EE511 Project:6

ZHANG Fan

USC ID 1417-68-5115

Contents

[Question 1: 2](#_Toc478227856)

[Analysis: 2](#_Toc478227857)

[Box-muller method 2](#_Toc478227858)

[Code for Box-muller method 2](#_Toc478227859)

[Code for the Marsaglia’s polar method 3](#_Toc478227860)

[Results and comment: 3](#_Toc478227861)

[Box-muller method 3](#_Toc478227862)

[Marsaglia’s polar method 4](#_Toc478227863)

[Question 2: 5](#_Toc478227864)

[Analysis and algorithm: 5](#_Toc478227865)

[Code: 5](#_Toc478227866)

[Results: 6](#_Toc478227867)

[Comment 6](#_Toc478227868)

[Question 3 6](#_Toc478227869)

[Introduction 6](#_Toc478227870)

[Simulation algorithm theory: 7](#_Toc478227871)

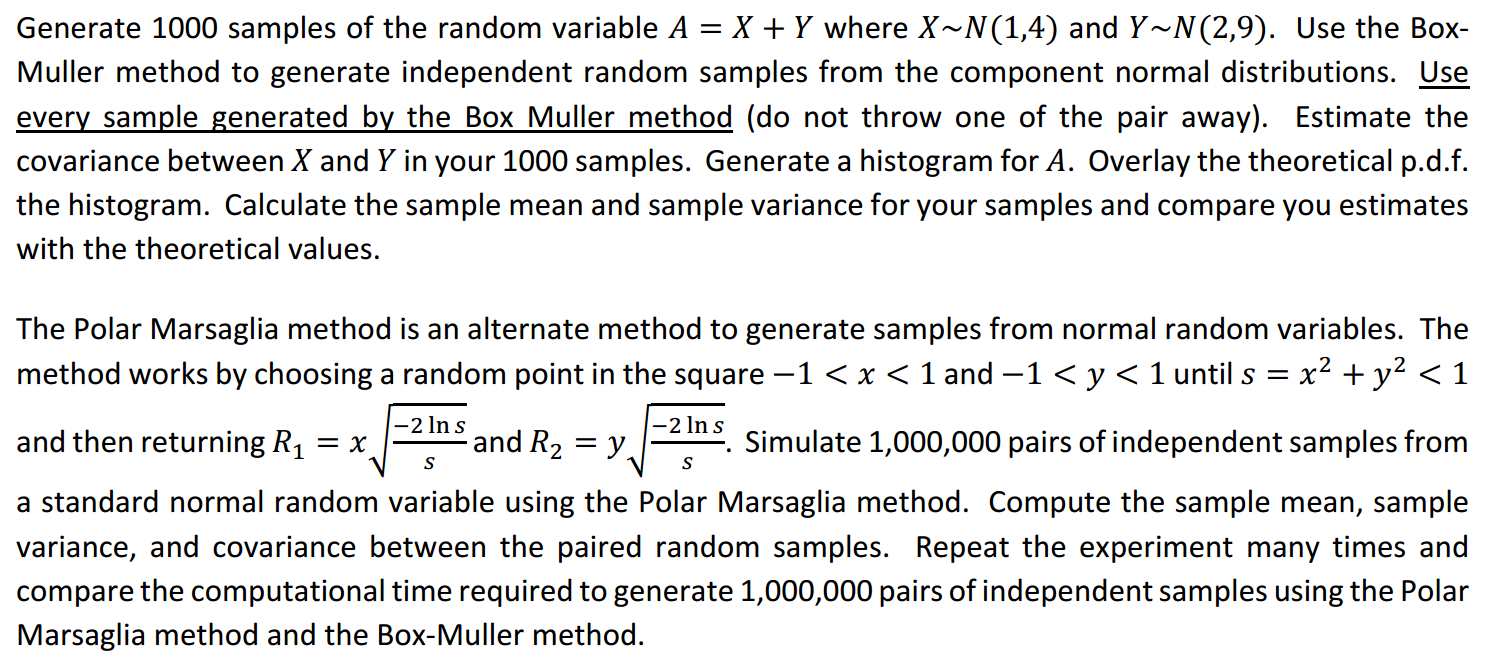
[Main code: 7](#_Toc478227872)

[Function code of simulation theory 8](#_Toc478227873)

