

C PROGRAMMING: TESTING RANDOM NUMBER GENERATORS

We introduced **pseudo random numbers** and wrote a program to generate them. We looked at a few of them and they sort of looked random, but let's examine them a bit more closely.

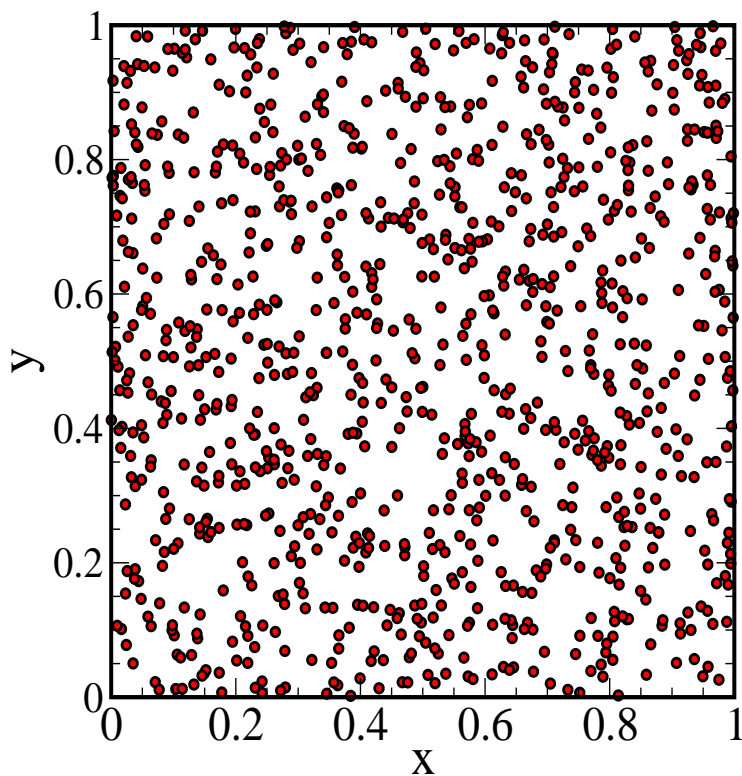
Your eyes are incredibly good at detecting patterns. A nice test takes advantage of your visual acuity: generate random numbers in pairs $(r_1, r_2), (r_3, r_4), (r_5, r_6) \dots$ and use `xmgrace` as an (x, y) plot. Take a minute to think about what you should see if your numbers are really random. Then write a program to generate the data. Then use `xmgrace` to plot the pairs.

```
#include <stdio.h>
#include <time.h>//This is new!
#include <stdlib.h>//This is new!
int main(){
    srand(time(NULL));
    int i,N;
    double R1,R2;
    FILE * fileout;
    fileout=fopen("randompairs.dat","w");
    printf("Enter the number of random pairs ");
    scanf("%d",&N);
    for(i=0;i<N;i++)
    {
        R1=(double)rand()/RAND_MAX;
        R2=(double)rand()/RAND_MAX;
        fprintf(fileout,"%lf %lf \n",R1,R2);
    }
    fclose(fileout);
}
```

Comments:

[1] We are writing the results to a file which ends in .dat so that xmgrace will pick it up easily.

[2] When you plot in xmgrace you might want to eliminate the lines connecting the points and only show the points themselves. The lines confuse things a bit. Here is a result for 1000 random pairs:



[3] This may *seem* like a silly test, but actually, a flaw in the most commonly used random number generator of its day was discovered by basically this method. The generator was much like the ‘bad’ one we introduced first, and its failure is described in the Proceedings of the National Academy of Sciences: “*Random Numbers Fall Mainly in the Planes,*” by George Marsaglia:

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC285899/>