Class-Work : Monte Carlo Simulation for PI

The idea is to uniformly distribute the points in a square of side 2 unit within a circle of unit radius lie. Now by intution we know the ratio of points will be same as the ratio of their areas.

$$\frac{N_{circle}}{N_{total}} = \frac{Area_{circle}}{Area_{square}}$$

Now $Area_{circle} = \pi$ (for unit radii) and $Area_{square} = 4$ (for side of 2 unit). it will lead the above equation in:

$$Area_{circle} = \pi = 4 * \frac{N_{circle}}{N_{total}}$$

Statistical formulas to be used in this program.

$$1.\text{Mean } \ < A >= \frac{\sum_{i=0}^{N} A_i}{N}$$

$$2.\text{Standard deviation in Area} = \sqrt{N_c(1 - \frac{N_c}{N_s})} * \frac{A_s}{N_s}$$

$$3.\text{Dispersion in mean} = \sqrt{\frac{\sum_{i=0}^{N}(A_i - < A >)^2}{N-1}}$$

In [ ]:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>
// generating random numbers b/w range
double randnum(double min, double max) {
    double random = ((double)rand())/(double)RAND_MAX;
    return (max-min)*random + min;
}
// calculating mean of an array
double mean(int N,double arr[N]) {
    double sum=0;
    for (int i=0;i<N;++i)
        sum += arr[i];
    return sum/N;
}
// calculating dispersion relation of an array
double dispersion(int N,double arr[N]) {
    double avg,std=0;
    avg=mean(N,arr);
    for (int i=0;i<N;++i)
        std += pow((arr[i]-avg),2);
    return sqrt(std/(N-1));
}
// adding the error from each expt
double std(int N,double arr[N]) {
    double sum=0;
    for(int i=0;i<N;++i)
```

```c
        sum += (arr[i]*arr[i]);
    return sqrt(sum)/N;
}
void main()
{
    int i,j,n=1000;
    int M=10;   // repeating the expt. M times
    double x[n],y[n],z[n];
    double As=4,area[M],sigma[M];

