Homework 8: Randomness

Bhishan Poudel

Oct 24,2015

Contents

List of Figures

1 Question 1: Random Sequences

In this question I studied some programs to test and generate the random numbers. The programs are mod(a,M), ran(flag), drand(flag), and a subroutine $sobseqn.f90$.

1.1 part abc: intrinsic function mod(a,p)

In this part I used the intrinsic program of fortran 90 compiler called mod(a,p).
Description:
MOD(A,P) computes the remainder of the division of A by P.

Arguments:

A Shall be a scalar of type INTEGER or REAL.
P Shall be a scalar of the same type and kind as A and not equal to zero.

Return value: The return value is the result of A - (INT(A/P) * P). The type and kind of the return value is the same as that of the arguments. The returned value has the same sign as A and a magnitude less than the magnitude of P.

In this question I varied A and P so that it gives different random numbers.
Here, first argument $A = r_i = ar_{i-1} + c$
second argument $M = 256$
I varied the values of $r(i)$ and fixed value of $m = 256$ so that I got 256 random numbers.
Then I plotted $r(i)$ vs. $r(i+1)$.

The solution directory is :

```
location             : hw8/qn1abc, qn1d and qn1e
source code          : hw8qn1abc.f90, hw8qn1d.f90, hw8qn1e.f90
plots                : hw8qn1c.eps, hw8qn1d.eps, hw8qn1e.eps
datafiles            : hw8qn1a.dat, hw8qn1c.dat, hw8qn1e.dat
provided subroutines : sobseqn.f90
makefile             : Makefile
```

The figures are shown below:

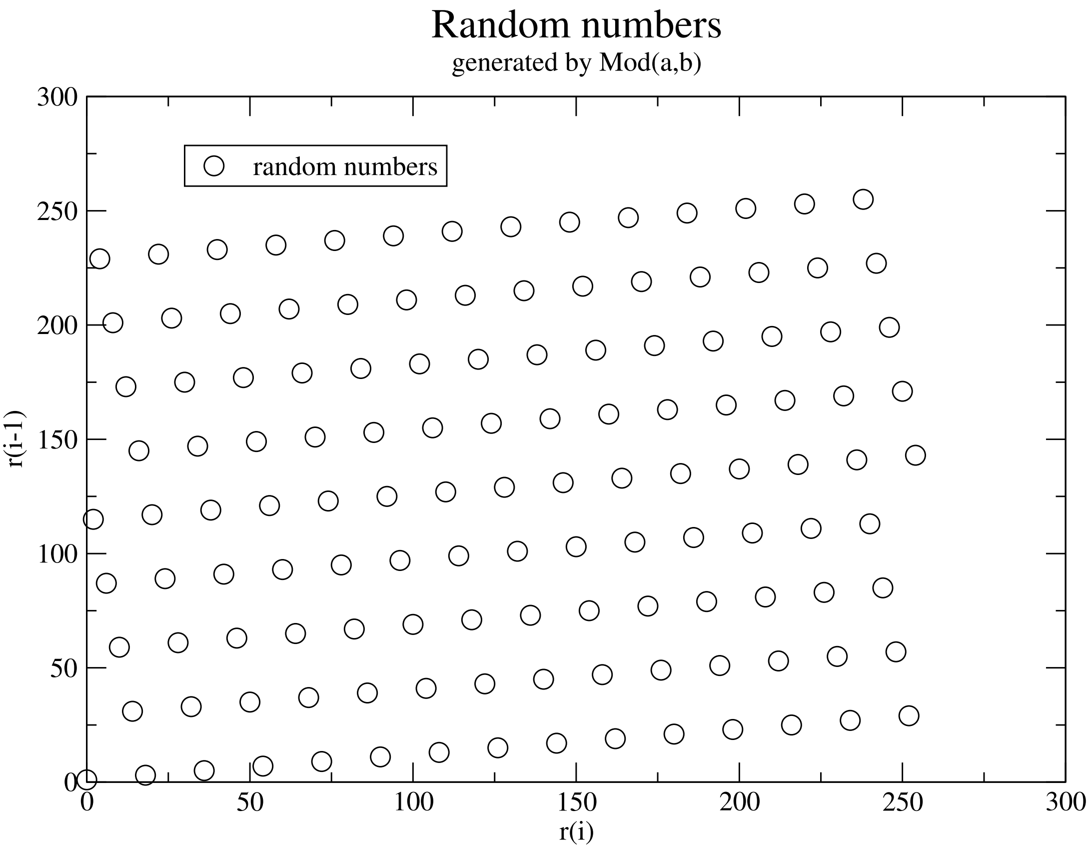

Figure 1: random numbers using mod(a,p)

1.2 part d: intrinsic function drand(seed)

In this part I studied the fortran intrinsic function drand(seed) to study random numbers. I created the data file for $r(i)$ vs. $r(i+1)$ and plotted the graph.