[Results: 8](#_Toc478227874)

[Comments 9](#_Toc478227875)

# Question 1:



## Analysis:

### Box-muller method

* + it takes u1, u2, two independent uniformly distributed random variables on (0, 1)
  + defines X = sqrt( −2 log(u1) cos(2π\*u2 )), Y= sqrt(−2 log(u1) sin(2π\*u2) )It can be proved that X and Y are N(0, 1) random variables, and independent.
  + Scale them to a particular mean and variance: if Z ∼ N(0, 1) then X := µ + σZ ∼ N(µ, σ^2 )
  + Calculate covariance, variance, and mean of x and y.
  + Generate histogram for A. Pay attention to scale the width and normalization.
  + Generate theoretical pdf. The theoretical pdf **A~N(M1+M2,var(V1+V2)=Var(V1)+Var(V2)+Cov(V1,V2))**
  + Compare with theoretical mean, variance, and overlay the figure.
* Then for the Marsaglia’s polar method,
  + Take u1, u2 from uniform distribution on ( − 1, 1)
  + Accept if s = u1^2 + u2^2 < 1, otherwise get new u1, u2
  + Let X = sqrt( −2 log(s)/s)\*u1, Y = sqrt( −2 log(s)/s)\*u2
* Compare the time elapse.

### Code for Box-muller method

tic;% Start the timer

clear all; close all; clc;

M1 = 1; % Mean of X

M2 = 2; % Mean of Y

V1 = 4; % Variance of X

V2 = 9; % Variance of Y

u1 = rand(1000000,1);

u2 = rand(1000000,1);

% Geberate X and Y that are N(0,1) random variables and independent

X = sqrt( - 2\*log(u1)).\*cos(2\*pi\*u2 );

Y = sqrt( - 2\*log(u1)).\*sin(2\*pi\*u2 );

% Scale them to a particular mean and variance

x = sqrt(V1)\*X + M1; % x~ N(M1,V1)

y = sqrt(V2)\*Y + M2; % y~ N(M2,V2)

toc; % Read elapsed time from stopwatch(Q1\_6 compare the computational time)

A=x+y; h=histogram(A,'Normalization','probability','BinWidth',1);hold on; %Q1\_2, generate histogram for A

NormX=-5:0.1:15;%Q1\_3 overlay theoretical norm distribution plot

plot(NormX,normpdf(NormX,M1+M2,sqrt(V1+V2)));

Cov\_XY=cov(x,y)%Q1\_1: calculate the covariance of X and Y

Meanx=mean(x)%Q1\_4 sample mean and var and compare

Meany=mean(y)

## Code for the Marsaglia’s polar method

tic;% Start the timer

clear all; close all; clc;

M1 = 1; % Mean of X

M2 = 2; % Mean of Y

V1 = 4; % Variance of X

V2 = 9; % Variance of Y

i = 0; % the random number generated by the algorithm

% Geberate X and Y that are N(0,1) random variables and independent

while(i<=999999)

u1 = 2\*rand()-1;

u2 = 2\*rand()-1;

s = u1^2 + u2^2;

if(s < 1)

i = i + 1;

X(i) = sqrt(-2\*log(s)/s)\*u1;

Y(i) = sqrt(-2\*log(s)/s)\*u2;

end

end

% Scale them to a particular mean and variance

x = sqrt(V1)\*X + M1; % x~ N(M1,V1)

y = sqrt(V2)\*Y + M2; % y~ N(M2,V2)

toc; % Read elapsed time from stopwatch,(Q1\_6 compare the computational time)

cov(x,y)%Q\_5 mean, variance, cov of x and y

meanx=mean(x)

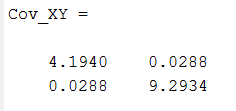
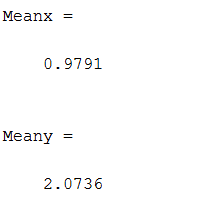
meany=mean(y)

## 

## Results and comment:

### Box-muller method

Mean, covariance and variance of sample X and Y:



Mean of sample x: 0.9791; theoretical mean of x: 1;

Mean of sample y: 2.0736; theoretical mean of y: 2;

Variance of sample x: 4.1940; theoretical variance of x: 4;

Variance of sample y: 9.2934; theoretical variance of y: 9;

Covariance of x and y: 0.0288≈0. Therefore, x and y is unrelated.