    FILE*fp=NULL;
    fp=fopen("pi10.txt","w");
    srand(time(0));
    //calculating the area multiple times
    for (j=0;j<M;++j) {
        double Naccept=0;
        for (i=0;i<n;++i) {
            x[i]=randnum(-1,1);
            y[i]=randnum(-1,1);
            z[i]=pow(x[i],2)+pow(y[i],2);
            if (z[i]<=1){
                Naccept++;
            }
        }
        area[j]=As*Naccept/n;
        sigma[j]=sqrt(Naccept*(1-Naccept/n))*As/n;
        //printf("M[%d]\t%lf\t%lf\n",j+1,area[j],sigma[j]);
        fprintf(fp,"%d\t%lf\t%lf\n",j+1,area[j],sigma[j]);
    }
    printf("\nTotal performed expt:%d\n",M);
    printf("mean: %lf\tbinomial std deviation: %lf\n",mean(M,area),std(M,sigma))
    printf("dispersion in mean: %lf\n",dispersion(M,area));
}
```

OUTPUT :

Total performed expt:500

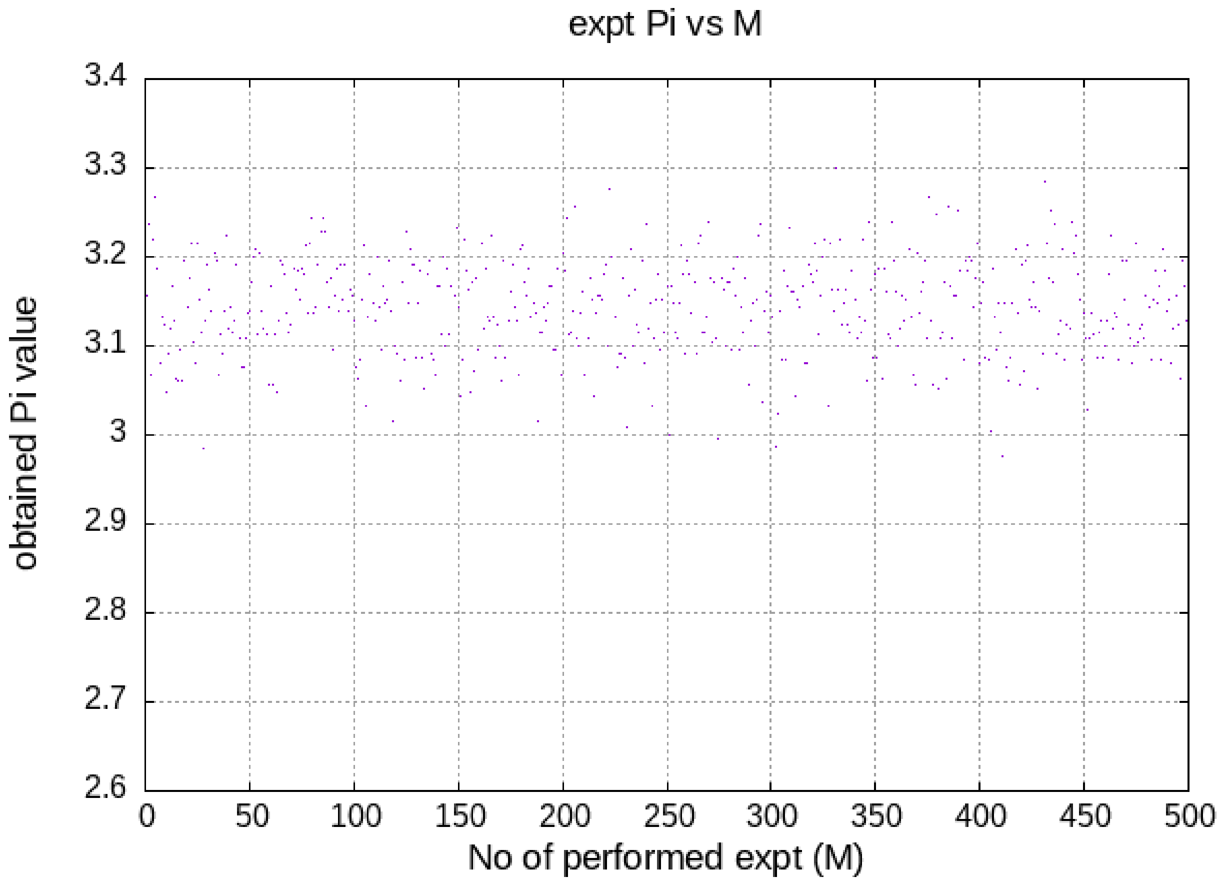
mean: 3.142240 binomial std deviation: 0.002320

dispersion in mean: 0.054359

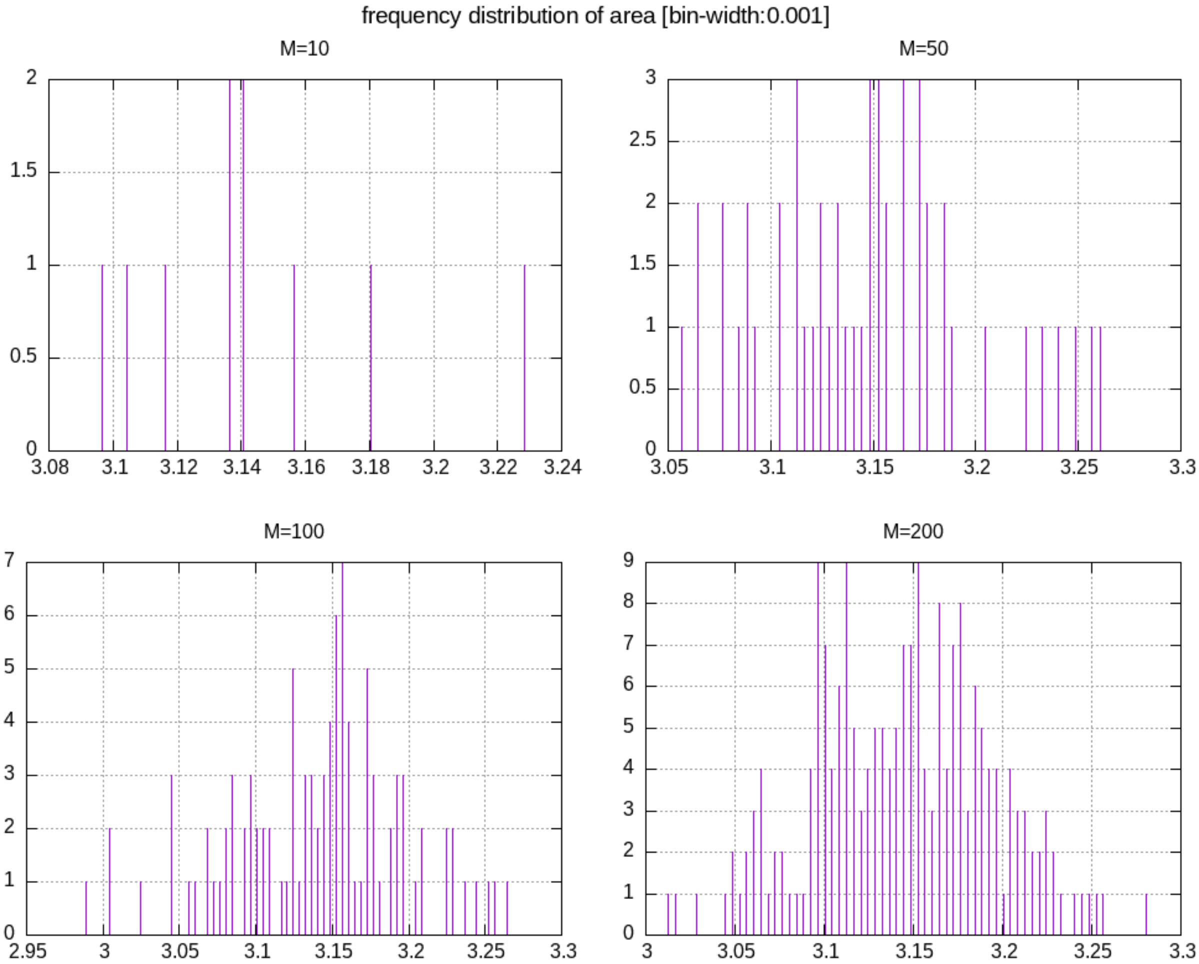
Performing the same experiment for $M = \{10, 50, 100, 500\}$

Also calculating the mean and the standard deviation using binomial distribution and the dispersion in the mean value which are tabulated below as a function of M.