The figures are shown below:

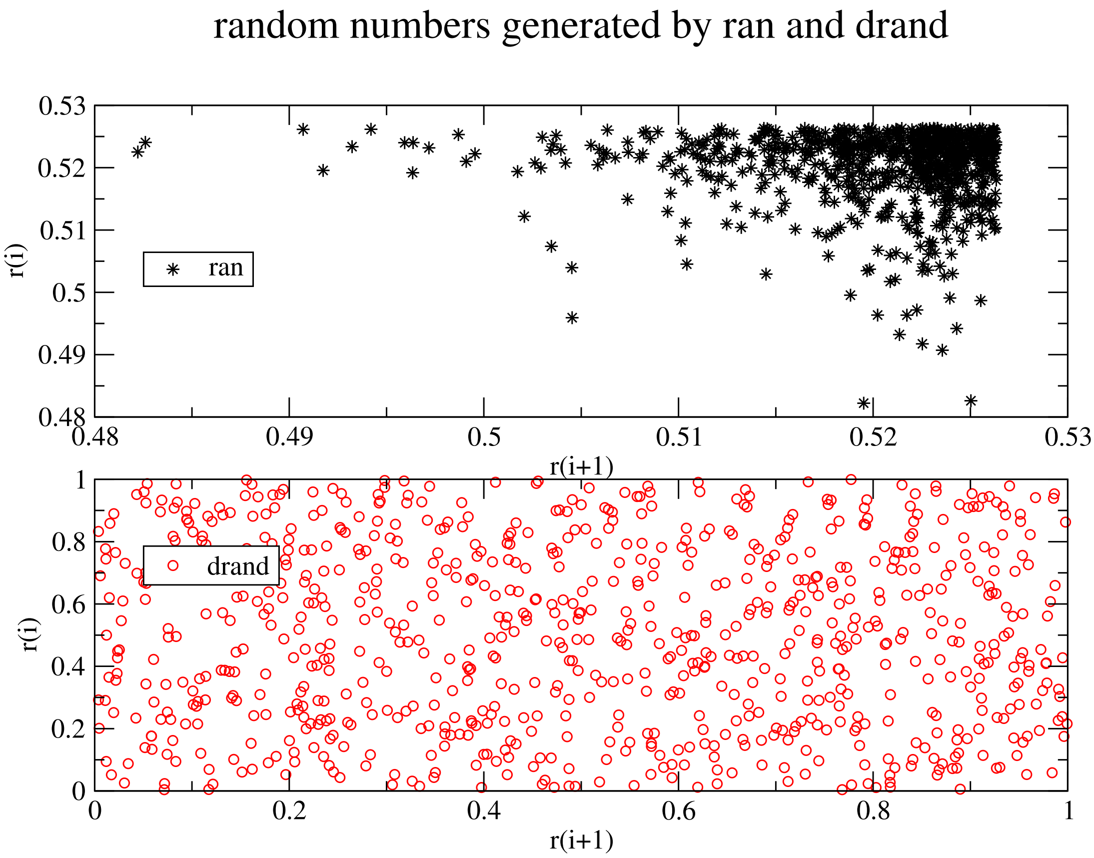

Figure 2: random numbers using drand(seed)

1.3 part e:

In this part I used the subroutine 'sobseqn' to create and study random numbers. Then, I plotted the graph of $r(i)$ vs. $r(i+1)$.

