An analysis using Monte Carlo simulations on the Eye Sight of Humans and Eagles

Julia Dedic and Thomas Yoon

Motivation

The process by which a light or system of waves is spread out or bend as a result passing through/moving around an aperture or object is known as diffraction. Diffraction occurs at the edges of the aperture or object where the light is bent and hence, as a result, the light is found in the geometrical shadow in the image plane of the aperture or object. There are many different classes of diffraction, in particular, a mathematical complex class being Frensel Diffraction. The reason for the complexity is due to the fact that the light sources are hitting the aperture or object at different angles of incidence therefore causing the wave fronts to be either spherical or cylindrical. A particularly interest application of the Frensel diffraction is how diffracted light reacts as it enters the pupil. Comparing the light energy absorbed by a human and an eagle's eye will be of the primary focus in this report. Since Frensel Diffraction applications do not have any analytically solutions, numerical approximations are done through varies means. In this report, the approximation of the difference between the light energy absorbed by a human and an eagle's eye will be done applying Monte Carlo methods. This in part will compare the theoretical eye sight differences between humans and eagles.

Background

Birds such as the eagles are known to have extraordinary abilities to see objects at far distances. In particular, an eagle's eye is the strongest in the animal kingdom. It has been found that eagles have an estimated eyesight that is four to eight times stronger than the average human. This is due to the fact that eagles have a much larger pupil compared to humans; humans pupils being around 2mm in radius compared to about 7.5mm in radius for eagles. Hence, under the same light conditions, eagles are able to see clearly at further distances because of that the fact that they are able to absorb more light energy based on pupil size.

Methodology

A comparison between the light energy absorbed from a humans and eagles eye respectively will be done using a worm as the object to be viewed. With the source of light being the sun, the Frensel Diffraction equation will be used to see how the light energy is absorbed after it is diffracted off of the yellow/orange worm. The amount of light energy absorbed will provide an indication of the eye sight difference between humans and eagles.

The worm will be the aperture, taking the typical size of a worm as 5mm and considering it as a circular object. λ ("lambda" in code) represents the wave length of the visible yellow/orange light from the worm measured in metres. The wave number will be represented as the value of k, which is given by $\frac{(2\pi)}{\lambda}$. Since the aperture is fixed (ie. the worm is not changing throughout the experiment), the aperture factor will be kept constant as 1 ("surface" variable in code). Viewing the worm from different points in your pupil will be considered as the x and y coordinates in the image plane where the image plane is the human and eagle eyes respectively. A radius of 2mm and 7.5mm will be used for the human and eagle eyes respectively. The resulting measurement will be the light energy absorbed from the eyes in volts per metre.

Monte Carlo methods will be discussed and compared to see which methods are able to generate results that agree with the proportional eyesight difference between humans and eagles.To compare methods, the eagle will be placed four times the distance away from the worm than the human (ie. eagle placed at at a distance of 40cm and the human 10cm away from the worm) implying that the method should provide the same light energy absorbed or more in volts per metre for the eagle than the human eye. This would mean that the eagle is able to absorb the same light energy or more as the human at four times the distance implying the eagle has just as clear vision as the human at a much further distance. Since the Frensel diffraction integral involved both a real and complex part, only the real number part will be taken into consideration since the real part pertains to the magnitude of the electric field (ie. the light energy absorbed from the eye) whereas the complex part measures the diffraction (ie. rapidly oscillating wavelengths that don't meet the eye).

The four Monte Carlo methods that will be investigated are: the classical Monte Carlo, auxiliary variable Monte Carlo, Independence Sampler and Metropolis Within-Gibbs.

The integral of interest is:

$$E(x,y,z) = \frac{1}{i\lambda}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E(x',y',0)\frac{e^{ikr}}{r}dx'dy'$$

where $E(x,y,z)$ corresponds to the electric field at the 3 dimensional coordinates, x,y,z. x', y' corresponds to the 2 dimensional coordinate of the aperture, and x, y corresponds to those at the image plane.

z is the shortest distance between the two planes.

r is defined to be $\sqrt{(x-x')^2+(y-y')^2+z^2}$.

k is the wave number, $\frac{2\pi}{\lambda}$ and i is the imaginary unit.

According to Wyant, J., $E(x', y', 0) = 0$ whenever x' and y' are outside the circular aperture of radius R, and 1 elsewhere as the apparatus will stay unmodified throughout the experiment. Thus, the integral can be simplified to $E(x,y,z) = \frac{1}{i\lambda}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \frac{e^{ikr}}{r} dy'dx'$

Using Monte Carlo technique, the integral can be estimated by the following:

$h = \frac{e^{ikr}}{r}$

$X' \sim Unif[-R, R]$

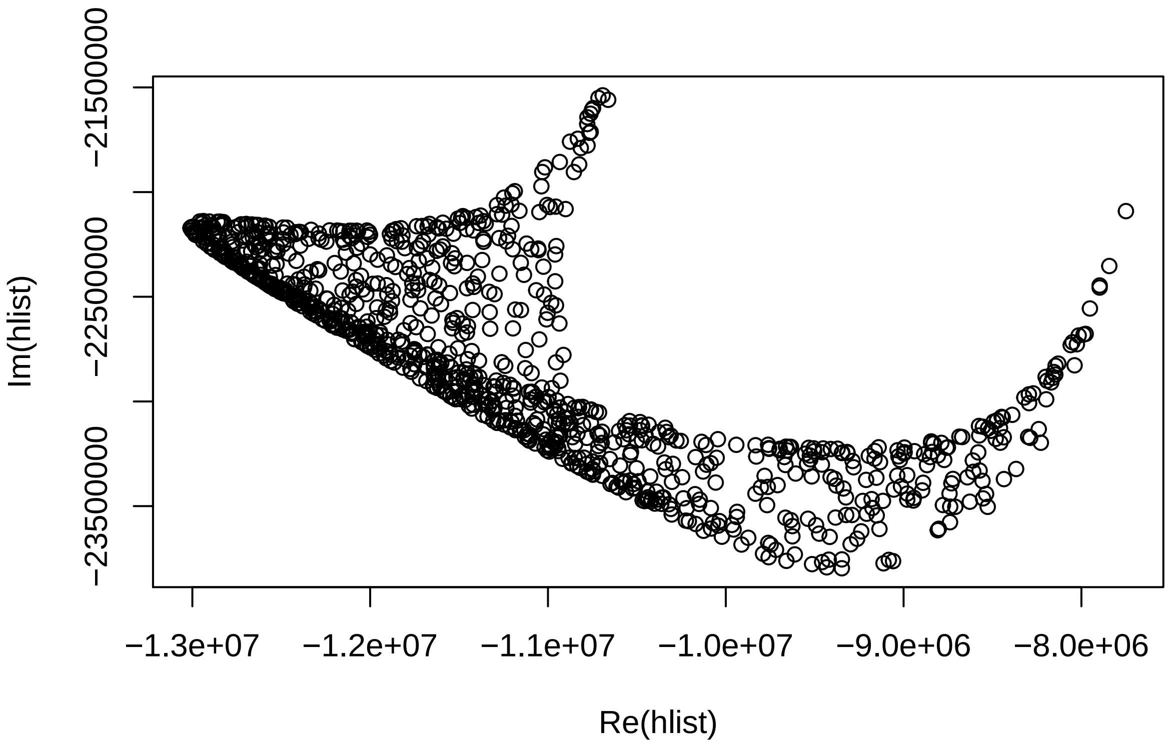
$Y'|X = x \sim Unif[-\sqrt{R^2 - x'^2}, \sqrt{R^2 - x'^2}]$ as aperture with coordinate X', Y' is circular.

$E(x,y,z) = \frac{1}{i\lambda}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \frac{e^{ikr}}{r} dy'dx' = \frac{1}{i\lambda}\int_{-R}^{R}\int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \frac{1}{2R}\frac{1}{2\sqrt{R^2-x^2}} h f_{X'} f_{Y'} dy'dx' \approx \frac{1}{i\lambda}\frac{\sum_{i=1}^{M} h(x',y')}{M}$

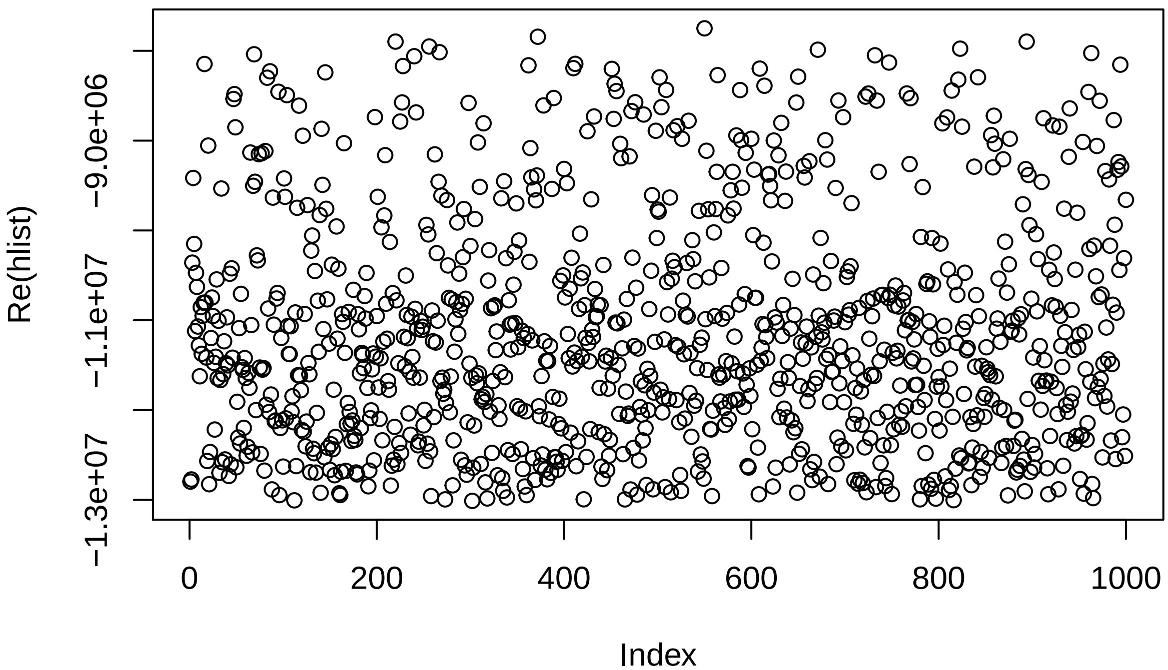
This integral will express the electric field absorbed by the single point in the pupil. We will generate 1000 random points on the area of the pupil for both human and eagle. From those 1000 points, we will look at the average electric field magnitude on both human and eagle eyes. If the average from same number of points in pupil is similar, we can conclude that the sum of total energy received by eyes is similar. Then we can conclude that eagles can see a circular object of 1 cm in diameter at distance of 40 cm just as clearly as humans can at distance of 10 cm.

