Problem Statement

Consider a continuous random variable defined by its probability density function,

$$f_X(x) = C\sin(x) \qquad 0 \le x \le \pi \ \text{(and 0 otherwise)}$$

where C is a normalizing constant.

A) Find the distribution function, $F_X(x)$, and evaluate C. Compute (analytically) the (population) mean, μ, and variance, σ^2.

B) By inverting the distribution function, you can use the rand() function in MATLAB to directly generate samples of X . Generate a set of N samples (for N=25,100, 1000). For each case plot the empirical distribution generated by these samples and compare to the true underlying distribution function.

C) For the case of N = 100, find the sample mean, m. Use the population variance to find the MSE of the mean estimate. Now suppose the population variance was not known, compute the MSE of the mean estimate based on the sample variance s. How do they compare?

D) For the case of N=100, use the bootstrapping technique to generate M bootstrap samples based on the empirical distribution found in part B and compute the mean and sample variance for each Bootstrap sample. Use M = 50. (If you have time you can try some other values of M)

E) We could try to find the MSE of the sample variance estimate based on the population variance.

$$MSE_F(s^2) = E_F[(\sigma^2 - s^2)^2] = E_F[(\sigma^4 - 2\sigma^2 s^2 + s^4)^2] = \sigma^4 - 2\sigma^2 E_F[s^2] + E_F[s^4]$$

We saw that $E_F[s^2] = \sigma^2$ so we have that $MSE_F(s^2) = E_F[s^4] - \sigma^4$ which is just var(s^2). It can be shown that:

$VAR(s^2) = \frac{1}{n}\left[\mu_4 - \frac{n-3}{n-1}\sigma^4\right]$ where $\mu_4 = E[(X-\mu)^4]$ the 4th central moment.

If F doesn't have simple structure, this seems a bit challenging. So, instead, we find an approximation using bootstrap approach. We can calculate the variance of the empirical distribution – call this ${\sigma_{F*}}^2$. We could evaluate

$$MSE_{F*}(s^2) = E_{F*}[(s^2 - {\sigma_{F*}}^2)^2]$$

by computing s^2 for all possible n^n samples that can be generated from the empirical distribution. That is a formidable computational task, so we consider only a (random) subset of such samples – i.e. the set of Bootstrap samples in part D and use the sample variances found in part D to estimate the MSE.

Part A

As we have

$$f_X(x) \ = \ Csin(x)\ 0 \ \leq \ x \ \leq \ \pi$$

By doing the integral, we can easily get:

$$\int_{-\infty}^{\infty} f_X(x)dx = \int_0^{\pi} Csindx = -Ccos(c)|_0^{\pi} = -C(-1-1) = 1$$

$$\therefore C = \frac{1}{2}$$

Then we get,

$$F_X(x) = \int_0^x \frac{1}{2}\sin(x)\,dx = \frac{1}{2}(1-\cos(x))\,, 0 \leq x \leq \ \pi$$

At x = 0, $F_X(x)$ = 0, and at $x = \pi$, $F_X(x)$ = 1.

Now, mean and variance are given by,

$$\mu = \int_{-\infty}^{\infty} x f_X(x)dx = \int_0^{\pi} \frac{x}{2}\sin(x)dx = \frac{\pi}{2}$$

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)dx = \int_0^{\pi} \frac{x^2}{2}\sin(x)dx = \frac{\pi^2}{2} - 2$$

$$\therefore \sigma^2 = E[X^2] - E[X]^2 = \frac{\pi^2}{4} - 2$$

Part B

To get the empirical distribution, we have

$$F_n^*(x) = \frac{1}{n}\sum_{i=1}^{n} \delta(X_i \leq x)$$

Now, we invert the distribution function $X = arcos(1-2Y)$. We can generate X where Y is U(0,1).

Code:

```
finv = finverse(cdf_fx)
% N = 25
y1 = zeros(0);
x1 = zeros(0);
for i =1:25
    y1(i) = rand();
    x1(i) = subs(finv,t,y1(i));
end;
figure(1);
cdfplot(x1);
hold on;
ezplot(cdf_fx,[0,pi]);


% N = 100
y2 = zeros(0);
```

```
x2 = zeros(0);
for i =1:100
    y2(i) = rand();
    x2(i) = subs(finv,t,y2(i));
end;
figure(2);
cdfplot(x2);
hold on;
ezplot(cdf_fx,[0,pi]);

% N = 1000
y3 = zeros(0);
x3 = zeros(0);
for i =1:1000
    y3(i) = rand();
    x3(i) = subs(finv,t,y3(i));
end;
figure(3);
cdfplot(x3);
hold on;
fplot(cdf_fx,[0,pi]);
```

Output:

N = 25,

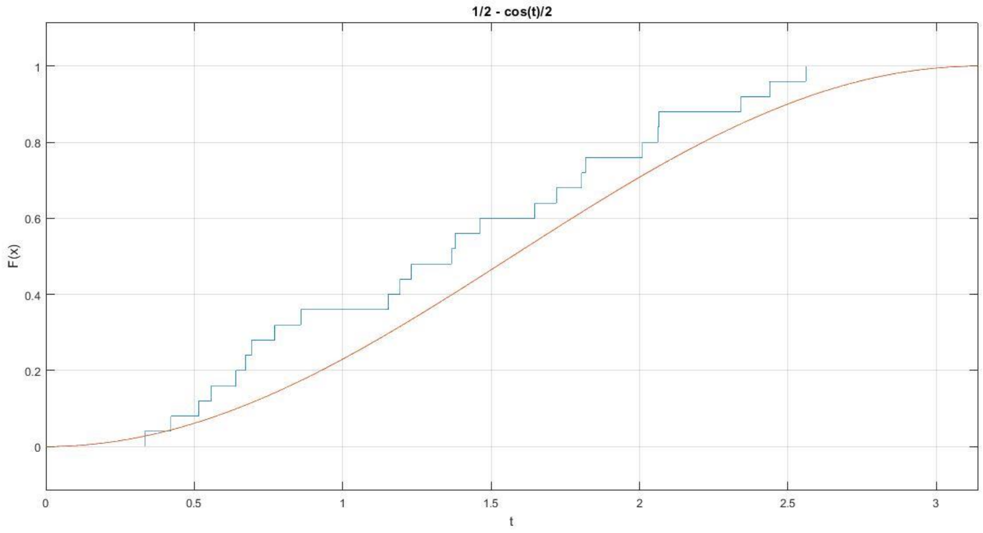

Figure 1 Empirical distribution VS true underlying distribution function n=25

As we can see in these three figures, when n is getting larger the empirical distribution curve is more and more closer to the true underlying distribution function curve. It is notable that when n is small, the empirical distribution is like a staircase function.