And since x and y is independent, so the theoretical covariance is 0. So the result is correct.

The histogram of A is as the following figure shows, as well as the theoretical pdf of A:

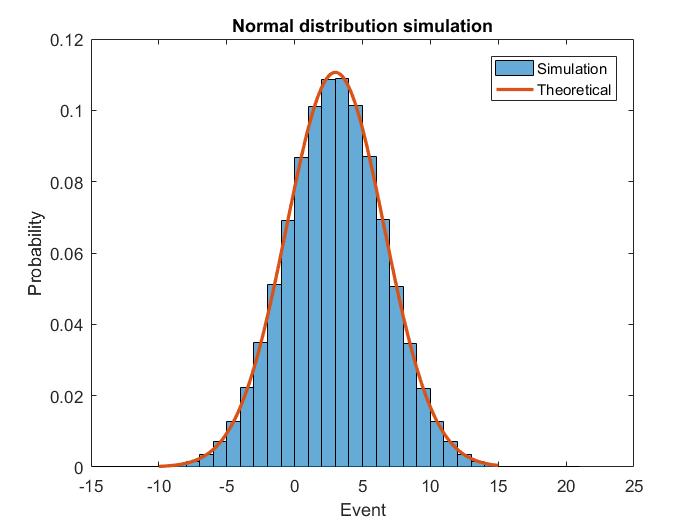
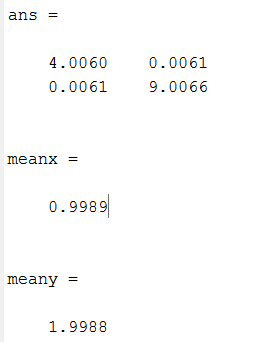


Figure 1 overlay of theoretical and simulation pdf of A

From the figure above shows we can see that the theoretical one and the histogram of sample A is very close.

### Marsaglia’s polar method



From the figure left we can see that:

Mean of sample x: 0.9989; compared to the theoretical mean of x: 1;

Mean of sample y: 1.9988; compared to the theoretical mean of y: 2;

Variance of sample x: 4.0060; compared to the theoretical variance of x: 4;

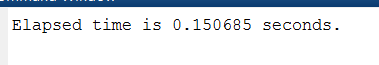
Variance of sample y: 9.0066; compared to the theoretical variance of y: 9;

Covariance of x and y: 0.0061≈0. Therefore, x and y is unrelated.

And since x and y is independent, so the theoretical covariance is 0. So the result is correct.

Plus, this results obviously are more accurate than those in Box-muller method, which are because the sample number is 1000000 compared to 1000 in Box-muller method.

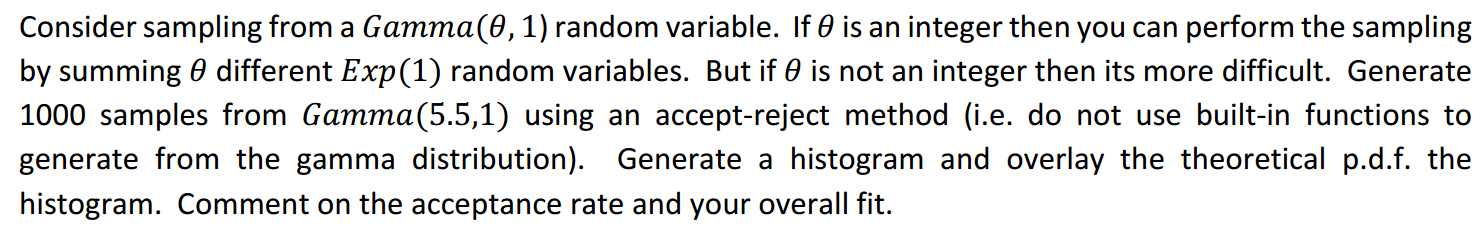
* Compare the time elapse.

Box-muller method:

Marsaglia’s polar method:

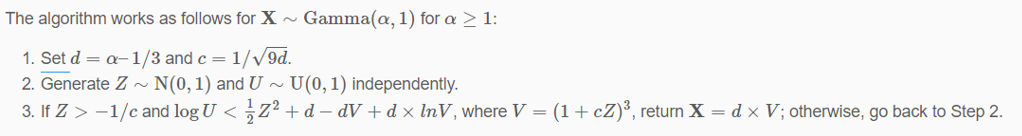
Therefore, the Box-muller method is much efficient than the Marsaglia’s polar method.