Results The following is the code for human pupil using basic Monte Carlo approach:

```
## Standard error: 6098.541 Lower CI: -14001861 Upper CI: -13977954

## Standard error: 10304.74 Lower CI: 23335642 Upper CI: 23376037

## Standard error: 13309.88 Lower CI: -25021553 Upper CI: -24969378

## Standard error: 17282.59 Lower CI: 18376673 Upper CI: 18444420

## Standard error: 22703.56 Lower CI: -5717631 Upper CI: -5628633

## Standard error: 14568.45 Lower CI: -8751261 Upper CI: -8694152

## Standard error: 14653.58 Lower CI: 20432247 Upper CI: 20489689

## Standard error: 11422.73 Lower CI: -25369164 Upper CI: -25324387

## Standard error: 21953.81 Lower CI: 21909107 Upper CI: 21995166

## Standard error: 40262.51 Lower CI: -11229285 Upper CI: -11071456
```



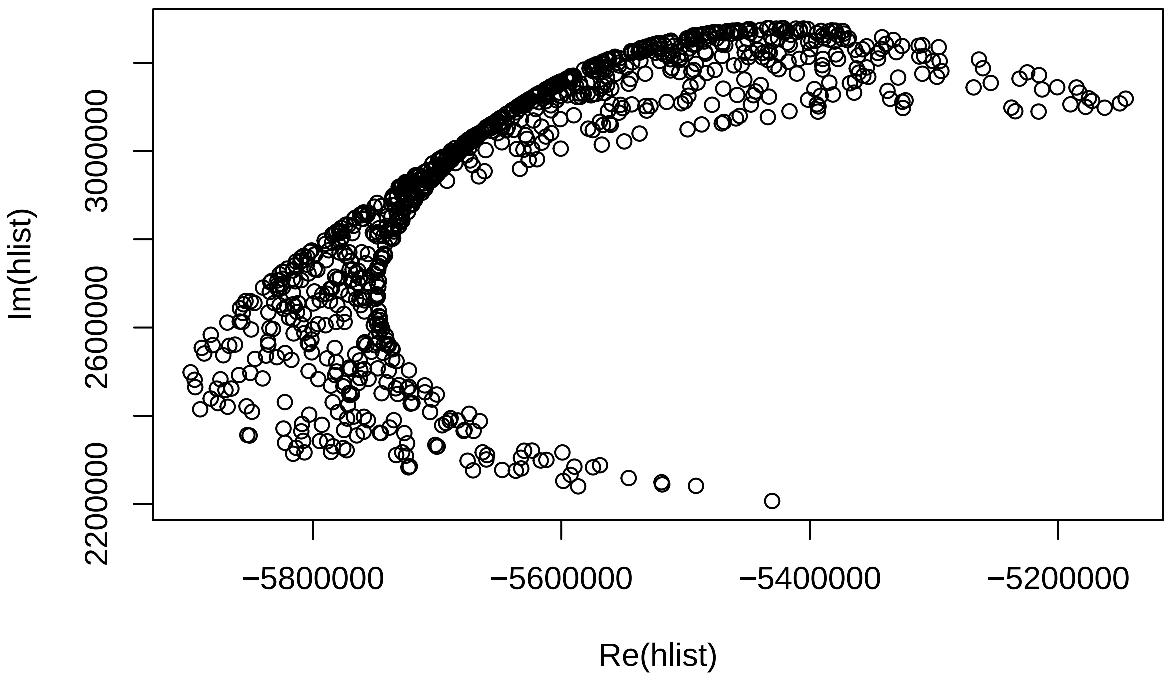
−21500000
−22500000
−23500000
Im(hlist)
−1.3e+07
−1.2e+07
−1.1e+07
−1.0e+07
−9.0e+06
−8.0e+06
Re(hlist)



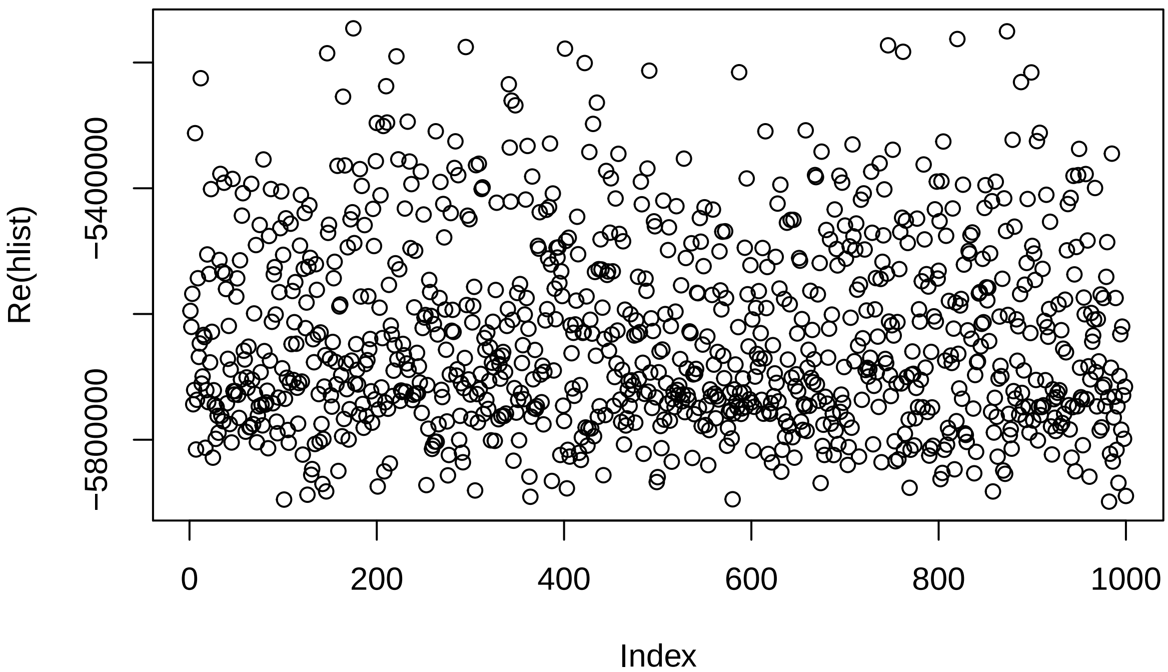

```
## The average energy obtained from the 1000 points of human pupil is
##  15236.62
```

Consider following result for eagles using basic Monte Carlo approach:

```
## Standard error: 2394.022 Lower CI: 4707809 Upper CI: 4717194
## Standard error: 2971.527 Lower CI: -6327979 Upper CI: -6316330
## Standard error: 2909.746 Lower CI: 3816266 Upper CI: 3827673
## Standard error: 3682.806 Lower CI: 1222558 Upper CI: 1236994
## Standard error: 3581.498 Lower CI: -5462278 Upper CI: -5448239
## Standard error: 3056.277 Lower CI: 6088004 Upper CI: 6099985
## Standard error: 4116.702 Lower CI: -2689535 Upper CI: -2673398
## Standard error: 5639.615 Lower CI: -2461811 Upper CI: -2439704
## Standard error: 4451.474 Lower CI: 5937974 Upper CI: 5955424
## Standard error: 4773.288 Lower CI: -5643860 Upper CI: -5625149
```



Im(hlist)
2200000
2600000
3000000
−5800000
−5600000
−5400000
−5200000
Re(hlist)




```
## The average energy obtained from the 1000 points of eagle pupil is
##  23123.49
```

```
## Note the difference is 7886.876
```

This indicates that the sum of "energy" received by a human's eye and an eagle's eye from 1000 points is very similar. Thus, we conclude that eagles can see the same object of $10mm$ in diameter at a distance of $40cm$ just as well as humans at distance of $10cm$.This suggests that the simply Monte Carlo approach seems to agree with the strength difference in eyesight between human's and eagle's. By computing the 95% confidence interval for every hundredth runs with different fixed coordinates for the location on the pupil for each the human's and eagle's eye respectively, we were able to see that the confidence intervals are quite narrow implying that the amount of variation between the volts per metre absorbed by the eye from different coordinates of the worm are quite similar.

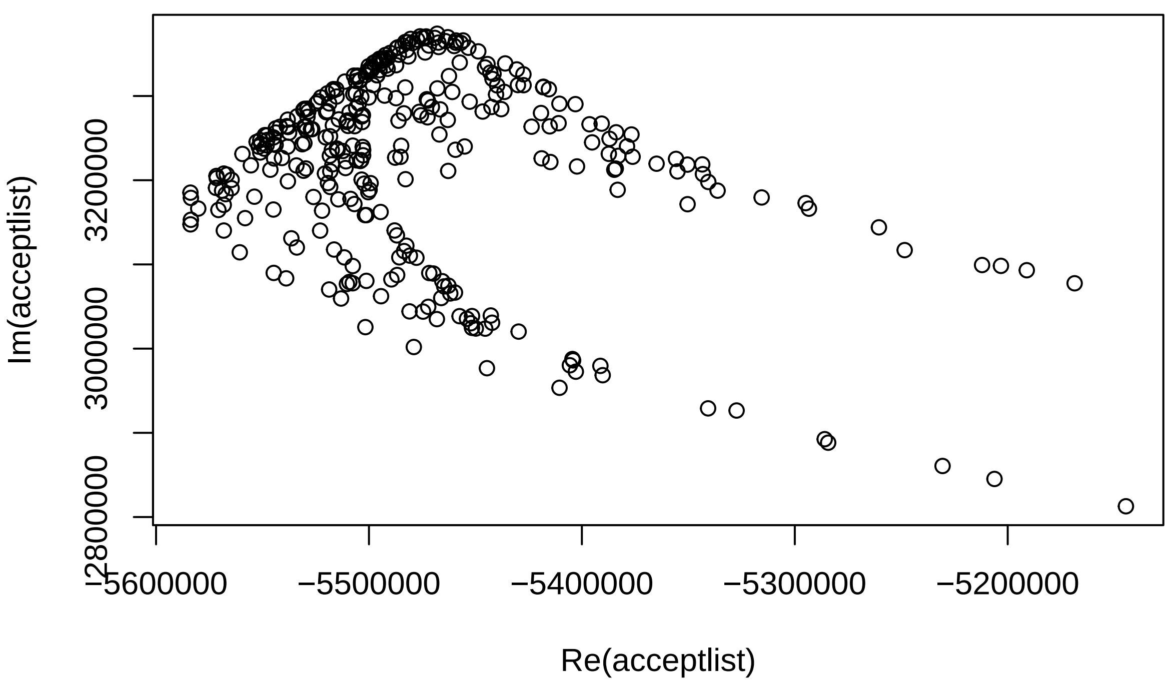
With the same setting, we will use Auxiliary Variable Rejection Sampling Approach. $f_X(x) = \frac{1}{2R} \leq \frac{1}{2R - e^{-15}}$. Similarly, $f_Y(y) = \frac{1}{2\sqrt{R^2 - x^2}} \leq \frac{1}{2\sqrt{R^2 - x^2} - e^{-5}}$

