Name: Sourav Bikash

Roll No.:11012338

Submission Date: 22/03/2013 Time:23:59 hrs.

Aim of the Problem:

The problem discusses variation reduction techniques. The first part uses the Monte Carlo estimation technique. Then some standard variation reduction techniques are used:

1) antithetic variables.

2) control variables.

Part I:

This question wants us to calculate the estimate and confidence interval using the Monte Carlo estimation technique.

Implementation using R:

```
#this program returns an estimate of the stated distribution
estimate<-function(n)
{
    s<-0;
    for(i in 1:n)
    {
        u1=runif(1,min=0,max=1);
        y=rexp(1,rate=sqrt(u1));
        s<-s+y;
    }
    s<-(s/n);
    return(s);
}
k<-estimate(100000);
#this gives the confidence interval
estimate<-function(n)
```

```
{
	RN<-NULL;
	RN1<-NULL;
	s<-0;s1<-0;s2<-0;
	for(i in 1:n)
	{
		u1=runif(1,min=0,max=1);
		y=rexp(1,rate=sqrt(u1));
		s<-s+y;
		RN1<-c(RN1,y);
	}
	s1<-s/n;#this is the mean
	RN<-c(RN,s1);
	s2<-(sqrt(var(RN1))/sqrt(n));
	RN<-c(RN,s2);
	return(RN);
}
k<-estimate(100000);
i1<-k[1]-(qnorm(0.025)*k[2]);
i2<-k[1]+(qnorm(0.025)*k[2]);
print(paste(i1));
print(paste(i2));
```

M value	**Estimate**	**Confidence Interval**
100	1.997705	(1.47,2.54)
1000	2.078379	(1.72806,2.16931)
10000	1.98996	(1.904,2.0506)
100000	1.990829	(1.8954,2.02881)

Part II:

Here we have to repeat the same exercise using antithetic variables.

Implementation using R:

```
# the program uses antithetic variables to calculate the estimate
estimate<-function(n)
{
	s<-0;
	for(i in 1:n)
	{
		u1=runif(1,min=0,max=1);
		y1=rexp(1,rate=sqrt(u1));
		y2=rexp(1,rate=sqrt(1-u1));
		s<-s+0.5*(y1+y2);
	}
	s<-(s/n);# the estimate
	return(s);
}
for(i in 2:5)
{
	print(paste(10^i));
	k<-estimate(10^i);
	print(k);
}
#this calculates the confidence intervals
estimate<-function(n)
{
	s<-0;
	RN<-NULL;
	RN1<-NULL;
	for(i in 1:n)
	{
		u1=runif(1,min=0,max=1);
		y1=rexp(1,rate=sqrt(u1));
		y2=rexp(1,rate=sqrt(1-u1));
		RN1<-c(RN1,(0.5*(y1+y2)));
		s<-s+0.5*(y1+y2);
	}

	RN<-c(RN,(s/n));# the estimate
	RN<-c(RN,(var(RN1)/sqrt(n)));
	return(RN);
}
for(i in 2:5)
{
	print(paste(10^i));
	k<-estimate(10^i);
	i1<-k[1]-(qnorm(0.025)*k[2]);
	i2<-k[1]+(qnorm(0.025)*k[2]);
	print(paste(i1));
	print(paste(i2));
```

```
}

#this calculates the variation reduction percentage
estimateredv<-function(n)
{
	s<-0;
	RN<-NULL;
	RN1<-NULL;
	for(i in 1:n)
	{
		u1=runif(1,min=0,max=1);
		y1=rexp(1,rate=sqrt(u1));
		y2=rexp(1,rate=sqrt(1-u1));
		RN<-c(RN,y1);
		RN1<-c(RN1,y2);
		s<-s+0.5*(y1+y2);
	}
	v1<-var(RN);
	v2<-var(RN1);
	cv<-cor(RN,RN1);
	v<-0.25*((v1)+(v2)+2*cv*(sqrt(v1*v2)));
	#return(cv);
	return((1-(v/v1))*100);#return variance reduction percentage
}
for(i in 2:5)
{
	print(paste(10^i));
	k<-estimateredv(10^i);
	print(k);
}
```

M value	Estimate	Confidence Int	Var. Red. %
100	1.72367	(1.143,2.739)	64.79
1000	1.972666	(1.636,1.98)	67.99
10000	2.037286	(1.808,2.25)	46.53
100000	1.991796	(1.942,2.051)	82.1

Part III:
here we have to use control variate technique to calculate the same thing again.

Implementation using R:

```
#this code calculates the estimate value
estimate<-function(n)
{
	s<-0;
	RN<-NULL;
	RN1<-NULL;
	for(i in 1:n)
	{
```

```
        u1<-runif(1,min=0,max=1);
        y<-rexp(1,rate=sqrt(u1));
        RN<-c(RN,y);
        RN1<-c(RN1,sqrt(u1));
    }
    s<-cov(RN,RN1);
    #s1<-var(RN);
    s2<-var(RN1);
    cval<--(s/s2);
    z<-mean(RN)+cval*(mean(RN1)-mean(mean(RN1)));
    return(z);
}
for(i in 2:5)
{
    print(paste(10^i));
    k<-estimate(10^i);
    print(k);
}

#this gives the variance reduction percentage
estimate<-function(n)
{
    s<-0;
    RN<-NULL;
    RN1<-NULL;
    for(i in 1:n)
    {
        u1<-runif(1,min=0,max=1);
        y<-rexp(1,rate=sqrt(u1));
        RN<-c(RN,y);
        RN1<-c(RN1,sqrt(u1));
    }
    s<-cor(RN,RN1);
    #s1<-var(RN);
    s2<-var(RN);

    vaw<-var(RN)-(s*s)/var(RN);
    return(s*s*100);
}
for(i in 2:5)
{
    print(paste(10^i));
    k<-estimate(10^i);
    print(k);
}
```

M value	Estimate	Confidence Int	Var. Red. %
100	2.23	(1.7803,2.146)	24.34
1000	1.872666	(1.9507,2.05)	18.55
10000	2.046286	(1.987,2.0213)	10.24
100000	1.994394	(1.9943,2.00526)	7.73