# Question 2:



## Analysis and algorithm:

1. Generate Y~g(x)
2. Find ,
3. generate U~(0,1), if U<, accept Y. let X=Y。



## Code:

clear

theta=5.5

d=theta-1/3;

c=1/sqrt(9\*d);

i=0;

while(i<1000)

Z=normrnd(0,1);

U=rand();

V=(1+c\*Z)^3;

if(Z>-1/c && log10(U)<1/2\*Z^2+d-d\*V+d\*log(V))

X(i+1)=d\*V;

i=i+1;

end

end

%theoretic gamma

histogram(X,'Normalization','probability','BinWidth',1)

hold on;

x=-5:0.1:15;

plot(x,gampdf(x,5.5,1),'LineWidth',2)

%set attributes of the figure

title('Gamma distribution simulation and theoretical curve')

xlabel('Event');

ylabel('Probability')

legend('Simulation','Theoretical')

## Results:

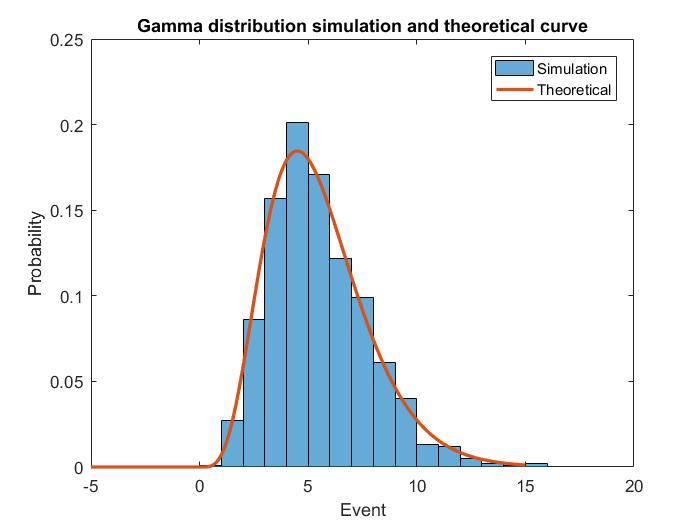
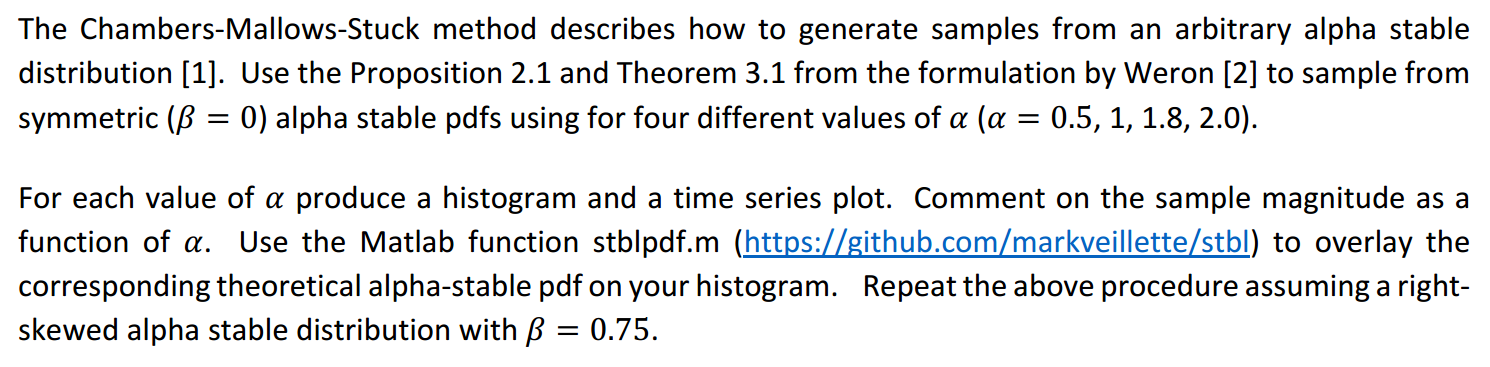


Figure 2overlay of theoretical and simulation pdf of Gamma distribution

## Comment

* My acceptance rate is according to pdf of and , , and the acceptance rate is: .
* The more similar that g(x) and f(x) are, the higher acceptance rate is, and the higher efficiency the algorithm is.
* The overall fit is close to the theoretical distribution.

