Robin Karlose
40089313

Problem 3

The Beta Function

It is easier to define a generalized function called the incomplete Beta function The incomplete beta function is defined by

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \frac{1}{1+\frac{d_1}{1+\frac{d_2}{1+\frac{d_3}{1+\dots}}}}$$

There are several ways to solve the incomplete Beta function 2 ways would be as follows

First convert the Incomplete Beta function into a continued fraction

Algorithm 1

Evaluation of continued fraction using modified Lentz method

Algorithm 2

Evaluation of continued fraction using Gaussian quadrature rule

I choose **Algorithm 1** for the technical reasons discussed in the next section

Technical Reasons and Explanation of the Algorithm

Advantages

1.No actual integration required

2.Slightly Faster performance

3.It is standalone and does not require computation of the Gamma function

4.It will work for all positive real numbers (not only integers)

Disadvantages

1.It is an approximation and hence not too accurate

2.It can cause overflow or underflow of the floating point variable

3.It is a mathematically derived estimation

Explanation of Algorithm in short

We can express the Beta function in short like this

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \frac{1}{1+\frac{d_1}{1+\frac{d_2}{1+\frac{d_3}{1+\dots}}}}$$

Then we use Lentz algorithm to solve the continued fraction

$$F = 1 + \cfrac{a_1}{1 + \cfrac{a_2}{1 + \cfrac{a_3}{1 + \cfrac{a_4}{1 + \dots}}}}$$

So , now as we have

$D_0 = 0$
$C_0 = 1$
$F_0 = 1$

$D_i = \frac{1}{1 + a_j D_{i-1}}$
$C_i = 1 + \frac{a_j}{C_{i-1}}$
$F_i = F_{i-1} C_i D_i$

We should iterate until C*D becomes really small We also have to make sure that C and D do not actually become 0 (or else it will become undefined). We could implement this by assigning C or D to a very small value if either C or D become too close to zero

Pseudocode

Algorithm 1: Incomplete Beta function using modified Lentz method

```
Function betacf(constant Doub a, constant Doub b, constant Doub x)
{
Variable Declarations
Integer m,m2;
Double  aa,c,d,del,h,qab,qam,qap;

qab = a+b
qap = a +1.0
qam = a-1.0

c=1.0
d=1.0-qab*x/qap

if (mod(d) < Floating Point minimum)
    d=Floating point minimum

d=1.0/d
h=d
```

```
for m = 1 to 9999
{
m2=2*m;
aa=m*(b-m)*x/((qam+m2)*(a+m2))
d=1.0+aa*d

if(mod(d) < Floating Point minimum)
    d= Floating Point minimum

c=1.0+aa/c

if (mod(c) < Floating Point minimum)
    c= Floating Point minimum
d=1.0/d
h *= d*c

aa = -(a+m)*(qab+m)*x/((a+m2)*(qap+m2))

d=1.0+aa*d

if (mod(d) < Floating Point Minimum)
    d=Floating Point Minimum;

c=1.0+aa/c

if (mod(c) < Floating Point Minimum)
    c=Floating Point Minimum

d=1.0/d
del=d*c
h *= del

if (mod(del-1.0) <= EPS)
    break

m=m + 1
}

return h
}
```

References

William H Press,Saul Teukolksky, WIlliam T .Vetterling,Brian P Flannery, The Art of Scientific Computing , (3rd Edition , Cambridge Press), Chapter 6 , Pg 270 - 273

Abramowitz, M., and Stegun, I.A. 1964, Handbook of Mathematical Functions (Washington: National Bureau of Standards); reprinted 1968 (New York: Dover); online at http://www.nr.

com/aands, Chapters 6 and 26.[1]

Pearson, E., and Johnson, N. 1968, Tables of the Incomplete Beta Function (Cambridge, UK: Cambridge University Press).