Statistical Methods (Autumn 2017): Homework Problems III

October 6, 2017

1 Problem 1

1.1 Multiplicative Linear Congruential Generator

A multiplicative linear congruential generator (MLCG) generates the following number n_{i+1} from the previous number n_i using the following formula

$$n_{i+1} = (an_i) \mathrm{mod} m, \tag{1}$$

where a (the multiplier) and m (the modulus) are free parameters. In the L'Ecuyer implementation these are chosen as $a = 40692$ and $m = 2147483399$. The choice of these numbers affects the period and therefore the apparent randomness of the generator. The modulo operation (mod) gives the remainder (jakojäännös) of the integer division of the two numbers an_i and m. In C++ and many other languages, % is the modulo operator.

The "previous number" $n_0$used for the first number is called the seed and this too is a matter of choice. The whole sequence of random numbers (assuming certain values for a and m) is fully determined by the seed. A common choice is using the time of day, because this can in many programming languages be easily read from the computer clock and gives a certain sence of randomness to the sequence of numbers. For testing purposes it is also possible to choose a constant seed, which guarantees you always get the same numbers every time you run your program.

Random numbers are generated using the MATLAB code shown in the appendix. The distribution in figure 1 looks uniformly random. The flatness ϕ is defined here as

$$\phi = \mathrm{mean}(|f(i) - \mathrm{mean}(f(i))| / N),$$

where f_i are the contents of the bins of the histogram with N bins. This is just one example of a test you could use. This gives a value from 0 to 1. Close to 0 means flat and 1 peaked.

The parameters for the distribution are $\phi = 0.0822$ (pretty flat), mean $\mu = 0.4990$ (0.50 expected for a uniform distribution from 0 to 1), variance $\sigma^2 = 0.0838$ ($1/12 \approxeq 0.0833$ expected) and the correlation coefficient for two consequtive numbers is $\rho_{n,n+1} = 0.0033$ (almost no correlation). This is given by

$$\rho_{n,n+1} = \frac{\sum_{i=0}^{i=n-1} (n_i - \overline{n})(n_{i+1} - \overline{n}_{i+1})}{(n-1)\sigma_n \sigma_{n+1}} \tag{2}$$

All in all the results agree with a uniform distribution from 0 to 1. The correlation is also shown in figure 2. The hist2.m function is used for plotting the 2D correlation histogram. This can be downloaded from http://www.mathworks.com/matlabcentral/fileexchange/9896-2d-histogram-calculation . Also based on this histogram there is not much correlation.

1.2 Central Limit Theorem

The central limit theorem (CLT) states that the sum of many random numbers following any distribution should follow a Gaussian. The test of the CLT resulted in figure 3. The mean and standard deviation for this histogram is expected to be $\mu = 12 \cdot 0.5 = 6$ and $\sigma^2 = \frac{1}{12} \cdot 12 = 1$. The calculated results are $\mu = 6.0072$ and $\sigma^2 = 1.0482$. Therefore the CLT seems to hold. Also the Gaussian with $\mu = 6$ and $\sigma = 1$ (scaled to have the same integral as the "CLT test histogram") looks very similar (with red circles).

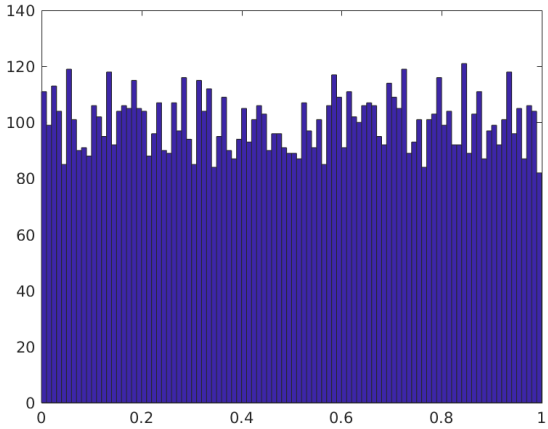

Figure 1: The distribution with 100 bins for the MLCG random number generator.

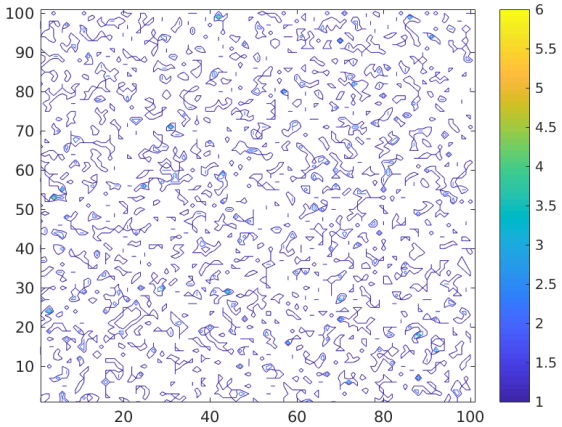

Figure 2: The correlation between two consequtive random numbers.