The figures are shown below:

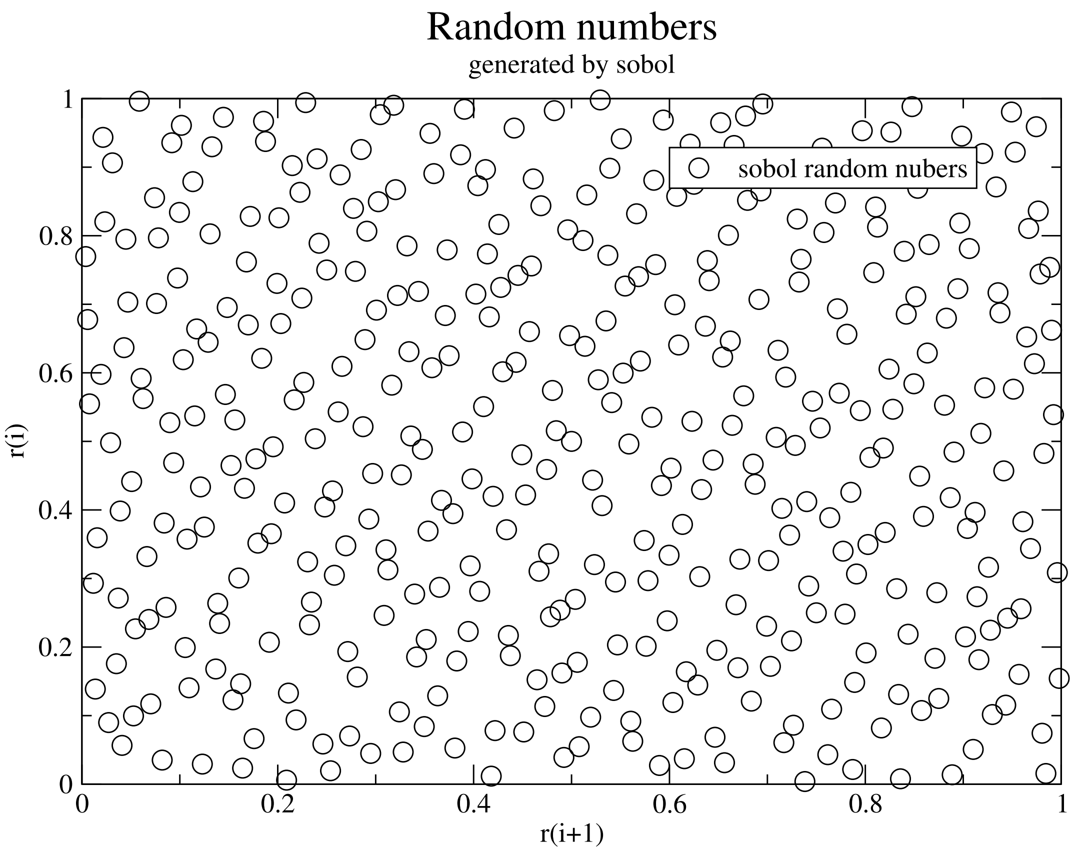

Figure 3: random numbers using subroutine 'sobseqn'

2 Question 2: Checks on Random Sequences

In this part I tested two random generator functions, viz. drand(seed) and 'sobseqn' to check the uniformity of these functions.

The solution directory is :

```
location             : hw8/qn2
source code          : rand_check.f90, drand_check.f90
datafiles            : rand_check.dat, drand_check.dat, hw8qn2b.dat
provided subroutines : randcheck.f90,stest.f90, sobseqn.f90
```

2.1 part a: checking uniformity for drand

In this part I tested fortran built-in function ran and drand for the uniformity. The code *randcheck.f90* was provided and I modified it.

2.2 part b: Testing sobol sequence

In this part I tested the given subroutine sobol sequence. The code *stest.f90* was modified. The source code is *hw8qn2b.f90*.

3 Question 3: Random Walk (Landau second edition page 147)

3.1 part ab:

In this part I modified the code *walk.f90*, normalized the plot and the plot looks like as I expected.

The figures are shown below:

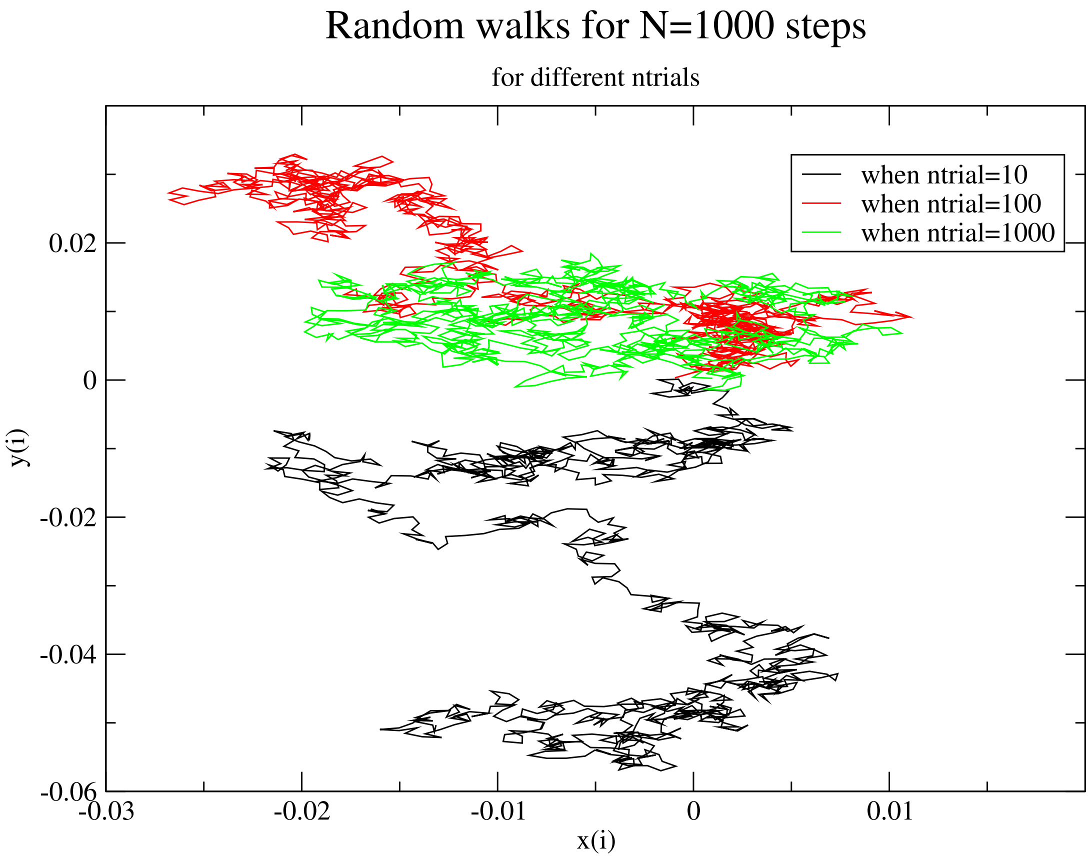

Figure 4: plot of random walks

3.2 part cde:

In this part each trial have 1000 steps and calculated the root mean square distance. I plotted R_{rms} versus $\sqrt{N}$. I started N with small value and I took 3 significant figures. Here I took $N = 1000$, when N increases the gaussian distribution fits well and values of rms distance and square root of N becomes closer and closer. The plot is shown below:

The figures are shown below:

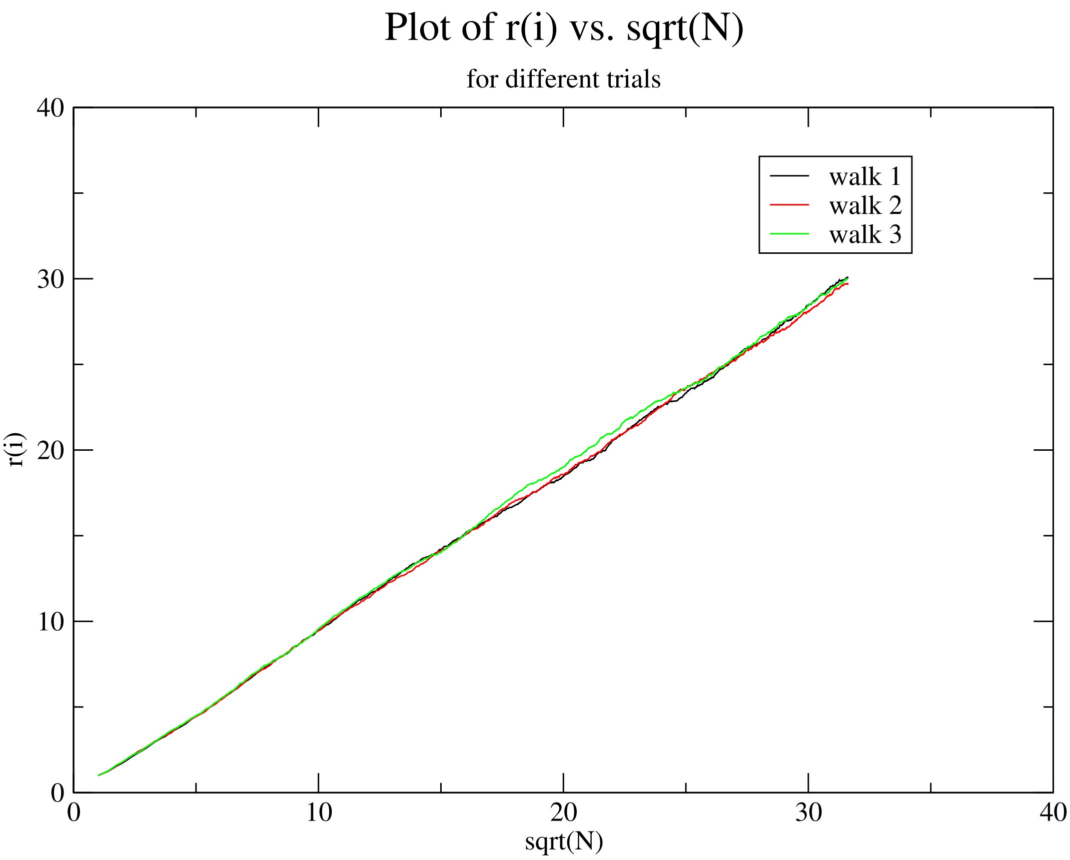

Figure 5: plot of rms distance versus square root of steps

3.3 part f:

In this part I plotted the scatterplot of random walk. The plot is uniform in all the four quadrant. The plot is shown below:

The figures are shown below:

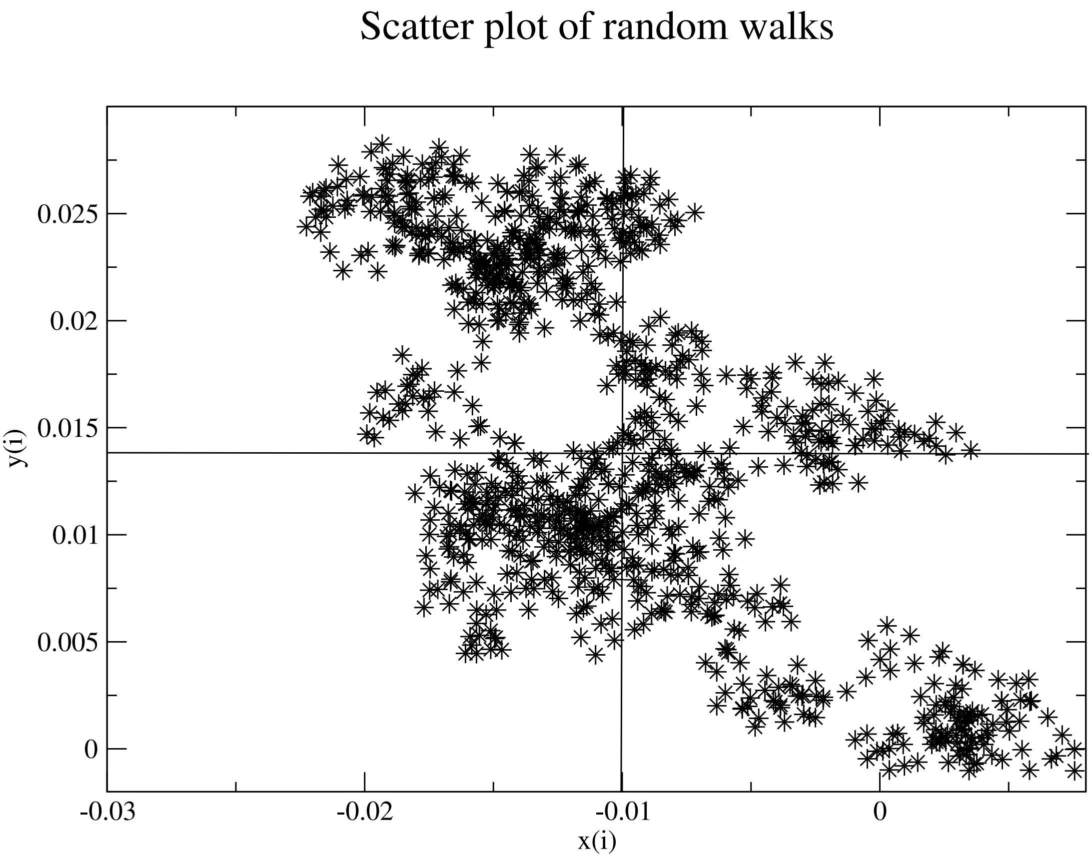

Figure 6: scatterplot