M	MEAN	ERROR	DISPERSION
10	3.14320	0.016407	0.038496
50	3.14640	0.007326	0.050745
100	3.14012	0.005193	0.056341
200	3.14336	0.003668	0.050349
500	3.14224	0.002320	0.054359



Thu Mar 25 14:09:21 2021


frequency distribution of area [bin-width:0.001]

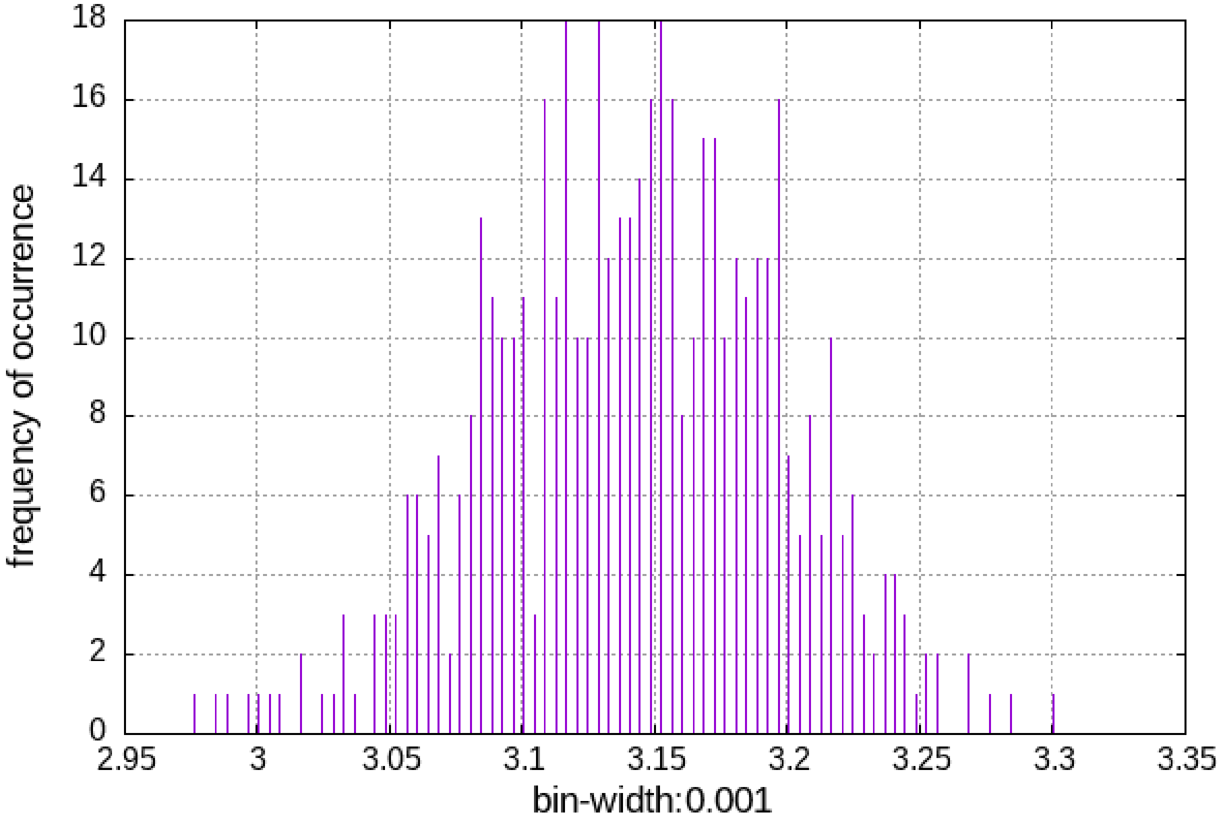
M=10

M=50

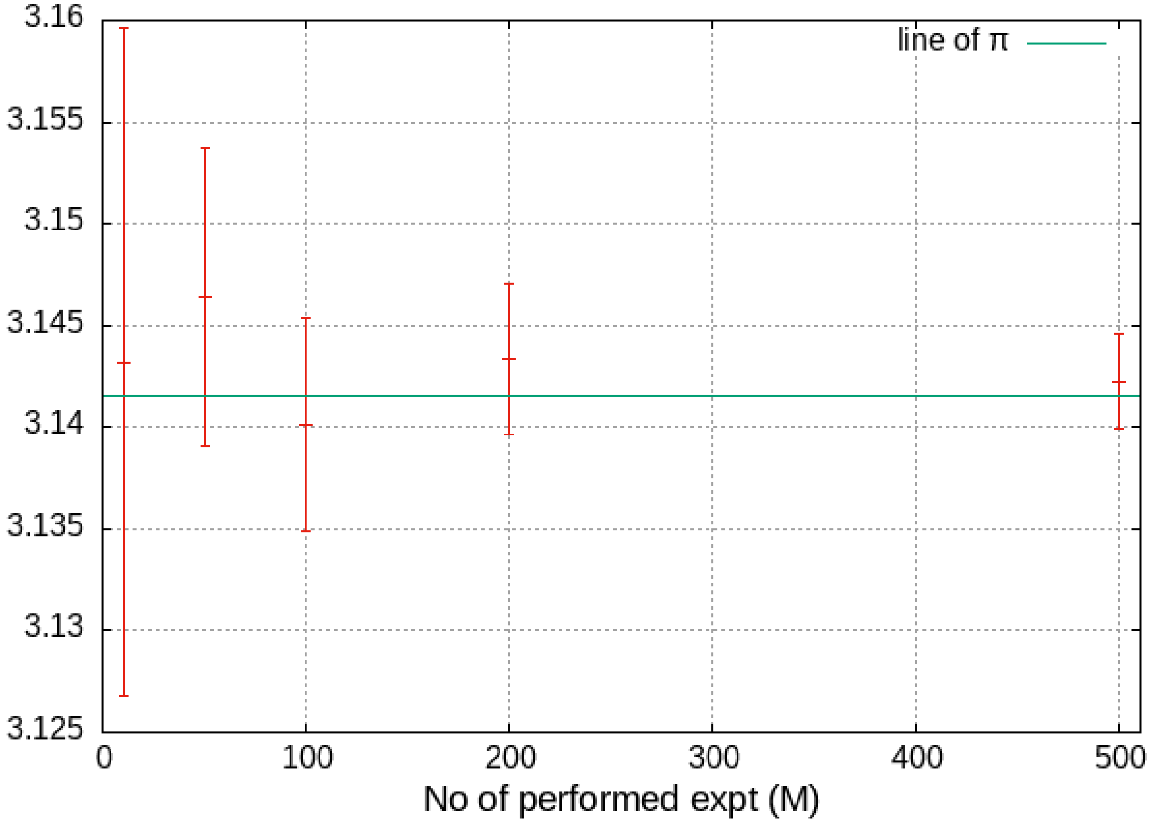
M=100

M=200

Thu Mar 25 14:15:25 2021


frequency distribution of area [M=500]

Thu Mar 25 14:10:56 2021


Mean value of Pi with std deviation

Thu Mar 25 14:28:15 2021