Numerical Computing II

Homework 17: Random Numbers

Marty Fuhry

February 21, 2010

Exercise 17.1

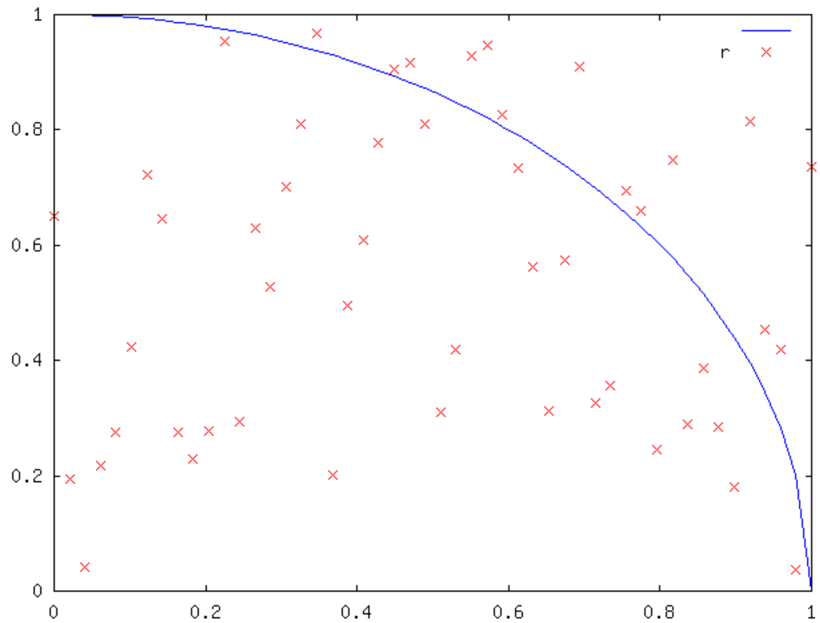
```
% create a sequence of random numbers
n  = 50;
r = rand(n,1);

% define our function values
x = linspace(0,1,n);
y = sqrt(1 - x.^2);

clf
hold on
plot(x,y);
plot(x,r, 'rx');

% find the ratio of randoms under the curve
u = 0;
for j = 1:n
        % count only if it's under the curve
        if r(j) < y(j)
                u = u + 1;
        endif
end
```

The graph produced by this code:



After running this code, we get:

$$
\begin{aligned}
u &= 36 \\
u/n &= 36/50 = 0.72000
\end{aligned}
$$

Since $\pi/4 \approx 0.78540$, our approximation of the area under the circle is very close. We may remark that by generating more and more random values, we will recieve closer approximations to the actual area of the quarter cirle. Observe what happens when we set $n = 10000$:

$$
\begin{aligned}
u &= 7885 \\
u/n &= 7885/10000 = 0.78850
\end{aligned}
$$

Now, this is a very close approximation.

Exercise 17.3

1. How long does it take for the sequence to repeat itself?

The following code generates this sequence:

```
n = 1000;
x = zeros(1,n);
x(1) = 1;
for j = 2:n
        x(j) = mod(3*x(j-1)-1, 8);
end
```

We can examine the random variables by looking at x:

$$x = 1, 2, 5, 6, 1, 2, 5, 6, 1, 2, 5, 6, ... \tag{1}$$

Clearly, this repeats every 4 numbers. If we were to write out the numbers we would see how quickly it repeats:

$$\begin{aligned}
&x_1 && = 1 \\
&x_2 = (3 * x_1 - 1)mod8 = 2mod8 && = 2 \\
&x_3 = (3 * x_2 - 1)mod8 = 5mod8 && = 5 \\
&x_4 = (3 * x_3 - 1)mod8 = 14mod8 && = 6 \\
&x_5 = (3 * x_4 - 1)mod8 = 17mod8 && = 1 = x_1 \\
&x_6 = (3 * x_5 - 1)mod8 = (3 * x_1 - 1)mod8 && = 2 = x_2
\end{aligned}$$

2. What are the smallest and largest values in the sequence?

The smallest value is 1, and the largest is 6.

3. Are the values in the sequence uniformly distributed in the interval defined in the last question?

No, not in the sense of a true uniform distribution where every value is equally spaced from the next. However, the middle values 2 and 5 are exactly 1 away from the end values, so we are indeed close.

Exercise 17.4

The odds of discovering 1, 1, 0 should be the same as the odds of discovering 0, 0, 1. This is simply because we have defined which random variables should be 1 and which should be 0. Had we rounded the other way, that is, had we switched every 1 and 0, we would duplicate our results.

Regardless, the probability of finding 1,1,0 before 0,1,1 (or vice-versa) is curiously low. This is not because of rarity of sequences of 1,1,0, but because we may not allow 0,1,1 to occur before 1,1,0. That is, we may not allow a 0 to preceed 1,1,0, or we will have formed the string 0,1,1,0, and thus, the string 0,1,1 will be formed before 1,1,0.

The following code will stop when either a 1,1,0 or 0,1,1 sequence occurs:

```
% generate n pseudo-random numbers
n = 100;
r = rand(n,1);

% round all numbers to closest 1 or 0
r = round(r);

found = -1;
% find first occuring pattern of 1,1,0
for j = 1:n-2
        if (r(j) == 1 & r(j+1) == 1 & r(j+2) == 0) | \
           (r(j) == 0 & r(j+1) == 1 & r(j+2) == 1)
                found = j;
                return;
        endif
end
```

Running this code a few times, we rarely find either of the two sequences beyond 4 numbers into the overall sequence.

We can find the number of times 0,1,1 occurs before 1,1,0 with the following code:

```
% determine number of random numbers to generate
n = 100;

count = 0;
for z = 1:100
        % generate n pseudo-random numbers and round them to 1 or 0
        r = rand(n,1);
        r = round(r);

        % find first occuring pattern of 1,1,0

    k = 1;
        while k < n-2
                if (r(k) == 1 & r(k+1) == 1 & r(k+2) == 0)
                        k = n-1;
                elseif (r(k) == 0 & r(k+1) == 1 & r(k+2) == 1)
                        count = count + 1;
                        k = n-1;
                endif
        k = k + 1;
        endwhile
end
```

Running this code, we find that 0,1,1 occurs before 1,1,0 a surprisingly high number of times. I found that 0,1,1 occurs before 1,1,0 around 75 times in 100 tests. This is because the string 0,1,1 contains the first two parts of the string 1,1,0. This means, in order for the string 1,1,0 to occur before the string 0,1,1, the string 1,1,0 needs to be preceeded by a 1, or be the first occurance in the set. If 1,1,0 is preceeded by a 0, then the string 0,1,1,0 is formed, and thus 0,1,1 occurs before 1,1,0. So what we are really looking for is either 1,1,0 occuring first or 1,1,1,0 occuring, which has much less probability of occuring than 0,1,1.

The same is not true for 0,1,1. We may put either a 1 or a 0 in front of 0,1,1, and it will still occur without 1,1,0 occuring. That is, 1,0,1,1 does not stop our search; neither does 0,0,1,1.