N = 100,

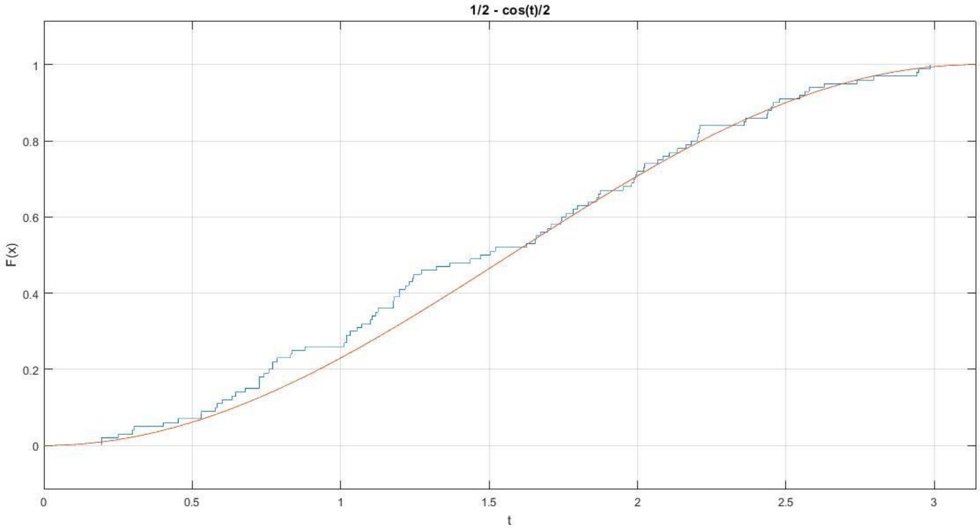

Figure 2 Empirical distribution VS true underlying distribution function n=100

N = 1000,

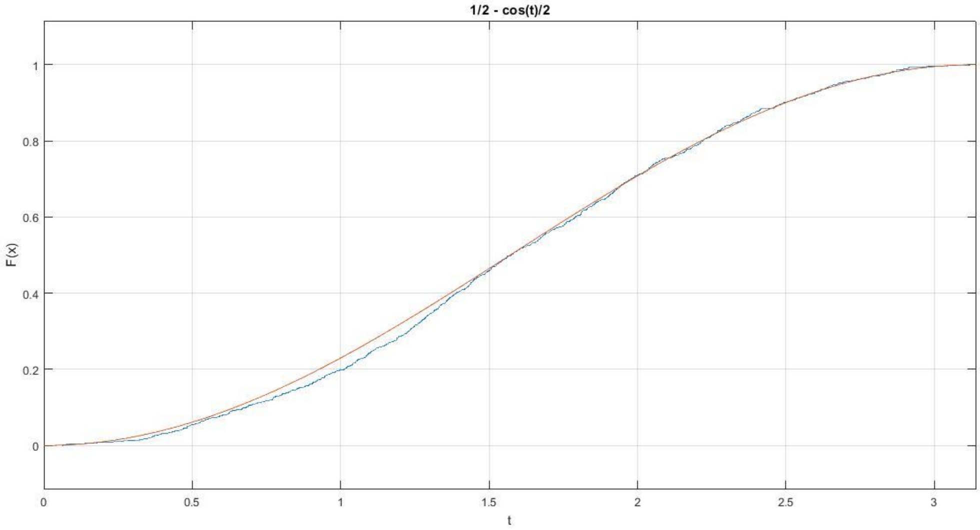

Figure 3 Empirical distribution VS true underlying distribution function n=1000

Part C

To get MSE of the mean estimate, we have

$$MSE(\bar{X}) = E[(\mu - \bar{X})^2] = var(\bar{X}) = \frac{\sigma^2}{n}$$

We already knew that $\sigma^2 = \frac{\pi^2}{4} - 2$. So, for n = 100, $MSE(\bar{X}) = \frac{\frac{\pi^2}{4}-2}{100} = 0.00467.$

Now suppose the population variance was not known, compute the MSE of the mean estimate based on the sample variance, we know that sample variance is

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Xi - \bar{X})^2$$

Then we can use the code blow to get sample mean and S^2. Note that var function in MATLAB is normalised by n-1 so it is already unbiased estimator of σ^2.

Code:

```
n=100; % no of samples
x=zeros(1,n);
for i=1:n
    t=rand(1);
    x(i)=acos(1-2*t); % inverse of distribution function
end
Sample_Mean=mean(x)
Sample_Variance=var(x)
```

Output:

Sample_Mean = 1.5461

Sample_Variance(S^2) = 0.4548

The sample mean is 1.5461 which is very close to population mean π/2 = 1.5754. Also, the $MSE(\bar{X}) = \frac{S^2}{n} = 0.004548$ which is very close to 0.00467.

Part D

Code:

```
N=100; % no of samples
M=50; % no of bootstrap samples
Mean_m= zeros(1,M);
Var_v= zeros(1,M);

for i=1:100
    t=rand(1);
    x(i)=acos(1-2*t);
end

for j=1:50
    X=randsample(x,M,1);
    Mean_m(j)=mean(X);
    Var_v(j)=var(X);
end
Mean_m
Var_v
```

Output:

We get output like this

```
Mean_m =
 Columns 1 through 12
   1.6023    1.6917    1.6397    1.7452    1.6164    1.5838    1.5017    1.6236    1.5736    1.4696    1.3177
1.6525
```

```
 Columns 13 through 24
   1.4139    1.5383    1.5102    1.4816    1.7691    1.5937    1.6641    1.5917    1.6771    1.7653    1.5701
1.6515
 Columns 25 through 36
   1.5781    1.7202    1.4822    1.6304    1.4651    1.6739    1.4720    1.4617    1.6938    1.5787    1.5356
1.6496
 Columns 37 through 48
   1.6507    1.7323    1.6878    1.6795    1.4345    1.6082    1.6120    1.5692    1.5117    1.7601    1.7021
1.6215
 Columns 49 through 50
   1.6476    1.5635

Var_v =
 Columns 1 through 12
   0.3854    0.3404    0.4928    0.5199    0.4794    0.3442    0.2948    0.4338    0.5673    0.4926    0.4271
0.5009
 Columns 13 through 24
   0.3684    0.4142    0.4280    0.4289    0.3760    0.5812    0.3542    0.4092    0.4401    0.6505    0.5315
0.5778
 Columns 25 through 36
   0.5001    0.2719    0.4453    0.5238    0.3555    0.4134    0.4450    0.4478    0.4293    0.7201    0.2829
0.4524
 Columns 37 through 48
   0.5594    0.3677    0.5872    0.3812    0.4672    0.4135    0.3902    0.5461    0.4885    0.4107    0.5631
0.3699
 Columns 49 through 50
   0.5439    0.4372
```

Although the result is various in some aspect we can see that most of them are close to the mathematical results we get in part A.

Part E

Now we already have about M=50 bootstrap samples, every sample has a sample variance, by computing the equation above we can get the estimator of the MSE.

Code:

```
E_F= zeros(1, M);
var_p=var(x);
for j=1:50
    X_F=zeros(1,N);
    X_F=randsample(x,M,1);
    v(j)=var(X_F);
    E_F(j)=(v(j)-var_p).^2;
end
MSE_var= mean(E_F)
```

Output:

```
MSE_var = 0.0049
```