

المستوي الرابع-شعبة الحاسب الالى- (نظام ساعات معتمدة)
كلية العلوم

الفصل الدراسي الاول 2014- 2015 م
تاريخ الامتحان: 28 / 12 / 2014

نموذج اجابة – ورقة كاملة

المادة: حسابات علمية (1)

اسم استاذ المادة: الدكتور/ عبدالحميد محمد عبدالحميد –
جامعة بنها – كلية العلوم – قسم الرياضيات



Department of Mathematics

Final Exam

Time: 2 Hours

Fac. of Science, Benha Univ. SCIENTIFIC COMPUTING (451 M.S.)

28 December 2014

Please answer all the following questions. Total Marks = 48 points:-

Question 1:

- 1) What are the difference between true random numbers and pseudo random numbers?
- 2) What are the properties that a good random number generator should satisfy?

Question 2:

- 1) State Hull and Dobell theorem of general linear congruential generator.
- 2) Consider the linear congruential generator in the form:

$$x_{i+1} = ax_i + c \pmod{m},$$

- a) Generate the random numbers for this generator if we take the seed number $x_0 = 4$, and select the constants as $a = 7$, $c = 4$ and $m = 15$.
- b) What is the period of this generator?
- c) Is this generator has full period? Explain your answer?

Question 3:

- 1) Consider the multiplicative linear congruential generator in the form:

$$x_{i+1} = 5x_i \pmod{2^5}, \quad x_0 = 1,$$

- a) What is the most period of this generator has? Explain your answer?
- b) Generate the random numbers for this generator.
- 2) Explain the combining generators for random numbers and write a Matlab program for this case.

Question 4:

- 1) What is Monte Carlo simulation and how does it work?
- 2) Describe how to use Monte-Carlo method to calculate π .
- 3) Write a Monte Carlo program using Matlab that estimates the radioactive decay equation.

ANSWER MODEL

Question 1:

1- True random numbers are numbers generated by a process whose outcome is unpredictable and which cannot be sequentially reliably reproduced. While, the pseudo random numbers are numbers that appear random, but are obtained in a deterministic, repeatable and predictable manner.

2- A good random generator should satisfy the following properties:

- Uniformity: The numbers generated appear to be distributed uniformly on $(0, 1)$.
- Independence: The numbers generated show no correlation with each other.
- Replication: The numbers should be replicable (e.g., for debugging or comparison of different systems).
- Cycle length: It should take long before numbers start to repeat.
- Speed: The generator should be fast.
- Memory storage: The generator should not require a lot of storage.

Question 2:

1- Hull and Dobell theorem:

The linear congruential generator (LCG) has full period if and only if the following three conditions hold:

- i)- if 4 divides $m \Rightarrow 4$ divides $a - 1$.
- ii)- The only positive integer that exactly divides m and c is 1, i.e., m and c are relatively prime.
- iii)- If q is a prime number that divides m then it divides $a - 1$.

2-

a)

$$x_{i+1} = 7x_i + 4 \pmod{15},$$

$$x(1) = (7 \cdot 4 + 4) \pmod{15} = 2$$

$$x(2) = (7 \cdot 2 + 4) \pmod{15} = 3$$

$$x(3) = (7 \cdot 3 + 4) \pmod{15} = 10$$

$$x(4) = (7 \cdot 10 + 4) \pmod{15} = 14$$

$$x(5) = (7 \cdot 14 + 4) \pmod{15} = 12$$

$x(6) = (7 \cdot 12 + 4) \bmod 15 = 13$
 $x(7) = (7 \cdot 13 + 4) \bmod 15 = 5$
 $x(8) = (7 \cdot 5 + 4) \bmod 15 = 9$
 $x(9) = (7 \cdot 9 + 4) \bmod 15 = 7$
 $x(10) = (7 \cdot 7 + 4) \bmod 15 = 8$
 $x(12) = (7 \cdot 8 + 4) \bmod 15 = 0$
 $x(13) = (7 \cdot 0 + 4) \bmod 15 = 4$
 $x(14) = (7 \cdot 4 + 4) \bmod 15 = 2 \quad \text{-----> stop}$

The random numbers are:

$R = x/m = x/15$

=

0.1333

0.2000

0.6667

0.9333

0.8000

0.8667

0.3333

0.6000

0.4667

0.5333

0

0.2667

0.1333

b) The period of this generator is 12.

c) This generator has not full period, since it has cycle length = 12. Also, using the theorem of Hull and Dobell, it is obvious that the third condition is not satisfied because if $q = 5$, then $5 \nmid 15$ while 5 does not divide $6 = a - 1$.

Question 3:

$$x_{i+1} = 5x_i \pmod{2^5}, \quad x_0 = 1.$$

a) According to the following theorem:

$$x_{i+1} = ax_i \pmod{2^n},$$

had period at most 2^{n-2} . This can be achieved whenever x_0 is odd and $a = 8k + 3$ or $a = 8k + 5$, for some k .