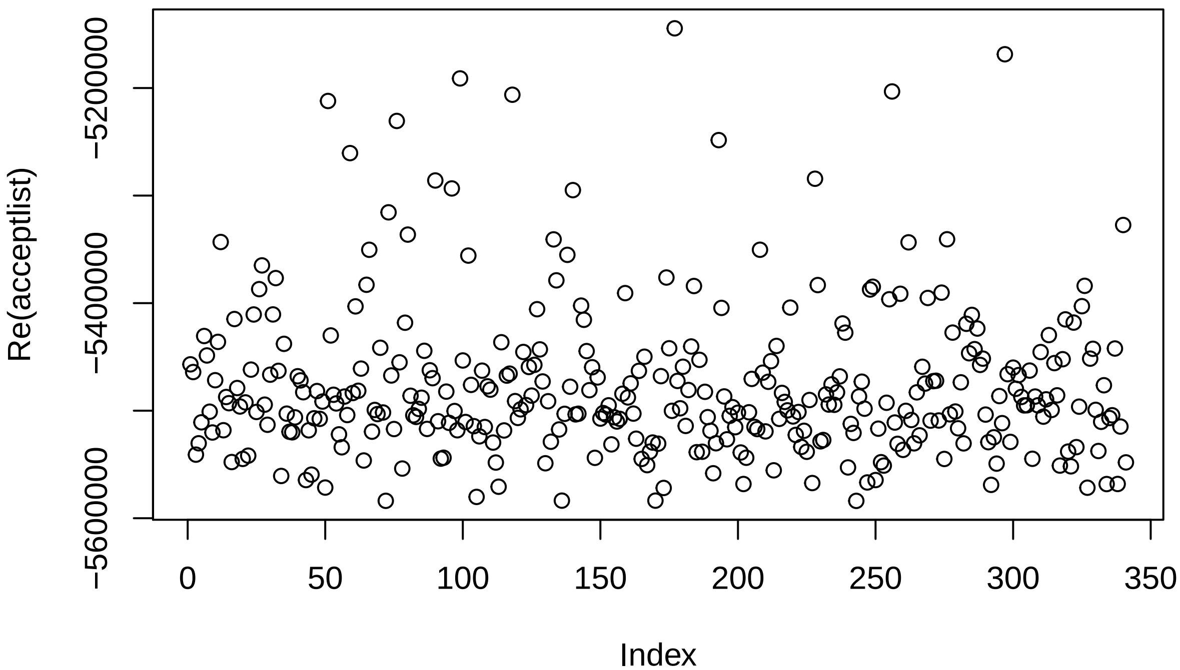
Let $(X', X'') \sim Unif((x', x'')|0 < x'' < \frac{1}{2R - e^{-5}})$.

Let $(Y'|X' = x, Y'') \sim Unif((y', y'')|0 < y'' < f_{Y|X}(y) = \frac{1}{2\sqrt{R^2 - x^2 - e^{-5}}}$

The results below are from the Auxiliary Variable Approach for a human's eye.

```
## Standard error: 4030.616 Lower CI: 4718056 Upper CI: 4733856

## Standard error: 5103.243 Lower CI: -6329476 Upper CI: -6309471

## Standard error: 2675.087 Lower CI: 3718339 Upper CI: 3728825

## Standard error: 2490.619 Lower CI: 1341827 Upper CI: 1351591

## Standard error: 5164.067 Lower CI: -5529814 Upper CI: -5509571

## Standard error: 4523.626 Lower CI: 6025175 Upper CI: 6042908

## Standard error: 3376.241 Lower CI: -2550858 Upper CI: -2537623

## Standard error: 4090.251 Lower CI: -2652158 Upper CI: -2636125

## Standard error: 5831.333 Lower CI: 6044049 Upper CI: 6066908

## Standard error: 3881.107 Lower CI: -5483716 Upper CI: -5468502
```





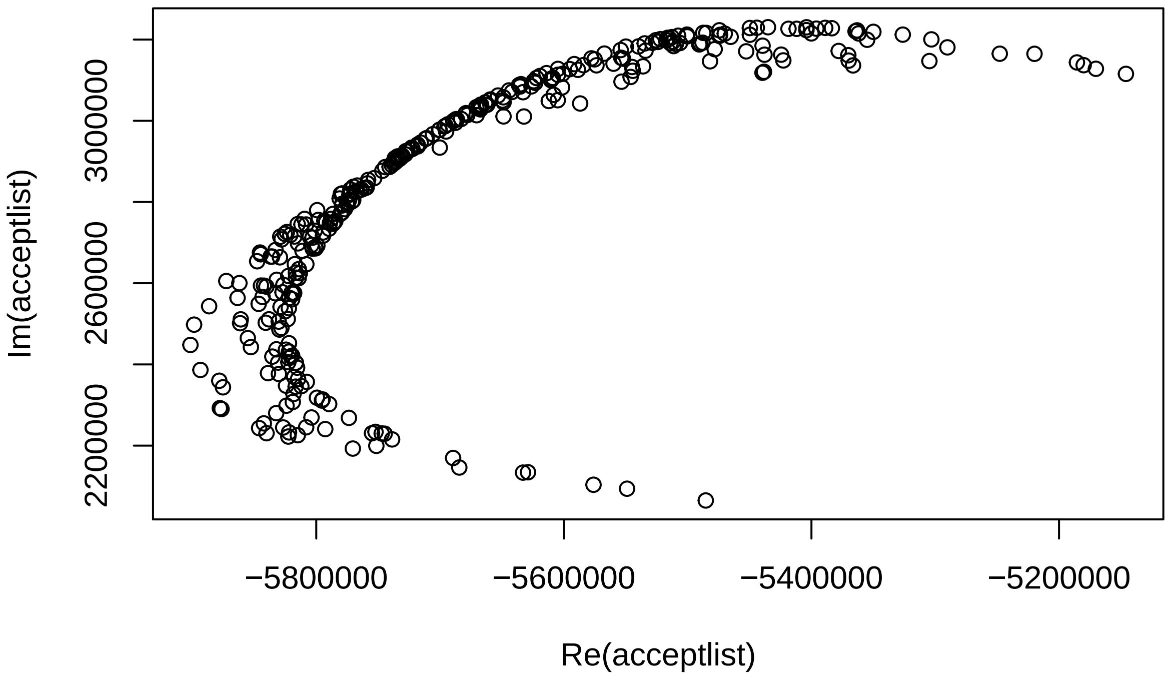

```
## Accepted 341000 number of samples out of 1e+06 total samples

## The average energy obtained from the 100 points of human pupil is
##  21029.62
```

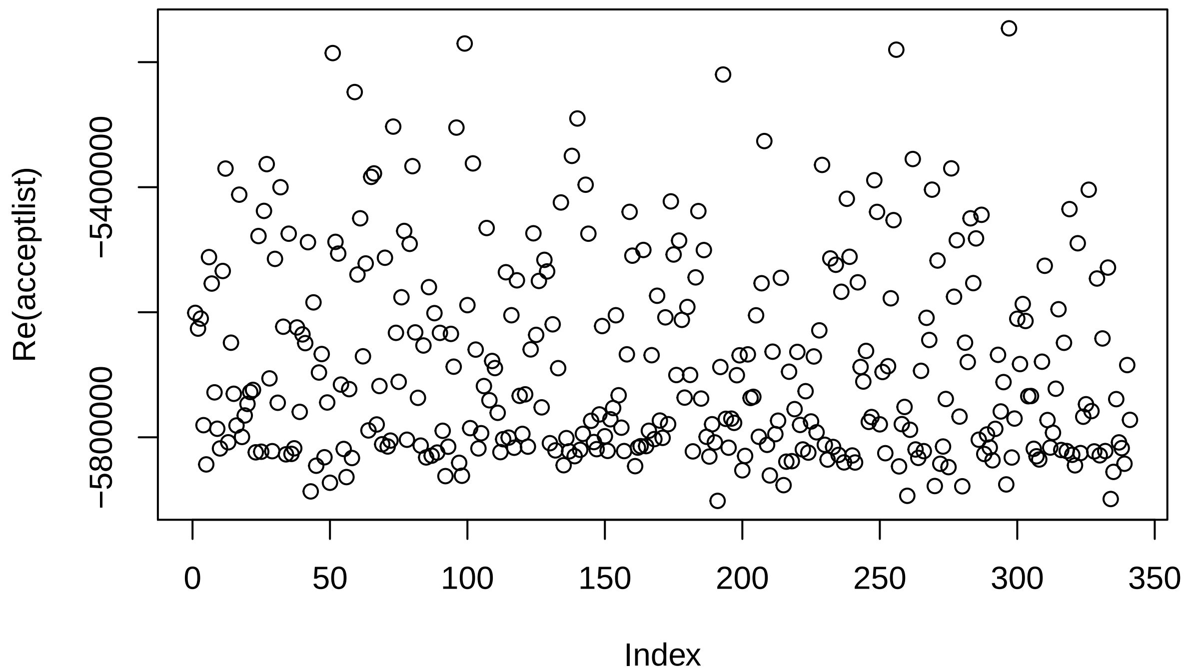
Notice that the obtained estimate is similar to the one obtained from the classical Monte Carlo method for the human eye. It is important to note that the standard error using this method was significantly lower than that from the classical Monte Carlo for the human eye.

