Indian Institute of Technology, Guwahati

A REPORT

ON

Dynamics of Active Particles

Bachelor of Technology, Physics

SUBMITTED BY

Somesh Raj (200121052)

UNDER THE GUIDANCE OF

Prof. Sitangshu Bikas Santra

(Academic Year: 2023-2024)

ACKNOWLEDGEMENT

It gives me great pleasure and satisfaction in presenting this mini project that is based on "Dynamics of Active Particles and there behaviour on External Noises".

I would like to express my deep sense of gratitude towards my project guide Prof. Sitangshu Bikas Santra and Dr. Jahir A Ahmed

Without them this project could never have been feasible I am also thankful to my parents and classmates for their encouragement and constant support

I would like to thank all those, who have directly or indirectly helped me for the completion of the work during this mini project.

Contents

Abstract

The report is basically the study of dynamics of active particles which are being studied using the vicsek 2D model, which mainly describes how the ordered motion of self propelled particles changes to disordered motion while applying the external noise in our case this can be time dependent too. Furthermore we are also going to understand what happens to the system when we apply time dependent external noise that is step "high and low", or more preferably square wave function and Sinusoidal Square Wave noise.

First part of this report talks about constant noise after that we jump onto understanding the behaviour of system under the time dependent noise.

Keywords-*Active Matter, 2-D Vicsek Model, Time-Dependent Noise*

Chapter 1

INTRODUCTION

1.1 Introduction

We are basically studying about the collective motion that is very widely observed in the active matter which includes School of fishes, flocks of birds, Human Crowd, though this flocking motion occurs at very different scales of length, the collective motion behaviour is very much similar in these wide range of systems. Here are some images of this behaviour in nature

We describe the collective motion of these particles by a simple model called as 'vicsek model', In this model a very large number of particles move (that are point like) move together with a constant velocity v_0 and they update their direction of motion as per the average direction of motion particles within an area enclosed by radius R. Also this average direction in subjected to external noises η.

We find out that a at a given density ρ_0, a change in orientational order to disorder phase transition occurs at $\eta = \eta_c$.

We are going to study the effect of constant noise on the system, and then we are going to apply time dependent noise.We are going to study the phase transition at critical noise by plotting by studying several parameters such as Order parameter, Binder cumulant and Variance, We are going to plot their variation with external noise.
In case of time dependent noise we are studying their variation with noise but around the critical noise value (η_c).

1.2 Active Matter

Active matter refers to a type of materials that exhibit self-organization and collective behavior due to the motion of their individual components. These components are typically microscopic or can be much smaller and these can be living or non-living. Examples of active matter systems can include flocks of birds, schools of fish, swarms of bacteria, and artificial micro- and nanorobots. They consume energy from the surrounding system and are so driven. This energy input allows the individual components

to move and interact with each other, leading to emergent behavior at larger scales. This self driven behaviour is called as self propelled system

1.3 Modelling Active Matter

There are several methods to study the collective motion of active matter in detail. These are some of the important methods:

1.3.1 Molecular Dynamics

The potential energy function is used in molecular dynamics simulations to represent interactions between atoms and molecules. Interactions between atoms and molecules in a system, such as intramolecular interactions and intermolecular interactions, are represented by potential energy functions.

Protein folding, drug discovery, materials science, and atmospheric chemistry are just a few of the many physical, chemical, and biological phenomena that can be studied with molecular dynamics simulations. They can provide information about atomic and molecular behavior in complex systems that are difficult or impossible to study experimentally.

1.3.2 Kinetic Monte Carlo

The system is modelled as a set of discrete states in KMC simulations, and transitions between these states take place probabilistically depending on rates estimated from the underlying physical processes. The simulation continues by picking a transition event at random and moving the system to the following state in accordance with the selected transition rate. Until the desired length of simulation time has passed or a steady state has been reached, this process is repeated.

1.3.3 Cellular Potts method

This method is used for modelling the biological aspects at cellular level, It is particularly useful studying the behaviour of the cells in complex tissues or multicellular organisms, it can also predict the growth and developement of biological structures.

