

TFY4235/FYS8904

Solution problemset 8 Spring 2015



Problem 1.

Here are fortran programs that generate gaussian distributed random numbers using (a) the Box-Müller algorithm (boxm) and (b) the Metropolis algorithm (gmetro):

Listing 1: boxm.f

```

1  program boxm
2  c Box-Mueller algoritme for aa generere Gaussisk-fordelte tall
3  parameter(nmax=250000,nh=100,nm=-nh,hmax=8.,nc=500)
4  dimension nhist(nm:nh),corr(nc),xold(nc)
5  hinv=1./hmax
6  nbin=nh-nm+1
7  nc2=nc/2
8  do i=nm,nh
9  nhist(i)=0
10 enddo
11 rinv=0.5/(2.**31-1.)
12 ibm=1955
13 do i=1,1000
14 ibm=ibm*16807
15 enddo
16 pi2=8*atan(1.)
17 do i=1,nmax
18 ibm=ibm*16807
19 y1=rinv*float(ibm)+0.5
20 ibm=ibm*16807
21 y2=rinv*float(ibm)+0.5
22 x1=sqrt(-2.*log(y1))*cos(pi2*y2)
23 x2=sqrt(-2.*log(y1))*sin(pi2*y2)
24 nh1=(x1*hinv)*nbin
25 nhist(nh1)=nhist(nh1)+1
26 nh2=(x2*hinv)*nbin
27 nhist(nh2)=nhist(nh2)+1
28 do j=2,nc
29 xold(j-1)=xold(j)
30 enddo
31 xold(nc)=x1
32 if(i.gt.nc2) then
33 do j=1,nc
34 corr(j)=corr(j)+xold(j)*xold(nc)
35 enddo
36 endif
37 do j=2,nc
38 xold(j-1)=xold(j)
39 enddo

```

```

40     xold(nc)=x2
41     if(i.gt.nc2) then
42     do j=1,nc
43     corr(j)=corr(j)+xold(j)*xold(nc)
44     enddo
45     endif
46     enddo
47     nhist(0)=nhist(0)/2
48     open(unit=1,file='boxm1.dat',status='unknown')
49     open(unit=2,file='boxm2.dat',status='unknown')
50     do i=nm,nh
51     write(1,*) float(i)*hmax/nbin,nhist(i)
52     enddo
53     close(1)
54     do j=nc,1,-1
55     write(2,*) nc-j,corr(j)/corr(nc)
56     enddo
57     close(2)
58     end

```

Listing 2: gmetro.f

```

1     program gmetro
2     c Metropolis algoritme for aa generere Gaussisk-fordelte tall
3     parameter(nmax=500000,dx=0.5,nh=100,nm=-nh,hmax=8.,nc=500)
4     dimension nhist(nm:nh),corr(nc),xold(nc)
5     hinv=1./hmax
6     nbin=nh-nm+1
7     do i=nm,nh
8     nhist(i)=0
9     enddo
10    do i=1,nc
11    corr(i)=0.
12    enddo
13    rinv=0.5/(2.**31-1.)
14    ridx=rinv*dx
15    ibm=1955
16    do i=1,1000
17    ibm=ibm*16807
18    enddo
19    xo=0.
20    po=exp(-xo**2*0.5)
21    do i=1,nmax
22    ibm=ibm*16807
23    xn=xo+ridx*float(ibm)
24    pn=exp(-xn**2*0.5)
25    if(pn.ge.po) then
26    nhn=(xn*hinv)*nbin
27    nhist(nhn)=nhist(nhn)+1
28    xo=xn
29    po=pn
30    else
31    ibm=ibm*16807

```

```

32     ran=rinv*float(ibm)+0.5
33     if(ran.lt.pn/po) then
34         nhn=(xn*hinv)*nbin
35         nhist(nhn)=nhist(nhn)+1
36         xo=xn
37         po=pn
38     else
39         nhn=(xo*hinv)*nbin
40         nhist(nhn)=nhist(nhn)+1
41     endif
42     endif
43     do j=2,nc
44         xold(j-1)=xold(j)
45     enddo
46     xold(nc)=xo
47     if(i.gt.nc) then
48         do j=1,nc
49             corr(j)=corr(j)+xold(j)*xold(nc)
50         enddo
51     endif
52     enddo
53     nhist(0)=nhist(0)/2
54     open(unit=1,file='gmetro1.dat',status='unknown')
55     open(unit=2,file='gmetro2.dat',status='unknown')
56     do i=nm,nh
57         write(1,*) float(i)*hmax/nbin,nhist(i)
58     enddo
59     close(1)
60     do j=nc,1,-1
61         write(2,*) nc-j,corr(j)/corr(nc)
62     enddo
63     close(2)
64     end

```

A small remark: Note the way that I get the constant $\pi = 3.14\dots$: It is $4 \arctan(1)$. By this construction, one ensures that π is represented with maximum precision in the machine.

Figure 1 shows a plot of the histograms that results from the two programs.

The correlation function $\langle x(t)x(t+\tau) \rangle$ for the random number sequences that result from the two algorithms is shown in figure 2. There are no visible correlations between the gaussian random numbers that result from the Box-Müller algorithm, while the correlations fall off exponentially in the Metropolis data. This is what was expected.

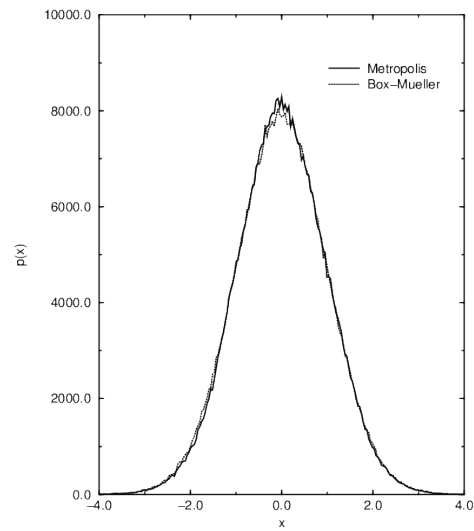


Figure 1: Histograms

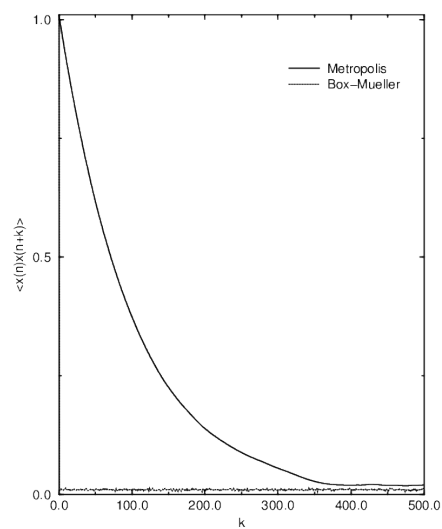


Figure 2: Correlation number