Consider the following results from the Auxiliary Variable Approach for an eagle's eye:

```
## Standard error: 4100.995 Lower CI: 4710027 Upper CI: 4726103

## Standard error: 5090.037 Lower CI: -6334772 Upper CI: -6314819

## Standard error: 3468.743 Lower CI: 3752616 Upper CI: 3766213

## Standard error: 5126.984 Lower CI: 1240744 Upper CI: 1260842

## Standard error: 5349.163 Lower CI: -5507280 Upper CI: -5486312

## Standard error: 4817.186 Lower CI: 6054031 Upper CI: 6072915

## Standard error: 11428.75 Lower CI: -2809753 Upper CI: -2764952
```

```
## Standard error: 7788.587 Lower CI: -2528747 Upper CI: -2498216

## Standard error: 6212.011 Lower CI: 6010037 Upper CI: 6034389

## Standard error: 8614.911 Lower CI: -5698878 Upper CI: -5665108
```





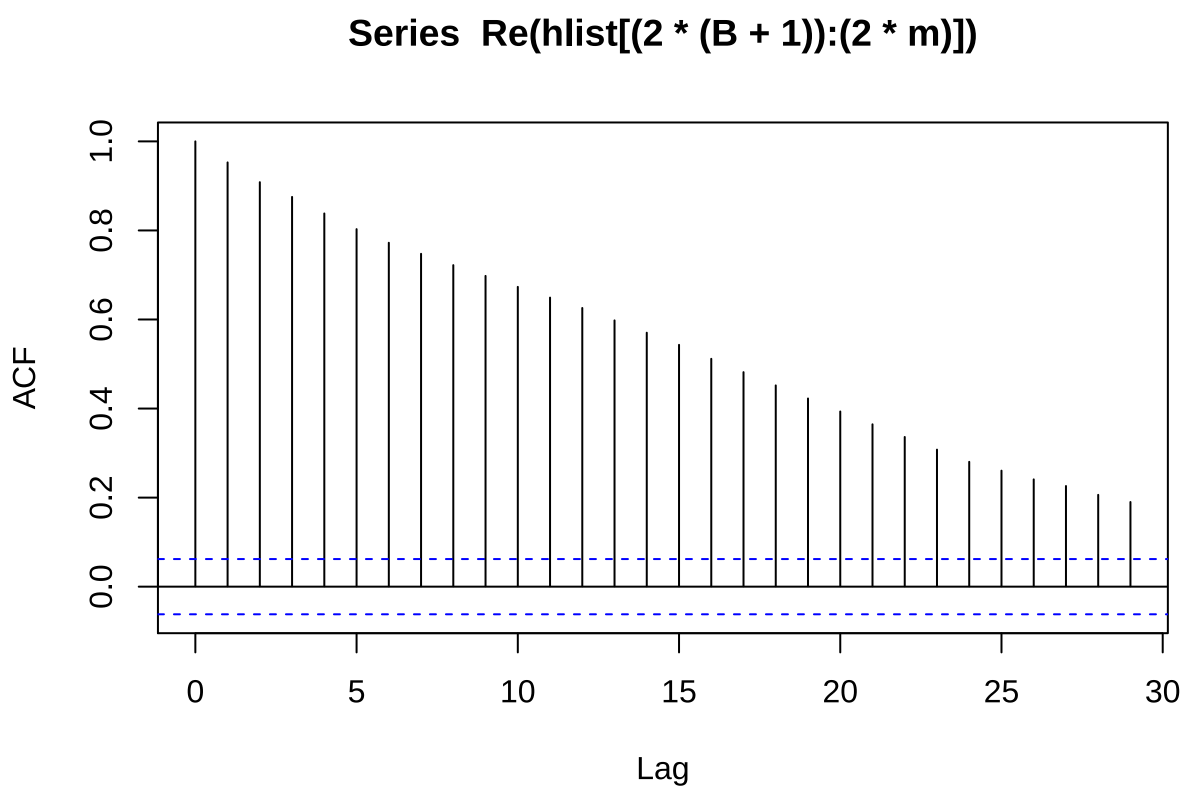

```
## Accepted 341000 number of samples out of 1e+06 total samples

## The average energy obtained from the 1000 points of eagle pupil is
##  19414.31
```

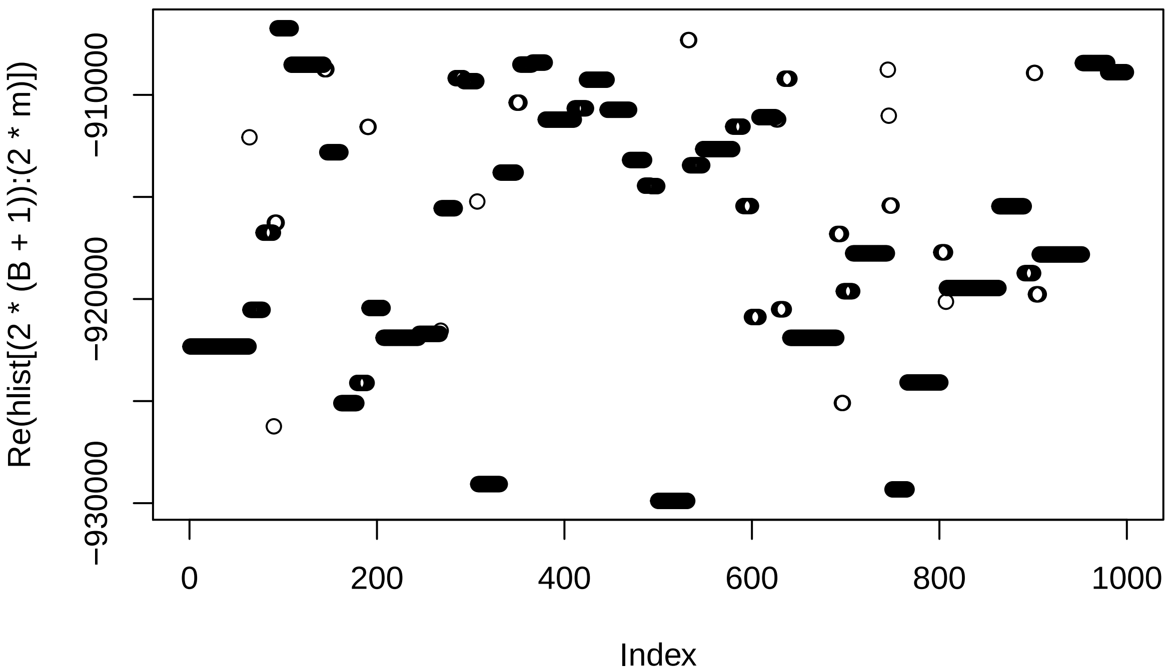
Similar to the results obtained from the comparison between the Auxiliary Variable Rejection Sampling method and the classical Monte Carlo method for a human's eye, the obtained estimate is very close to the obtained estimate for eagles using the classical Monte Carlo method. However, the standard error from the both approaches are quite similar for an eagles eye with the Auxiliary Variable approach producing a slightly lower estimate of standard error.Noticing again that the 95% confidence interval for every hundredth runs with different fixed coordinates for the location on the pupil for each the human's and eagle's eye respectively, we see that the confidence intervals are quite narrow for the auxiliary variation rejection sampling method implying that the amount of variation between the volts per metre absorbed by the eye from different coordinates of the worm are quite similar. This concludes that the two Monte Carlo methods explored obtain quite similar results and these results seem to be accurate based on the eye strength difference among human's and eagle's.

Comparing our results with two MCMC methods, we will be comparing Metropolis within Gibbs and the Independence Sampler.

Using the Metropolis within Gibbs method, we propose that the stationary distribution is $\pi(x', y') \sim MVN(\mathbf{0}, \sigma^2)$ where $x' and y'$ are independent. Setting our surface (i.e. the worm), which in our integral of interest corresponds to $E(x', y', 0)$ to $E(x', y', 0) = 0$ if x' and y' are outside the aperture (worm), and set $E(x', y', 0) = 1$ if x' and y' are inside the aperture.

```
## For the last point in the image plane (i.e., when k=1000):
## iid standard error would be about 197.6132
## Varfact= 32.30238
## True standard error is about 1123.138
## Approximate 95% confidence interval is
## ( -919220.3 , -914817.6 )
```






```
## The average energy obtained from human pupil is 300772.4
```

```
## Probability of acceptance is 0.06748
```

Notice that the obtained estimate is quite different from those obtained using basic Monte Carlo and Auxiliary Rejection method. It is probably because of the fact that the probability of acceptance was only at around 7%. The plot by index and hlist supports the claim. There is no sign of convergence to a particular value of real component of hlist at such low number of iteration. Also, it is known that the variance of MCMC tends to be bigger than basic MC method simply because each proposals are dependent on where the chain was previously. This can be supported by the acf graph because it shows the correlations are quite high even at lag of 30. However, in this example, the true standard error was quite small. The accepted probability is quite lower than the optimal accepted probability of 0.234.

Let's consider the output from same approach on an eagle's pupil.

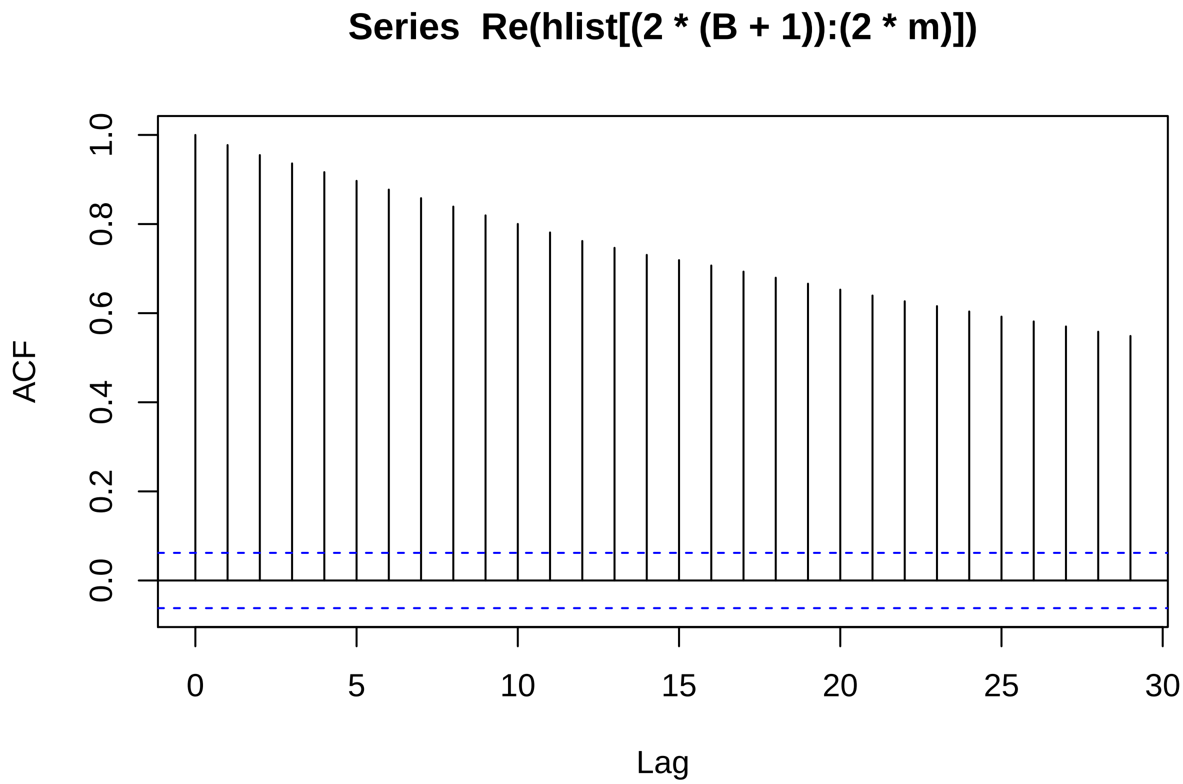
```
## For the last point in the image plane (i.e., when k=1000):
```