1.3.4 Vicsek Model

The Vicsek model is widely used to study emergent collective behavior in large groups of self-propelled particles. This model can be used, for example, to simulate flock formation, the emergence of coordinated movements, or transitions to a chaotic state. It has been applied to various systems such as animal herds, human herds, and even self-organizing robots.

We are going to use the vicsek model to simulate the active matter, in this next chapter we are going to find out mathematical model of the vicsek model and discuss what parameteres are there with it.

Chapter 2

Vicsek Model

2.1 Basic Introduction

This is very simple model to study the dynamics of the active matter, This is mainly a mathematical model we are going use this mathematical model as the basis to our simulation.
In this model we are considering 2-D vicsek model,It consist of polar-point particles which are self propelled moving with constant velocity v_0, these particles can move randomly in any direction and there is no mutual interaction between these particles, At every time step each particle updates it's position along the average direction of particle in a circle of unit radius around it, where we add some external perturbation (Noise).

This model exhibits a phase transition from an ordered state to a disordered state as the noise level is increased above a critical value.The liquid-gas transition and the ferromagnetic-paramagnetic transition are two examples of phase transitions that are similar to the order-disorder transition in the Vicsek model. In such transitions, altering a control parameter, such as the temperature or magnetic field, causes the system's overall behaviour to change.

2.2 Mathematical Modelling of Vicsek Model

In this section we are going to mathematically describe the vicsek model.

2.2.1 Description

We are considering a 2D box of length and breadth L, Let there we total N particles in the box, So the number density $\rho = \frac{N}{L^2}$, which remains constant in every part of simulation, Hence we are not changing the box shape and the number of particles.
Let v_0 be constant velocity associated with each particle in the simulation, Let us focus at a particular i^{th} particle, the direction of this particle depends upon the average direction of all the in a circle of radius R, i^{th} as center, this circle is called as Circle of Influence.

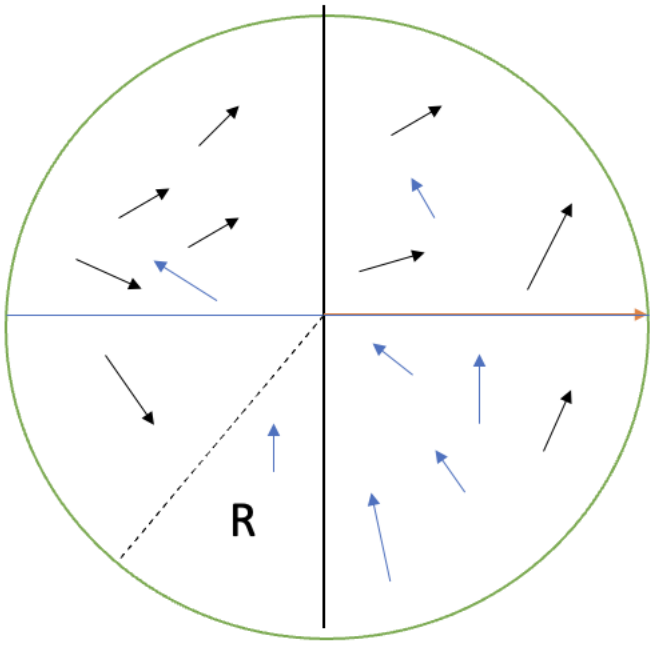


Fig1. Particle align to the average direction of all the particles.

We consider this particle $\vec{r}_i$ it has a velocity v_0 in θ_i direction ,this θ_i is randomly distributed between -π to +π, These particles align their directions of motion with that of their neighbours in the presence of some external constant noise. The radius of interaction is set to R = 1.
At every step increase of time Δt, we evaluate the new direction of this particle is given as $\theta_i(t + \Delta t) = \langle\theta(t)\rangle_R + \Delta\theta$,
$\langle....\rangle_R$ denotes the average of the direction of all the particles around the circle of influence centered around (x_i, y_i),
$\Delta\theta$ is a random orientation which belongs to [-$\eta\pi$, +$\eta\pi$] and $\eta \in [0, 1]$ is the noise parameter it is the main parameter of the simulation.