Since $x_0 = 1$ is odd and $a = 5 = 8 * 0 + 5$, then this generator has period equal to $2^{n-2} = 2^3 = 8$.

b)

$$\begin{aligned}x(1) &= 5*1 \pmod{2^5} = 5 \\x(2) &= 5*5 \pmod{2^5} = 25 \\x(3) &= 5*25 \pmod{2^5} = 29 \\x(4) &= 5*29 \pmod{2^5} = 17 \\x(5) &= 5*17 \pmod{2^5} = 21 \\x(6) &= 5*21 \pmod{2^5} = 9 \\x(7) &= 5*9 \pmod{2^5} = 13 \\x(8) &= 5*13 \pmod{2^5} = 1 \\x(9) &= 5*1 \pmod{2^5} = 5\end{aligned}$$

The random numbers are:

$$R = x/m = x/2^5$$

=

0.1563

0.7813

0.9063

0.5313

0.6563

0.2813

0.4063

0.0313

0.1563

c) Combining Generators

One way to enhance the linear congruential generators is to combine two or more LCGs through summation. A combination of generators allows to generate much longer sequences than any sequence generated by one of its components. Assume we have J generators, the j th having parameters a_j, m_j :

$$x_{j,i+1} = a_j x_{j,i} \bmod m_j, \quad u_{j,i+1} = x_{j,i+1}/m_j$$

As a simple example consider two multiplicative generators to generate the sequences x_1, x_2 with moduli m_1 and m_2 where $m_1 > m_2$. This sequences can now be added or subtracted and we are able to obtain an enhanced sequence of uniformly distributed pseudorandom numbers with the following generator:

$$x_i = (x_{1,i} + x_{2,i}) \bmod m_1, \quad u_i = x_i/m_1$$

Another possibility to enhance the basic linear congruential generator uses a high order recursion called multiple recursive generator(MRG):

$$x_i = (a_1 x_{i-1} + a_2 x_{i-2} + \dots + a_k x_{i-k}) \bmod m, \quad u_i = x_i/m$$

The seed for MRG is given by the initial values for $x_{k-1}, x_{k-2}, \dots, x_0$.

```
function u = CombinedGenerator(a,m,seed,n)
nbrGen = length(a);
x = nan(n,nbrGen);
% Assign seed values
for j = 1:nbrGen
x(1,j) = seed(j);
end
% Get the largest modulus
maxModulus = max(m);
% Calculate the x_js for the x
```

```

for i = 2:n
for j = 1:nbrGen
x(i,j) = mod(a(j)*x(i-1,j),m(j));
end
end
x = mod(sum(x,2),maxModulus);
% Calculate the uniformly distributed numbers
u = x/maxModulus;

```

Question 4:

1)

• **What is Monte Carlo Simulation?**

Monte Carlo simulation is a computerized mathematical technique that allows people to account for risk in quantitative analysis and decision making. The technique is used by professionals in such widely disparate fields as finance, project management, energy, manufacturing, engineering, research and development, insurance, oil & gas, transportation, and the environment.

Monte Carlo simulation furnishes the decision-maker with a range of possible outcomes and the probabilities they will occur for any choice of action. It shows the extreme possibilities the outcomes of going for broke and for the most conservative decision along with all possible consequences for middle-of-the-road decisions.

The technique was first used by scientists working on the atom bomb; it was named for Monte Carlo, the Monaco resort town renowned for its casinos. Since its introduction in World War II, Monte Carlo simulation has been used to model a variety of physical and conceptual systems.

• **How It Works**

In a Monte Carlo simulation, a random value is selected for each of the tasks, based on the range of estimates. The model is calculated based on this random value. The result of the model is recorded, and the process is repeated. A typical Monte Carlo simulation calculates the model hundreds or thousands of times, each time using different randomly-selected values. When the simulation is complete, we have a large number of results from the model, each based on random input values. These results are used to describe the likelihood, or probability, of reaching various results in the model.

2) The Monte Carlo method can be used to determine an unknown quantity through experiments involving random numbers. We illustrate the approach through an example that can be used to estimate the value of π . Suppose we have a square target with vertices $(1, 1)$, $(1, -1)$, $(-1, 1)$, $(-1, -1)$ with a circle having radius equal to 1 inside the square.

Suppose we now throw darts randomly at the target, and that they can land anywhere on the square with probability (i.e., uniform distribution). Then if we throw enough darts, it is reasonable to use the approximate:

$$\frac{\text{Number of throws inside circle}}{\text{Numbers of throws}} \approx \frac{\text{Area of circle}}{\text{Area of square}}.$$

Hence, this experiment can be used to approximate the value of π as:

$$\pi \approx 4 \frac{\text{Number of throws inside circle}}{\text{Numbers of throws}}.$$

3)

```
function [nt,nta] = radio_active(n,N,p,lambda)
% -----
% To simulate radio active decay by Monte Carlo method
clc;
clear;
dt=p/lambda;
nt=zeros(N,1); %nt holds MC result
tmax=N*dt;
t=(0:dt:tmax)';
nta=n*exp(-lambda*t); %Analytic solution
nt(1)=n;
% -----Monte Carlo Method-----
for i=2:N+1
    random=rand(n,1);
    g = random(random<=p);
    count=length(g)
    n=n-count;
    if(n<=0)
```



```
        break;
    end
    nt(i)=n;
end
% -----
plot(t,nt,'r',t,nta,'b');
xlabel('Time');
ylabel('No. of nuclides');
```