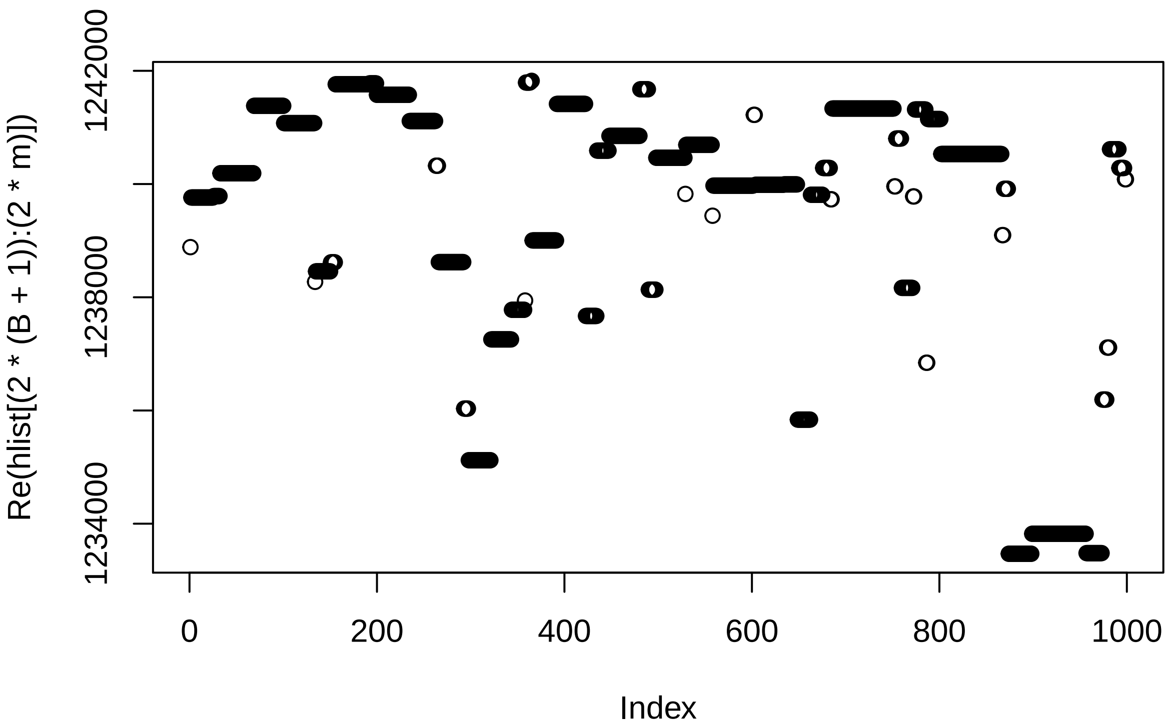
```
## iid standard error would be about 76.88672
```

```
## Varfact= 43.70492
```

```
## True standard error is about 508.2958
```

```
## Approximate 95% confidence interval is
## ( 1238435 , 1240427 )
```






```
## The average energy obtained from Metropolis Within Gibbs on eagle pupil is
##  74997.33
```

```
## Probability of acceptance is 0.067185
```

Similar to the previous case, the plot has shown that the real part of hlist has not yet reached stationary distribution at such low number of iteration. Therefore, the estimate is inaccurate. The Metropolis within Gibbs have failed for both human and eagle pupils. Let's consider the output obtained using independence sampler on human pupil.

Using Independence Sampler, we know that if there exists $\delta > 0$ such that $q(x) \geq \delta\pi(x)$, then the independence sampler is geometrically ergodic. In our case, we have set $\pi = f_X(x)f_Y(y)$ where $X, Y \sim iidN(0,1)$, and $q(X,Y) \sim iidN(0, 0.05^2)$. Then, $\frac{1}{\sqrt{2\pi(0.05^2)}\sqrt{2\pi(0.05^2)}}e^{\frac{-(x^2+y^2)}{2(0.05)^2}} \geq \delta\frac{1}{\sqrt{2\pi}\sqrt{2\pi}}e^{\frac{-(x^2+y^2)}{2}}$. Then by choosing $\delta \leq 400$, $q(x) \geq \delta\pi(x)$. Hence, the independence sampler is geometrically ergodic.

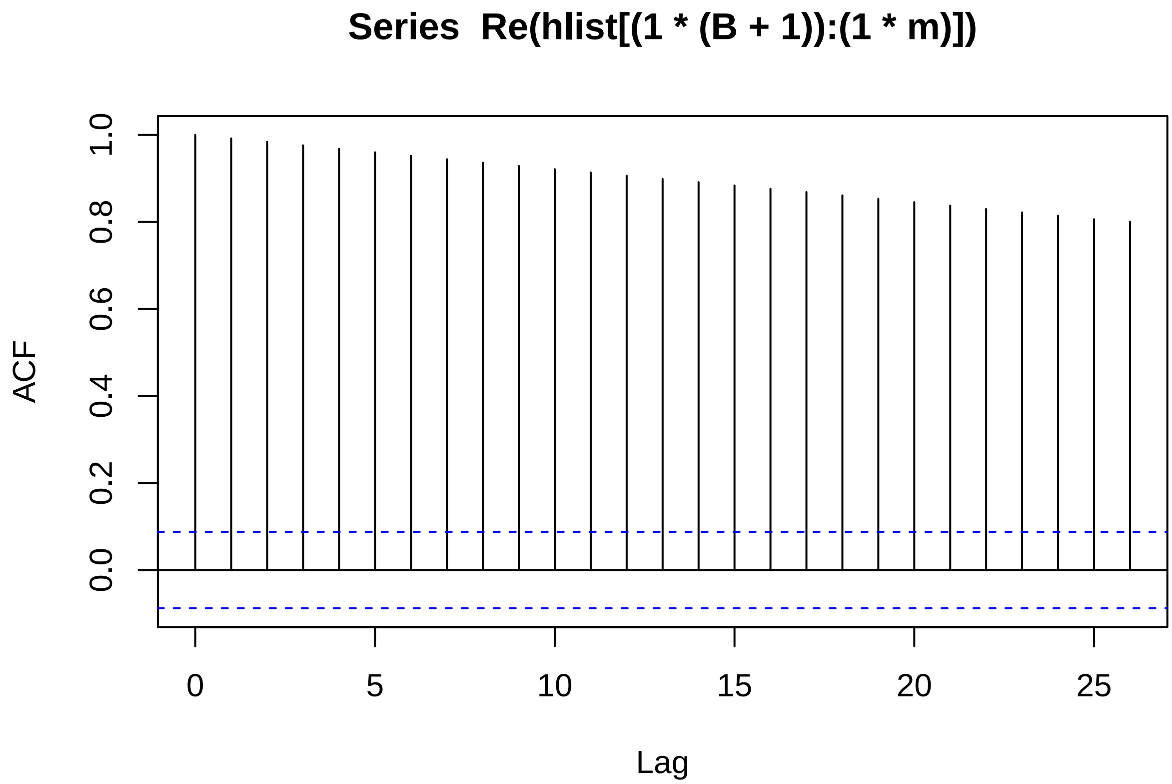
```
## For the last point in the image plane (i.e., when k=1000):
```

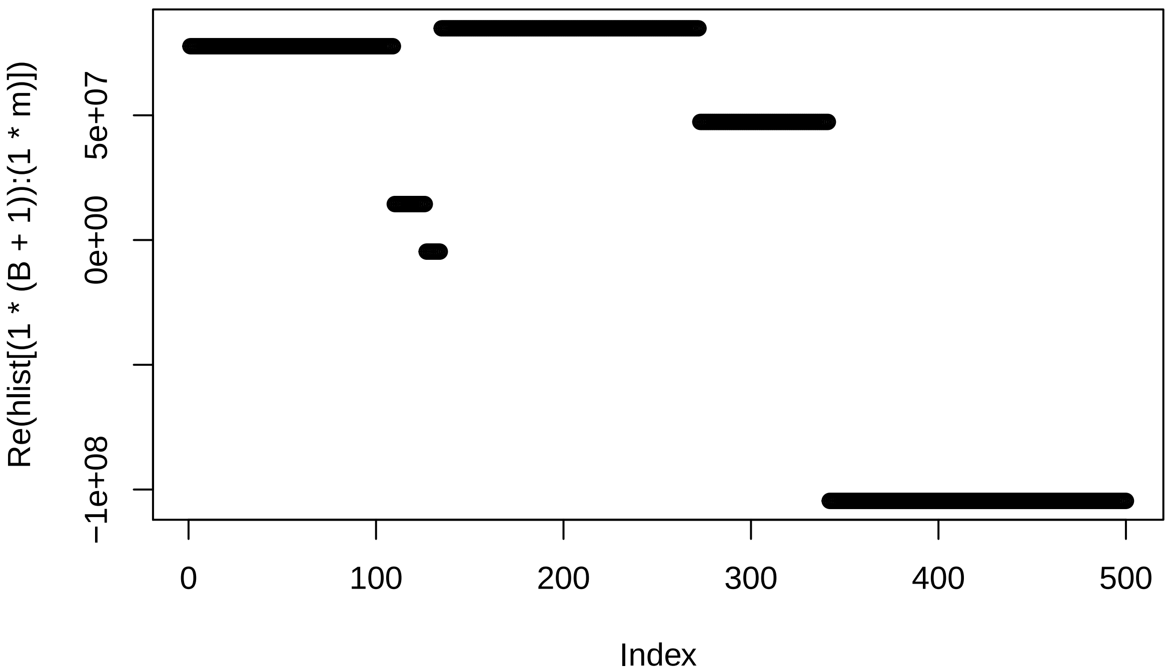
```
## iid standard error would be about 3717833
```

```
## Varfact= 47.54681
```

```
## True standard error is about 25636016
```

```
## Approximate 95% confidence interval is
## ( -36186958 , 64306225 )
```



