をもとにその時々に変化するλに対応づけて逆関数法で乱数発生をさせている。これは、前の２つの関数の composition のケースをｎ個の関数にも一般化したプログラムである。

水平方向の分割

ラプラス分布やその非対称分布などの composition のケースでは暗黙のうちに垂直方向に分布を分割させ、それぞれの乱数を作成し確率（例えば 0.5 と 0.5）で composition をしていた。垂直方向に分割できるのであれば、水平方向にも分割することは可能である。以下ではその簡単な例について見る。一番簡単なのは、以下のように台形を横に倒したような形を三角分布と一様分布に分割することである。

```
new; cls;
a=0.7;
x=seqa(0,0.01,101);
fn f(x,a)=a*1+(1-a)*2*x;
fn g(x,a)=a*1*ones(rows(x),1);

library pgraph;
graphset;
xtics(0,1,0.1,0);
ytics(0,2-a,0.1,0);
xy(x,f(x,a)~g(x,a));
```



上のような０と１の間の単純な分布の density を式で表すと

$f(x)=a+2x(1-a)$　　$0 \leqq x \leqq 1$

$=0$　　Otherwise

となる。これを composition の適用できる確率×f(x)の和の形である

$f(x)=\alpha_1 f_1(x)+\alpha_2 f_2(x)+ \ldots +\alpha_n f_n(x)$　　ここで$\alpha_1+\alpha_2+ \ldots +\alpha_n=1$

の形にするため簡単な式に操作を行なうと、

$f(x)=a \cdot 1+ (1-a) \cdot 2x$　　$0 \leqq x \leqq 1$

$=0$　　Otherwise

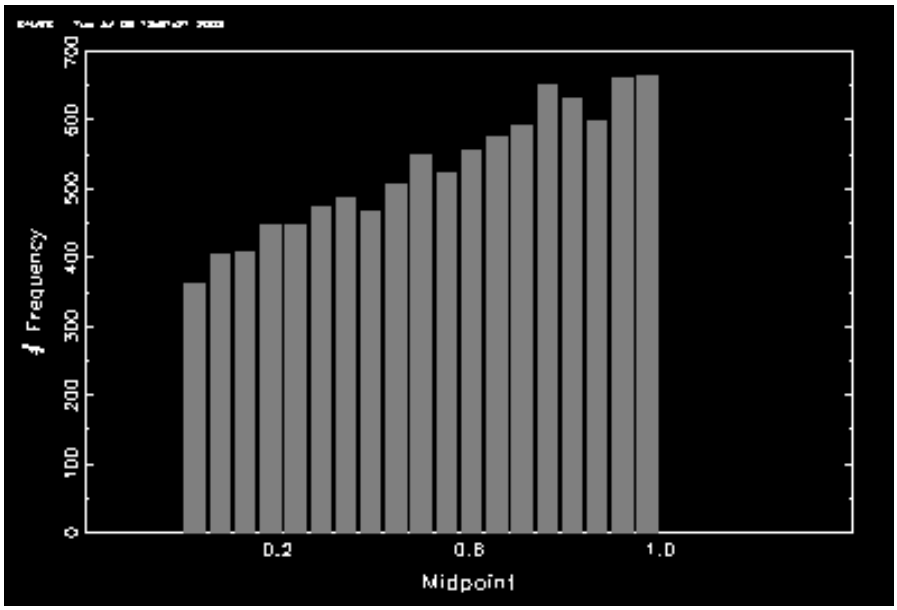
というような確率×f(x)の和の形に変形できる。これをもとに、実際にａに確率にふさわしい数値を何か代入して計算してみよう。以下はａ＝0.7 のケースの上のような分布の乱数を上下に水平分割して確率 0.7 と 0.3 でつなぎ合わせて composition したものである。

プログラム

```
new; cls;
a=0.7;
n=10000;
x=rndhori(a,n);

proc rndhori(a,n);
    local U,X,j;
    U=rndu(n,1);
    X=zeros(n,1);
    j=1;
    do while j<=n;
        if rndu(1,1)<a;
            X[j]=U[j];
        else;
            X[j]=sqrt(U[j]);
        endif;
        j=j+1;
    endo;
/* Histogram of X */
    library pgraph;
    graphset;
    call hist(X,19);
    retp(X);
endp;
```

グラフ表示



上のプログラムでは、先にＵという０と１の間の一様乱数をｎ個作成しておき、確率ａで（１のうち新たな一様乱数がａよりちいさければ）そのままＵを用い、そうでなければ（すなわち確率(1－a)で）三角分布を逆関数のルートのｘで作成しその中にＵを入れている。それらからできた数列をプロットしたものができあがったうえのような台形をたおしたような長方形と三角形に水平方向に分割できる分布である。なおｘ＝０のときは f(x)＝ａ、そしてｘ＝１のときは f(x)＝２－ａとなるから、ａ＝0.7 の上のケースでは、理論的には f(0)=0.7 かつ f(1)=1.3 となるはずである。また、下の長方形の面積は 0.7、上の三角形の面積は 0.3 となり合計は当然のことながら１となる。乱数を作成するときは確率がかかるので、もとの単独で面積が１になるように通常どおり発生させるのである。

このように、composition method というのは、分布を分割し、それぞれの分布の乱数を発生させたものを確率比で加重平均したものと言える。