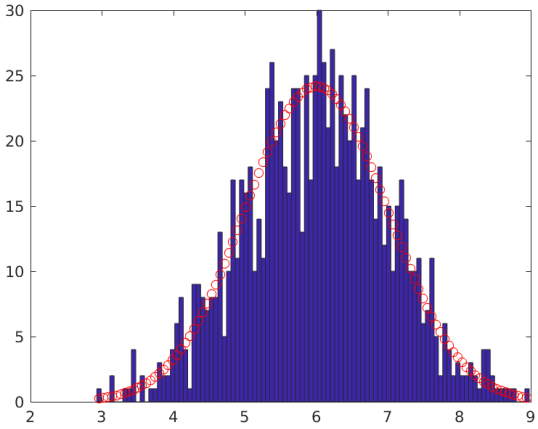

Figure 3: The sum of 12 random numbers. The histogram has 100 bins.

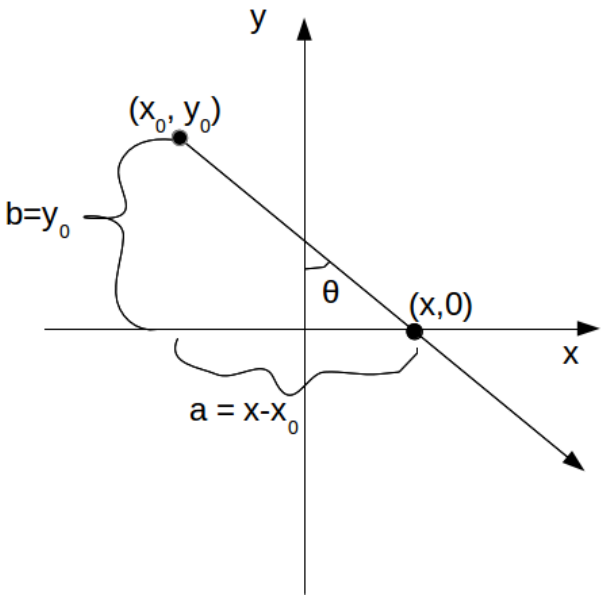

Figure 4: The coordinates and angles used in problem 2.

2 Problem 2

i) This problem is about deriving the Cauchy distribution. The coordinates and angles given are shown in fig. 4. We know from trigonometry that the angle θ can be expressed using the lengths of the sides of the triangle. The relation is the tangent:

$$\tan\theta = \frac{a}{b} = \frac{X - X_0}{Y_0}.$$

Then of course $\theta = \arctan(\frac{X-X_{0)}}{Y_0})$. The probability density function (PDF) $f(X)$ (apart from normalization) of X can be obtained as the derivative $d\theta/dX$. This applies because θ is uniformly distributed in the whole range of angles allowed for the tangent function.

$$f(X) = N\frac{d\theta}{dX} = N\frac{d\arctan\frac{X-X_0}{Y_0}}{dX} = N\frac{1}{Y_0(1+(\frac{X-X_0}{Y_0})^2)}$$

N is inserted for normalization purposes. Next, calculate N, by integrating over the full range of X and setting this total probability to 1.

$$1 = \int_{-\infty}^{\infty} N\frac{1}{Y_0(1+(\frac{X-X_0}{Y_0})^2)}dX = N\pi$$

$$\Longrightarrow N = \frac{1}{\pi}.$$

The PDF is then

$$f(X) = \frac{1}{\pi}\frac{1}{Y_0(1+(\frac{X-X_0}{Y_0})^2)}.$$

The cumulative distribution function (CDF) $F(X)$ is obtained by integrating the PDF from $-\infty$ to X.

$$F(X) = \int_{-\infty}^{X} f(u)du = \frac{1}{\pi}\int_{-\infty}^{X}\frac{1}{Y_0(1+(\frac{u-X_0}{Y_0})^2)}du = \frac{1}{\pi}/_{-\infty}^{X}\arctan(\frac{2u-2X_0}{2Y_0}) = \frac{1}{2}+\frac{1}{\pi}\arctan(\frac{X-X_0}{Y_0})$$

In the inverse transform method a uniform distribution is transformed into the desired (Cauchy in this case) distribution. This is done by using a function $E(r)$, which gives the desired PDF $f(E)$ when evaluated using r values uniformly distributed between 0 and 1. The probability to obtain a value of r in $[r,\ r+dr]$ is $g(r)dr$ and this should be equal to the probability to obtain E in $[E(r),\ E(r)+dE(r)]$, which is $f(E)$. This can be obtained by finding $E(r)$ such that the cumulative distributions $F(E(r))$ and $G(r)$ relate by $F(E(r)) = G(r)$.

$$G(r) = r,\ r = [0,1]$$

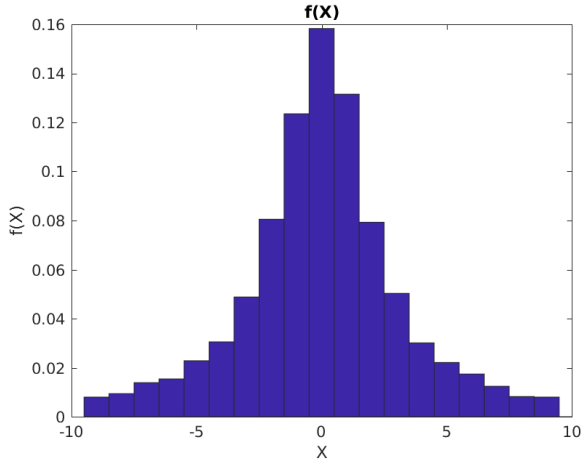

Figure 5: The distribution of X-values.