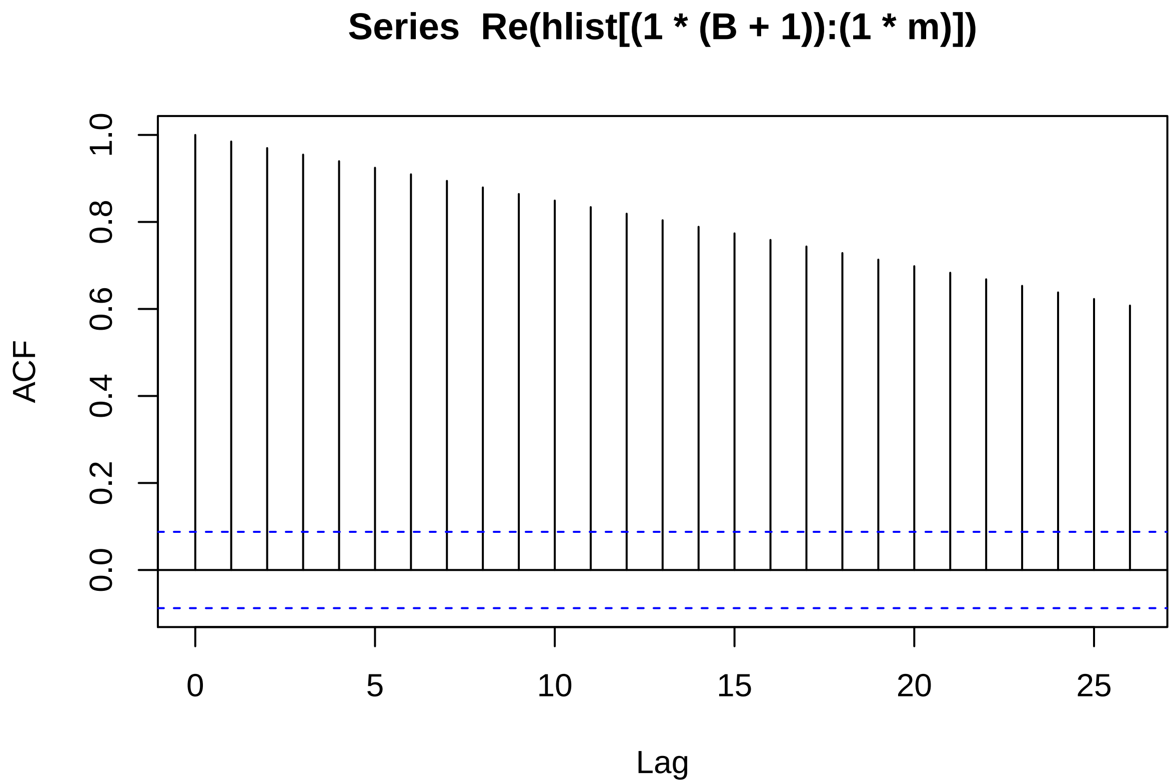


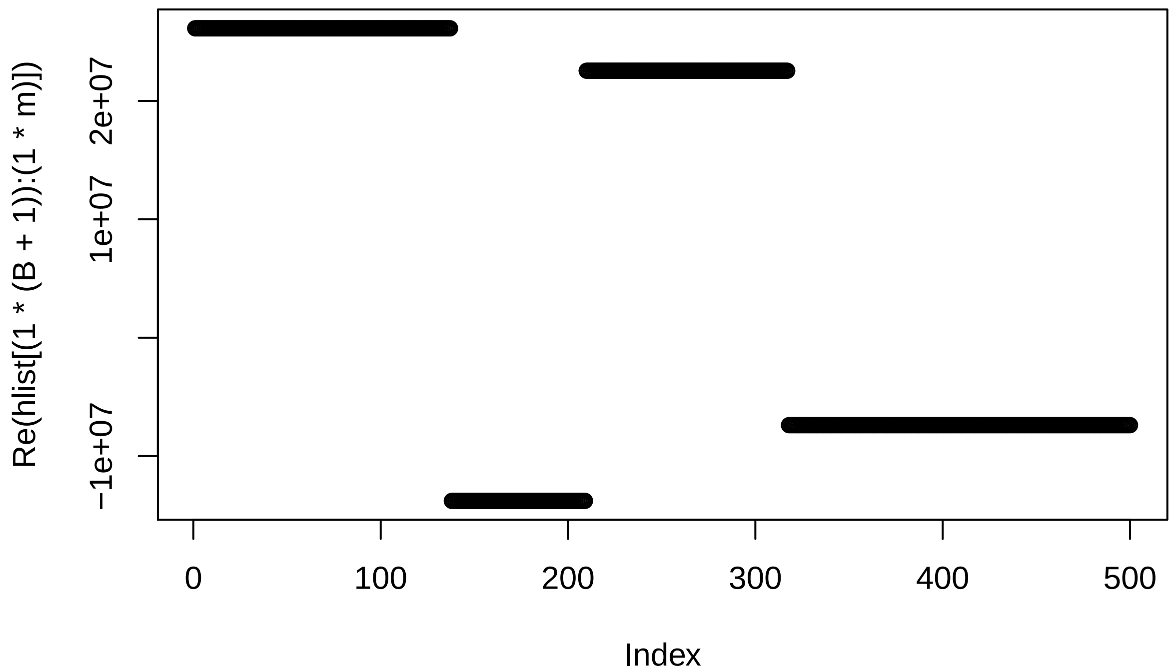

```
## acceptance rate = 0.005078

## mean of h from Independence Sampler is about
##  2020382
```

Looking at the plot of real component of hlist, it is clear that it has not reached a stationary distribution. Unlike Metropolis within Gibbs, this method shows that hlist tends to make a jump then stay there for quite some time before it makes another jump. Looking at the acf graph, it is quite obvious that the hlists are heavily dependent on each other. Hence, varfact is greater than those from Metropolis Within Gibbs methods. Also, the standard error seems to be quite high as well. With same proposal scaling factor of 0.05, it is quite interesting to see the very small acceptance probability. So the variation is generated from very few number of points. It is very likely to go down as more samples are accepted. Let's consider the result obtained from Independence Sampler method on eagle's pupil.

```
## For the last point in the image plane (i.e., when k=1000):
## iid standard error would be about 762957.8
## Varfact= 42.41176
## True standard error is about 4968710
```

```
## Approximate 95% confidence interval is
## ( -2396005 , 17081338 )
```






```
## acceptance rate = 0.005126
```

```
## mean of h from independence sampler from Eagle is about
##  -155252.1
```

Similar to previous model, independence model on eagle's pupil has very low acceptance probability at the same scaling factor. ACF plot for eagle's pupil on independence sampler is also similar to that for human. Looking at the plot of real components of hlist, it has also not reached the stationary distribution and has similar trend of staying in the same position for long time before making a jump. However, this could potentially because of the fact that we accepted very few number of proposals.

Conclusion

Based on the results obtained, we can conclude that the two MC methods worked better than the MCMC methods. The classical Monte Carlo method and the Auxiliary variable sampling method produced results that seem to be consistent with the estimated difference between eyesight for human's and eagle's. However, the same cannot be said about the MCMC method. We obtained results that were quite contradictory. We believe the reason for this is because the number of iterations per fixed point in the eye was only 1000. since our goal was to do the process 1000 for each fixed point in the eye meaning we had a total of 1000*1000 iterations. Hence, due to lack of computational resources available, we were not able to increase the number of iterations for fixed coordinates of the eye ("m" value in the code). This ultimately means that the value of "m" was probably not high enough to achieve stationary distribution. Therefore this means we were probably getting our output (i.e. volts per metre) from a non-stationary distribution meaning it was not very accurate. Given this, our estimates seem to reflect this inaccuracy. Also, notice the i.i.d standard deviations of the estimates from the MCMC methods are quite low in comparison to the MC methods standard deviations. This was probably due to the fact that our Markov chains in both MCMC methods did not move very much and hence, were clustered in one spot. Again, this points to the fact that our chains did not have a chance to achieve stationary distribution as it remained stuck in one position for a long period of time.

Since our surface was so small (i.e. the worm was 1 cm in diameter), we had to lower the standard deviation a lot in order to have even a small number of proposals accepted. After playing around with the the proposal scaling factor, we noticed that we would reject a lot of the samples if the proposal scaling was not small enough, to the point where almost no points would be accepted. This claim is visually supported by all four graphs of the hlist for both human and eagle eyes. Even at proposal scaling factor of 0.05 we used, in all computations with the MCMC methods, the accepted probability was far lower than the optimal rate of 0.234. However, if we reduced the proposal factor to increase the accepted probability, we ran into the problem that the proposed coordinate would very close to the prior one. This has the consequence that the chain no longer becomes irreducible at our given number ("m") iterations.

To further explore the possible uncertainty in our two Monte Carlo methods (classical Monte Carlo and Auxiliary Variable Sampling Method) due to high values in variations, we decided to explore further and did some research. After conducting a lot of research, we were able to conclude the reason for large variation may be based on the physics behind the problem of interest. The behaviour of an electron is not observable because of Heisenberg's uncertainty principle (i.e. the position and the velocity of an object cannot both be measured exactly at the same time). The momentum of an electron cannot be measured at the same time you are trying to measure its position so this creates major variation. As well, the environment is not being controlled for (i.e. we are not in a vacuum) so the electric field (energy absorbed from the eye) is influenced by other electromagnetic fields like the sunlight or earth's natural field which can cause major variation in the diffraction of the light off the worm. This means each run has potential to be very different in terms of the volts per metre that is actually absorbed as light energy by the human and eagle's eye respectively. It seems that all the Monte Carlo methods used seem to follow this reasoning as not only are the estimate between the methods quite different but there is also very large variation between runs meaning, given different fixed coordinates on the worm, the amount of light energy absorbed at different points in the image plane (i.e. the human and eagles eye) varies.