# Question 3



## Introduction

The variable http://math.bu.edu/people/mveillet/html/alphastablepub_eq14833.png is usually called **a standard alpha-stable random variable** (but keep in mind the word "standard" depends on the choice of parameterization!). The alpha-stable distribution is a four-parameter family of distributions and is (usually) denoted by http://math.bu.edu/people/mveillet/html/alphastablepub_eq07061.png.

* The first parameter http://math.bu.edu/people/mveillet/html/alphastablepub_eq67893.png is called the characteristic exponent, and describes the tail of the distribution.
* The second, http://math.bu.edu/people/mveillet/html/alphastablepub_eq32153.png is the skewness, and as the name implies, specifies if the distribution is right- (http://math.bu.edu/people/mveillet/html/alphastablepub_eq76915.png) or left- (http://math.bu.edu/people/mveillet/html/alphastablepub_eq75308.png) skewed.
* The last two parameters are the scale, http://math.bu.edu/people/mveillet/html/alphastablepub_eq08866.png, and the location http://math.bu.edu/people/mveillet/html/alphastablepub_eq88041.png. One can think of these two as being similar to the variance and mean in the normal distribution in the following sense - if http://math.bu.edu/people/mveillet/html/alphastablepub_eq27343.png, then if http://math.bu.edu/people/mveillet/html/alphastablepub_eq47709.png, http://math.bu.edu/people/mveillet/html/alphastablepub_eq68478.png.

The family of alpha-stable distributions is a rich class, nd includes the following distributions as subclasses:

* For α = 2 the distribution reduces to a [Gaussian distribution](https://en.wikipedia.org/wiki/Gaussian_distribution) with variance σ2 = 2c2 and mean μ; the skewness parameter β has no effect.
* For α = 1 and β = 0 the distribution reduces to a [Cauchy distribution](https://en.wikipedia.org/wiki/Cauchy_distribution) with scale parameter c and shift parameter μ.
* For α = 1/2 and β = 1 the distribution reduces to a [Lévy distribution](https://en.wikipedia.org/wiki/L%C3%A9vy_distribution) with scale parameter c and shift parameter μ.

## Simulation algorithm theory:

According to the paper 2,

,

for , :

for ,:

## Main code:

clear

t=-10:0.1:10;

alpha=[0.5,1,1.8,2.0];%set alpha

beta=0.75;%set beta

gama=1;%why 1?

for z=1:length(alpha)

for i=1:10000

X(i)=Sim(alpha(z),beta);

end

subplot(2,2,z);

histogram(X,'Normalization','probability','Binwidth',1)

hold on;%above code obtain simulation histogram

T=stblpdf(t,alpha(z),beta,gama,0);%obtain theoretical values

plot(t,T,'LineWidth',2)

%set plot attributes

axis([-10 10 0 0.6])

title(['alpha=', num2str(alpha(z)),' beta=',num2str(beta)])

xlabel('Event');

ylabel('Probability')

legend('Simulation','Theoretical')

end

## Function code of simulation theory

function X=Sim(alpha,beta2)

gama=pi\*rand()-pi/2;

W=exprnd(1);

K=alpha-1+sign(1-alpha);

gama0=-pi/2\*beta2\*K/alpha;

if alpha ~=1

X=sin(alpha\*(gama-gama0))/(cos(gama))^(1/alpha)\*(cos(gama-alpha\*(gama-gama0))/W)^((1-alpha)/alpha);

if X>10 X=11;

elseif X<-10 X=-11;

%some of too large or too small values are set to a fixed value so that when plot the figure,

%they will not influence the other value.

end

elseif alpha==1

X=(pi/2+beta2\*gama)\*tan(gama)-beta2\*log10(W\*cos(gama)/((pi/2)+beta2\*gama));

end

## C:\Users\program\OneDrive\document\USC\2017 USC\EE511\hw6\beta0.jpgResults:

Figure 3overlay of theoretical and simulation pdf of α-stable distribution (β=0)