The generated random numbers r are generated uniformly in the interval $[0, 1]$. Therefore the CDF has the simple form shown above.

$$F(E(r)) = \frac{1}{2} + \frac{1}{\pi}\arctan(\frac{E(r) - X_0}{Y_0}) = \int_{-\infty}^{r} g(r')dr' = r \tag{3}$$

Now let's solve for $E(r)$.

$$(r - \frac{1}{2})\pi = \arctan(\frac{E(r) - X_0}{Y_0}) \tag{4}$$

$$E(r) = Y_0(\tan((r - \frac{1}{2})\pi) + X_0) \tag{5}$$

This gives values of energy distributed like the Cauchy distribution we started out with when evaluated using $r = [0, 1]$.

ii) See fig. 5.

iii) Expected: $\mu = X_0 = 0$ and HWFM $2Y_0 = 2 \cdot 2 = 2.355\sigma \rightarrow \sigma \approx 1.70$. This relation between HWFM and σ actually applies for the Gaussian distribution only and σ does not really exist for the Cauchy distribution. Also the mean is not defined for the Cauchy distribution, but the median is this X_0.

The actual values obtained from the fit depend heavily on the choice of the fitting region. In $X = [-1.5,\ 1.5]$ I obtained $\mu \approx 0.07$ and $\sigma \approx 1.52$. These almost agree with the expected values. If a (much) larger region is chosen the σ will start to diverge.

A MATLAB code for ex. 1 Ex1.m

```
format compact;
uniformnumbers = zeros(10000,1);
seed = 1235644;

for i=1:10000
    [uniformnumbers(i),seed]=MyMLCG(seed);
end

histo = hist(uniformnumbers,100);
hist(uniformnumbers,100);
print -dpng 'Uniform.png'
disp('Flatness: ')
mean(abs(histo(:)-mean(histo))/length(histo))
disp('The mean is: ')
mean(uniformnumbers)
disp('The variance is: ')
var(uniformnumbers)
```

```
firsthalf = zeros(5000,1);
secondhalf = zeros(5000,1);

for i=1:5000
    firsthalf(i)=uniformnumbers(i*2-1);
    secondhalf(i)=uniformnumbers(i*2);
end

correlationhisto=hist2(firsthalf,secondhalf,[0:0.01:1], [0:0.01:1]);
contour(correlationhisto);
colorbar
print -dpng 'CorrelationHisto.png'


disp('Correlation coefficient: ')
corr(firsthalf,secondhalf)

testclt = zeros(12,1000);

for i=1:12
    for j=1:1000
        [testclt(i,j),seed]=MyMLCG(seed);
    end
end

[y,x]=hist(sum(testclt(:,:)),100);
hist(sum(testclt(:,:)),100);
disp('Mean for the CLT histogram is: ')
mean(sum(testclt(:,:)))
disp('Variance for the CLT histogram is: ')
var(sum(testclt(:,:)))
%plot(x,y,'.');
hold on
gauss = normpdf(x,6,1);
plot(x,gauss*trapz(y)/trapz(gauss),'or')
print -dpng 'CLT.png'
hold off
```

B MATLAB code for random number generator for ex. 1 MyMLCG.m

```
function [g,s] = MyMLCG(u)
%MYMLCG Function that implements a MLCG random number generator
%   This function takes one imput parameter, which is the seed for the
%   generator. The output parameters are the uniformely distributed random
%   number and the seed for the next random number.

% Define parameters
a = 16807;
m = 2147483647;
c = 0; % This has to be zero for this to be an MLCG

% Calculate random number
s = mod((a*u+c),m);

% Normalize to 0..1
g = s/m;

end
```

C MATLAB code for Lorentz uniform sampler for ex. 2 Ex2.m

```
%Create 10000 uniformly distributed random numbers r = [0..1]
r = rand(10000,1);
%Use the transformation that was derived in part i
y_0 = 2;
x_0 = 0;
X = y_0*(tan((r-1/2)*pi)+x_0);

%Choose binning for histogram
bins = [-9.5:1:9.5]';
centers = bins + 0.5;
%Plot histogram
N = histc(X, bins);
bar(bins,N/10000,'histc');
xlim([-10,10]);
xlabel('X');
ylabel('f(X)');
title('f(X)');
print -dpng 'Ex3ii.png'
%Fit central part
myfit = fit(centers(9:11),N(9:11),'gauss1')
widefit = fit(centers(4:16),N(4:16),'gauss1')
ci = confint(widefit,0.95)
mymean = myfit.b1
mysigma = myfit.c1/sqrt(2)
mymean = widefit.b1
mysigma = widefit.c1/sqrt(2)
```