References

1.https://link.springer.com/article/10.3103/S1541308X10040072)

2.http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/fresnelcon.html

3.https://www.youtube.com/watch?v=Q-oQKSLhLKw

Appendix Below is all the source code from all four MC methods explored.

```
lambda <- 600*10^-9 #Wavelength of typical visible light of yellow orange worm
k <- 2*pi/lambda #k= wavenumber
m <- 1000 #Total number of iterations choosing x' and y' at the aperture(worm)
l <- 1000 #Total number of points in the image plane (human/ eagle eye)
radius=5*10^-3 #typical size of worm
humaneyeradius=2*10^(-3)#human eyes have 2mm radius pupil on average
xprimelist <- runif(m,min=-1*radius,max=radius) #list of all points in the aperture
yprimelist <- runif(m,min=-sqrt(radius^2-xprimelist^2),max=sqrt(radius^2-xprimelist^2))
#list of all points in the aperture
surface <- rep(1,m) #fixed aperture, so it can be 1 throughout the experiment
hlist <- rep(0,m)
humaneyesum=rep(0,l)
im <- complex(real=0, imaginary=1) #initializing complex number, i
selist=rep(0,l) #standard error list
lcilist=rep(0,l)#Lower Confidence Interval list
ucilist=rep(0,l)#Upper Confidence Interval list
z <- 100*10^-3 #Shortest distance between object and pupils centre 10cm for humans
for(k in 1:l){
  x=runif(1,min=-humaneyeradius,max=humaneyeradius)#Choosing points in the image plane (eye)
  y=runif(1,min=-sqrt(humaneyeradius^2-x^2),max=sqrt(humaneyeradius^2-x^2))
  r <- sqrt((x-xprimelist)^2+(y-yprimelist)^2+z^2)
```

```r
  #distance between the point in the aperture and the point in the pupil
  for(j in 1:m){
  hlist[j]=as.complex(log(surface[j]))+as.complex(im*k*r[j])-as.complex(log(r[j]))-
    as.complex(log(2*radius)^(-1))-as.complex(log(2*sqrt(radius^2-xprimelist[j]^2))^(-1))
  #function of interest
  }
  hlist= exp(hlist)/(im*lambda) #undoing the log and applying the constant that was in front
  humaneyesum[k]=mean(hlist)
  selist[k]=sd(Re(hlist))/sqrt(m)
  lcilist[k]=mean(Re(hlist))-1.96*selist[k]
  ucilist[k]=mean(Re(hlist))+1.96*selist[k]
  if(k%%100==0){ #Print standard errors and confidence intervals of every 100th iteration
    cat("Standard error:", selist[k], "Lower CI:", lcilist[k], "Upper CI:", ucilist[k],"\n")
  }
  if(k==l){ #For last point on the pupil, let's plot
    plot(hlist) #Just to see how integral is represented on a complex plane
    plot(Re(hlist)) #The real part of the integral of interest
  }
}
cat("The average energy obtained from the 1000 points of human pupil is",
    mean(Re(humaneyesum)))
```

```r
eagleeyeradius=(7.5)*10^(-3)#eagle eye has 7.5 mm radius pupil on average
hlist <- rep(0,m)
eagleeyesum= rep(0,l)
z=0.4 #Shortest distance between object and pupils centre, 40cm for eagles
for(k in 1:l){
  x=runif(1,min=-eagleeyeradius,max=eagleeyeradius) #Choosing particular points on the eagle pupil
  y=runif(1,min=-sqrt(eagleeyeradius^2-x^2),max=sqrt(eagleeyeradius^2-x^2))
  r <- sqrt((x-xprimelist)^2+(y-yprimelist)^2+z^2)
  for(j in 1:m){
  hlist[j]=as.complex(log(surface[j]))+as.complex(im*k*r[j])-as.complex(log(r[j]))-
    as.complex(log(2*radius)^(-1))-as.complex(log(2*sqrt(radius^2-xprimelist[j]^2))^(-1))
```

```
  }
  hlist= exp(hlist)/(im*lambda)
  eagleeyesum[k]=mean(hlist)
  selist[k]=sd(Re(hlist))/sqrt(m)
  lcilist[k]=mean(Re(hlist))-1.96*selist[k]
  ucilist[k]=mean(Re(hlist))+1.96*selist[k]
  if(k%%100==0){
    cat("Standard error:", selist[k], "Lower CI:", lcilist[k], "Upper CI:", ucilist[k],"\n")
  }
  if(k==l){
    plot(hlist)
    plot(Re(hlist))
  }
}
cat("The average energy obtained from the 1000 points of eagle pupil is",
    mean(Re(eagleeyesum)))
cat("Note the difference is",abs(mean(Re(eagleeyesum))-mean(Re(humaneyesum))))
```

```
xprimeprime=runif(m,min=0, max=1/(2*radius-exp(-5))) #Auxilary variable for xprime (points in the apert
yprimeprime=rep(0,m) #Auxilary variable for yprime (poits in the aperture)
for(k in 1:m){
  if(1/(2*sqrt((radius)^2-xprimelist^2)-exp(-5))>0){
    yprimeprime[j]=runif(1,min=0, max=1/(2*sqrt((radius)^2-xprimelist^2)-exp(-5)))
  }
}
numacc=0
humaneyesum=rep(0,l)
for(k in 1:l){
  acceptlist=complex()
  x=runif(1,min=-humaneyeradius,max=humaneyeradius)
  y=runif(1,min=-sqrt(humaneyeradius^2-x^2),max=sqrt(humaneyeradius^2-x^2))
  for(j in 1:m){
    if(xprimeprime[j]< 1/(2*radius) &&
```

```
     yprimeprime[j]< 1/(2*sqrt((radius)^2-xprimelist[j]^2))){
      #Accept based on auxilary rejection method
      r <- sqrt((x-xprimelist[j])^2+(y-yprimelist[j])^2+z^2)
      acceptlist=c(acceptlist, as.complex(log(surface[j]))+as.complex(im*k*r)
                   -as.complex(log(r))-as.complex(log(2*radius)^(-1))-
                     as.complex(log(2*sqrt(radius^2-xprimelist[j]^2))^(-1)))
    }
  }
  acceptlist= exp(acceptlist)/(im*lambda)
  humaneyesum[k]=mean(acceptlist)
  numacc=numacc+length(acceptlist)
  selist[k]=sd(Re(acceptlist))/sqrt(length(acceptlist))
  lcilist[k]=mean(Re(acceptlist))-1.96*selist[k]
  ucilist[k]=mean(Re(acceptlist))+1.96*selist[k]
  if(k%%100==0){
    cat("Standard error:", selist[k], "Lower CI:", lcilist[k], "Upper CI:", ucilist[k],"\n")
  }
  if(k==1){
    plot(acceptlist)
    plot(Re(acceptlist))
  }
}

cat("Accepted", numacc, "number of samples out of", m*l,"total samples")
cat("The average energy obtained from the 100 points of human pupil is",
    mean(Re(humaneyesum)))
```

```
eagleeyesum=rep(0,l)
numacc=0
for(k in 1:l){
  acceptlist=complex()
  x=runif(1,min=-eagleeyeradius,max=eagleeyeradius)
  y=runif(1,min=-sqrt(eagleeyeradius^2-x^2),max=sqrt(eagleeyeradius^2-x^2))
```

```
  for(j in 1:m){
    if(xprimeprime[j]< 1/(2*radius) &&
       yprimeprime[j]< 1/(2*sqrt((radius)^2-xprimelist[j]^2))){
      r <- sqrt((x-xprimelist[j])^2+(y-yprimelist[j])^2+z^2)
      acceptlist=c(acceptlist, as.complex(log(surface[j]))+as.complex(im*k*r)
                   -as.complex(log(r))-as.complex(log(2*radius)^(-1))
                   -as.complex(log(2*sqrt(radius^2-xprimelist[j]^2))^(-1)))
    }
  }
  acceptlist= exp(acceptlist)/(im*lambda)
  eagleeyesum[k]=mean(acceptlist)
  numacc=numacc+length(acceptlist)
  selist[k]=sd(Re(acceptlist))/sqrt(length(acceptlist))
  lcilist[k]=mean(Re(acceptlist))-1.96*selist[k]
  ucilist[k]=mean(Re(acceptlist))+1.96*selist[k]
  if(k%%100==0){
    cat("Standard error:", selist[k], "Lower CI:", lcilist[k], "Upper CI:", ucilist[k],"\n")
  }
  if(k==1){
    plot(acceptlist)
    plot(Re(acceptlist))
  }
}

cat("Accepted", numacc, "number of samples out of", m*l,"total samples")
cat("The average energy obtained from the 1000 points of eagle pupil is",
    mean(Re(eagleeyesum)))
```

```
primelist=c(0,0) #Starting position at the centre of the aperture
varfact=function(xxx){ #Function to calculate varfact
  2*sum(acf(xxx,plot=FALSE)$acf)-1
}
surface=double()
```

```
#Initializing surface
#(going to be 1 if points x', y' are inside the aperture, and 0 if outside)
sigma =0.05 #Proposal scaling factor
pipdf = function(x) {
  #pdf of pi. Notice that the normalizing constant will be cancelled out
  if(x[1]^2+x[2]^2<=radius^2){
    return(dnorm(x[1],mean=0, sd=1)*dnorm(x[2],mean=0, sd=1))
  }
  return(0)
}
numacc=0
B=500 #Burn-in
l=1000
m=1000
surface=rep(1,2*m)
humaneyesum=rep(0,l)
z=0.1
for(k in 1:l){
  x=runif(1,min=-humaneyeradius,max=humaneyeradius)
  y=runif(1,min=-sqrt(humaneyeradius^2-x^2),max=sqrt(humaneyeradius^2-x^2))
  hlist = rep(0,2*m)
  for(j in 1:m){
    for(coord in 1:length(primelist)){
      futureprimelist=primelist #futureprimelist will be containing proposal coordinates
      futureprimelist[coord]=primelist[coord]+sigma*rnorm(1)#propose in a direction "coord"
      U=runif(1)
      alpha= pipdf(futureprimelist)/pipdf(primelist) #for accept/reject