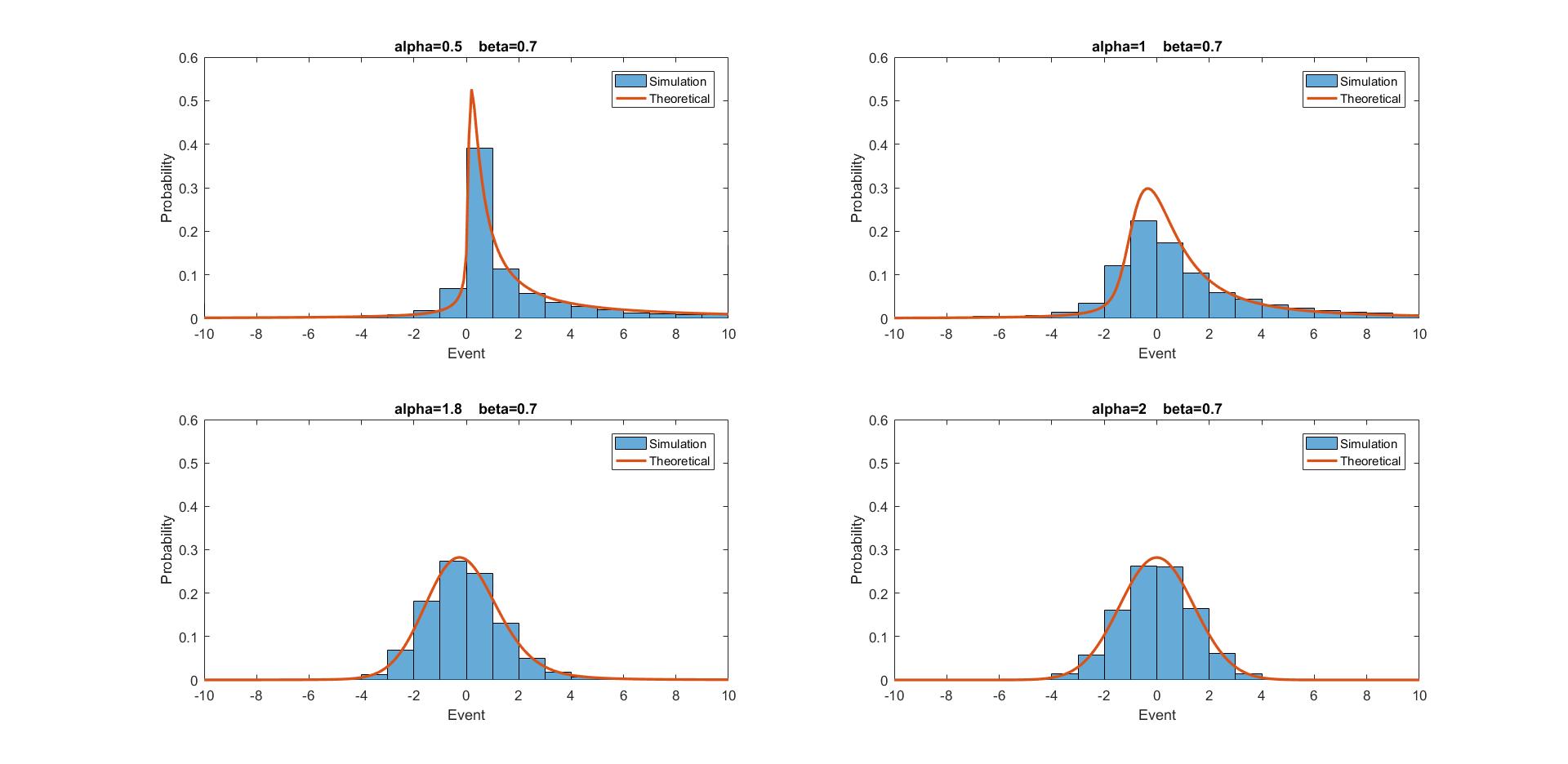
Figure 3: overlay of theoretical and simulation pdf of -stable distribution ()

Figure 4 overlay of theoretical and simulation pdf of α-stable distribution (β=0.7)

## Comments

As we can see from the above two figures, we can draw some conclusions:

* when <1, the probability of occurrence of too large or too small greatly increases.
* When , the figure is symmetric; while , the figure skews to right side; and while , the figure skews to left side.
* When , the method of the paper will cause a relatively large error while
* .