      if(U<alpha){ #If accepted
        numacc=numacc+1
        primelist=futureprimelist #Accept the proposed value
      }
      r <- sqrt((x-primelist[1])^2+(y-primelist[2])^2+z^2)
```

```
      hlist[2*j-2+coord]= as.complex(log(surface[2*j-2+coord]))+as.complex(im*k*r)
        -as.complex(log(r))-as.complex(log(dnorm(primelist[1],0,1)))
        -as.complex(log(dnorm(primelist[2],0,1)))
    }
  }

  hlist= exp(hlist)/(im*lambda)
  humaneyesum[k]=mean(Re(hlist[(2*(B+1)):(2*m)]))
  if(k==l){
    thevarfact=varfact(Re(hlist[(2*(B+1)):(2*m)]))
    se1=sd(Re(hlist[(2*(B+1)):(2*m)]))/sqrt(2*m-2*B)
    se=se1*sqrt(thevarfact)
    cat("For the last point in the image plane (i.e., when k=1000):")
    cat("iid standard error would be about", se1)
    cat("Varfact=", thevarfact)
    cat("True standard error is about", se)
    cat("Approximate 95% confidence interval is (",
        mean(Re(hlist[(2*(B+1)):(2*m)]))-1.96*se,",",
         mean(Re(hlist[(2*(B+1)):(2*m)]))+1.96*se,")")
    acf(Re(hlist[(2*(B+1)):(2*m)]))
    plot(Re(hlist[(2*(B+1)):(2*m)]))
  }
}
cat("The average energy obtained from human pupil is",
    mean(Re(humaneyesum)))
cat("Probability of acceptance is", numacc/(2*m*l))
```

```
primelist=c(0,0)
varfact=function(xxx){
  2*sum(acf(xxx,plot=FALSE)$acf)-1
}
surface=double()
sigma =0.05
```

```
pipdf = function(x) {
  if(x[1]^2+x[2]^2<=radius^2){
    return(dnorm(x[1],mean=0, sd=1)*dnorm(x[2],mean=0, sd=1))
  }
  return(0)
}
numacc=0
B=500
l=100
m=1000
surface=rep(1,2*m)
eagleeyesum=rep(0,l)
z=0.40
for(k in 1:l){
  x=runif(1,min=-eagleeyeradius,max=eagleeyeradius)
  y=runif(1,min=-sqrt(eagleeyeradius^2-x^2),max=sqrt(eagleeyeradius^2-x^2))
  hlist = rep(0,2*m)
  for(j in 1:m){
    for(coord in 1:length(primelist)){
      futureprimelist=primelist
      futureprimelist[coord]=primelist[coord]+sigma*rnorm(1)
      #propose in a direction "coord"
      U=runif(1)
      alpha= pipdf(futureprimelist)/pipdf(primelist) #for accept/reject
      if(U<alpha){
        numacc=numacc+1
        primelist=futureprimelist
      }
      r <- sqrt((x-primelist[1])^2+(y-primelist[2])^2+z^2)
      hlist[2*j-2+coord]= as.complex(log(surface[2*j-2+coord]))+as.complex(im*k*r)
        -as.complex(log(r))-as.complex(log(dnorm(primelist[1],0,1)))
        -as.complex(log(dnorm(primelist[2],0,1)))
```

```
    }
  }
  hlist= exp(hlist)/(im*lambda)
  eagleeyesum[k]=mean(Re(hlist[(2*(B+1)):(2*m)]))
  if(k==l){
    thevarfact=varfact(Re(hlist[(2*(B+1)):(2*m)]))
    # logvarfact=varfact(log(Re(hlist[(2*(B+1)):(2*m)])))
    se1=sd(Re(hlist[(2*(B+1)):(2*m)]))/sqrt(2*m-2*B)
    se=se1*sqrt(thevarfact)
    cat("For the last point in the image plane (i.e., when k=1000):")
    cat("iid standard error would be about", se1)
    cat("Varfact=", thevarfact)
    cat("True standard error is about", se)
    cat("Approximate 95% confidence interval is (", mean(Re(hlist[(2*(B+1)):(2*m)]))-1.96*se,",",
         mean(Re(hlist[(2*(B+1)):(2*m)]))+1.96*se,")")
    acf(Re(hlist[(2*(B+1)):(2*m)]))
    plot(Re(hlist[(2*(B+1)):(2*m)]))
  }
}
# acf(log(Re(hlist)))
cat("The average energy obtained from Metropolis Within Gibbs on eagle pupil is",
    mean(Re(eagleeyesum)))
cat("Probability of acceptance is", numacc/(2*m*l))
```

```
# independence sampler q= c*pi, assuming pi distribution is mvn (0,1)
lambda <- 600*10^(-9)
k <- (2*pi)/lambda
m <- 1000
l <- 1000
B = 500
numacc = 0
xprimelist=0
yprimelist=0
```

```
radius=5*10^-3
humaneyeradius=2*10^(-3)
humaneyesum=rep(0,l)
im <- complex(real=0, imaginary=1)
z <- 100*10^-3
pipdf=function(x,y){ #Pdf of stationary distribution \pi
  if ( x^2+y^2>(5*10^-3)^2 ){
      return(0)
  }
  return(dnorm(x)*dnorm(y))
}
h = function(x,y,r) { #Function of interest
    surface=1
    return(h=as.complex(log(surface))+as.complex(im*k*r)-as.complex(log(r))
           -as.complex(log(dnorm(primelist[1],0,1)))-as.complex(log(dnorm(primelist[2],0,1))))

}
Y = function(x,y){ #Proposal distribution pdf
  return(dnorm(x,sd=0.05)*dnorm(y,sd=0.05))
}
for (j in 1:m){
  x=runif(1,min=-humaneyeradius,max=humaneyeradius)
  y=runif(1,min=-sqrt(humaneyeradius^2-x^2),max=sqrt(humaneyeradius^2-x^2))
    hlist <- rep(0,m)
  for (i in 1:l) {
    xprimeprop=rnorm(1,sd=0.05)#To match the proposal scaling factor of 0.05 of previous example
    yprimeprop=rnorm(1,sd=0.05)
    A=(pipdf(xprimeprop,yprimeprop)*Y(xprimelist,yprimelist))/
      (pipdf(xprimelist, yprimelist)*Y(xprimeprop, yprimeprop))
    U = runif(1);
    if (U < A) {
    # accept proposal
```

```
        xprimelist=xprimeprop
        yprimelist=yprimeprop
        numacc = numacc + 1
    }
     rnew <- sqrt((x-xprimelist)^2+(y-yprimelist)^2+z^2)
     #distance between a point in the aperture and a point in the pupil
   hlist[i]=exp(h(xprimelist,yprimelist,rnew))/(im*lambda)
  }
    humaneyesum[j]=mean(Re(hlist[(B+1):m]))
    if(j==m){
    thevarfact=varfact(Re(hlist[(1*(B+1)):(1*m)]))
    se1=sd(Re(hlist[(1*(B+1)):(1*m)]))/sqrt(1*m-1*B)
    se=se1*sqrt(thevarfact)
    cat("For the last point in the image plane (i.e., when k=1000):")
    cat("iid standard error would be about", se1)
    cat("Varfact=", thevarfact)
    cat("True standard error is about", se)
    cat("Approximate 95% confidence interval is (", mean(Re(hlist[(1*(B+1)):(1*m)]))-1.96*se,",",
        mean(Re(hlist[(1*(B+1)):(1*m)]))+1.96*se,")")
    acf(Re(hlist[(1*(B+1)):(1*m)]))
    plot(Re(hlist[(1*(B+1)):(1*m)]))
  }
}
cat("acceptance rate =", numacc/(l*m), "\n");
cat("mean of h from Independence Sampler is about", mean(Re(humaneyesum)), "\n")
```

```
# independence sampler q= c*pi, assuming pi distribution is mvn (0,1)
lambda <- 600*10^(-9)
k <- (2*pi)/lambda
m <- 1000
l <- 1000
B = 500
numacc = 0
```

```
xprimelist=0
yprimelist=0
radius=5*10^-3
eagleeyeradius=2*10^(-3)
eagleeyesum=rep(0,l)
im <- complex(real=0, imaginary=1)
z <- 400*10^-3
pipdf=function(x,y){
  if ( x^2+y^2>(5*10^-3)^2 ){
       return(0)
  }
  return(dnorm(x)*dnorm(y))
}
h = function(x,y,r) {
    surface=1
    return(h=as.complex(log(surface))+as.complex(im*k*r)-as.complex(log(r))
           -as.complex(log(dnorm(primelist[1],0,1)))
           -as.complex(log(dnorm(primelist[2],0,1))))
}
Y = function(x,y){
  return(dnorm(x,sd=0.05)*dnorm(y,sd=0.05))
}
for (j in 1:m){
  x=runif(1,min=-eagleeyeradius,max=eagleeyeradius)
  y=runif(1,min=-sqrt(eagleeyeradius^2-x^2),max=sqrt(eagleeyeradius^2-x^2))
    hlist <- rep(0,m)
  for (i in 1:l) {
    xprimeprop=rnorm(1,sd=0.05)
    yprimeprop=rnorm(1,sd=0.05)
    A=(pipdf(xprimeprop,yprimeprop)*Y(xprimelist,yprimelist))/
      (pipdf(xprimelist, yprimelist)*Y(xprimeprop, yprimeprop))
    U = runif(1);
```

```r
    if (U < A) {
    # accept proposal
        xprimelist=xprimeprop
        yprimelist=yprimeprop
        numacc = numacc + 1
    }
     rnew <- sqrt((x-xprimelist)^2+(y-yprimelist)^2+z^2)
   hlist[i]=exp(h(xprimelist,yprimelist,rnew))/(im*lambda)
  }
    if(j==m){
    thevarfact=varfact(Re(hlist[(1*(B+1)):(1*m)]))
    se1=sd(Re(hlist[(1*(B+1)):(1*m)]))/sqrt(1*m-1*B)
    se=se1*sqrt(thevarfact)
    cat("For the last point in the image plane (i.e., when k=1000):")
    cat("iid standard error would be about", se1)
    cat("Varfact=", thevarfact)
    cat("True standard error is about", se)
    cat("Approximate 95% confidence interval is (", mean(Re(hlist[(1*(B+1)):(1*m)]))-1.96*se,",",
        mean(Re(hlist[(1*(B+1)):(1*m)]))+1.96*se,")")
    acf(Re(hlist[(1*(B+1)):(1*m)]))
    plot(Re(hlist[(1*(B+1)):(1*m)]))
  }
    eagleeyesum[j]=mean(Re(hlist[(B+1):m]))
}
cat("acceptance rate =", numacc/(l*m), "\n");
cat("mean of h from independence sampler from Eagle is about", mean(Re(eagleeyesum)), "\n")
```