Chapter 3

Simulating Vicsek Model

3.1 Algorithm

3.1.1 Random Number Generator

We are using random number generator for generating the random position and directions of the swimmers at the starting of the simulation

```
float ran2(long *idum)
{
    int j;
    long k;
    static long idum2 = 123456789;
    static long iy = 0;
    static long iv[NTAB];
    float temp;
    if (*idum <= 0)
    {
        if (-(*idum) < 1)
            *idum = 1;
        else
            *idum = -(*idum);
        idum2 = (*idum);
        for (j = NTAB + 7; j >= 0; j--)
        {
            k = (*idum) / IQ1;
            *idum = IA1 * (*idum - k * IQ1) - k * IR1;
            if (*idum < 0)
                *idum += IM1;
            if (j < NTAB)
                iv[j] = *idum;
        }
        iy = iv[0];
    }
    k = (*idum) / IQ1;
    *idum = IA1 * (*idum - k * IQ1) - k * IR1;
    if (*idum < 0)
        *idum += IM1;
    k = idum2 / IQ2;
    idum2 = IA2 * (idum2 - k * IQ2) - k * IR2;
    if (idum2 < 0)
        idum2 += IM2;
    j = iy / NDIV;
    iy = iv[j] - idum2;
    iv[j] = *idum;
    if (iy < 1)
        iy += IMM1;
    if ((temp = AM * iy) > RNMX)
        return RNMX;
    else
        return temp;
}
```

Algorithm behind Random Generator Function

we implemented the random number generator algorithm, it generates a sequence of uniformly distributed pseudo-random numbers between 0 and 1. Uses linear congru-

ential generators and subtraction methods to improve randomness, and use a set of static variables to maintain generator state across multiple function calls. This algorithm performs two LCG operations with different sets of coefficients, subtracts the resulting value, selects an element from the array based on the result, and multiplies the selected element by a constant to calculate the output value. to generate a pseudo-random number. This algorithm keeps generating new pseudo-random numbers each time the function is called.

3.1.2 Initialization

The configuration of the box are , L=32 unit, LS=L*L $unit^2$, Lh=L/2 unit, R=1 unit, ETA is the amplitude of the noise function (this is constant in (i) part and periodic in the other two sections), SATTIME=10,000 s, TTIME=10,000 s,
These things will become clear later.

```
//============ INITIALIZATION ==============
for (i = 0; i < LS; i++)
{
    NBOX[i] = 0; // No of the particles in the box i
    for (j = 0; j < N; j++)
        BOX[i][j] = 0; // index of the particles in the box i
}

for (i = 0; i < N; i++)
{
    x[i] = ran2(&iseed) * L;                // x position
    y[i] = ran2(&iseed) * L;                // y position
    theta[i][0] = ran2(&iseed) * 2 * PI - PI; // oriented angle
    theta[i][1] = 0;                        // to avoid garbage value...
    list[i] = 0;
}
```

NBOX is a 1-D list which is of the 32×32 length it signifies a grid and each grid element is initially initialised with zero swimmers inside of it.
Note that this is just a one dimensional array that is being used to represent the number of swimmers in the n^{th} box,

NBOX [i] = Number of Swimmers in i^{th} box.

BOX [LS] [N] : This is 2-D list it signifies the index of swimmers present in particular i^{th} box, Initially we are initializing this to zero.

In the second for loop:

We are having two lists X [N] and Y [N] which are basically the position coordinates of the swimmers.
We are initializing these coordinates Random values between [0,32] using the random generator function.
In the next line we have theta [N] [2] which are containing the random direction between [-π, +π]
In the next line we have list[N] which is being used for finding the number of swimmers inside the circle of the influence, we are initializing each element to zero.

3.1.3 Assigning Swimmer's index to BOX [LS] [N]

```
for (i = 0; i < N; i++) // loop over the particles
{
    l = x[i];
    m = y[i];
    box = m * L + l;
    BOX[box][NBOX[box]] = i; // which particles are in the box
    NBOX[box]++;
}
```

In this code we are finding the integer 'box' which denotes in which box a particular (x,y) coordinate lies in.

box=m*L +l ,this thing gives the box values according to the l,m value.
After this we enter the index of this particular particle with reference of which box it actually in inside of:

BOX [box] [NBOX [box]]= i;
After this we increment the number of particles inside that particular box.
NBOX [box] ++ ;

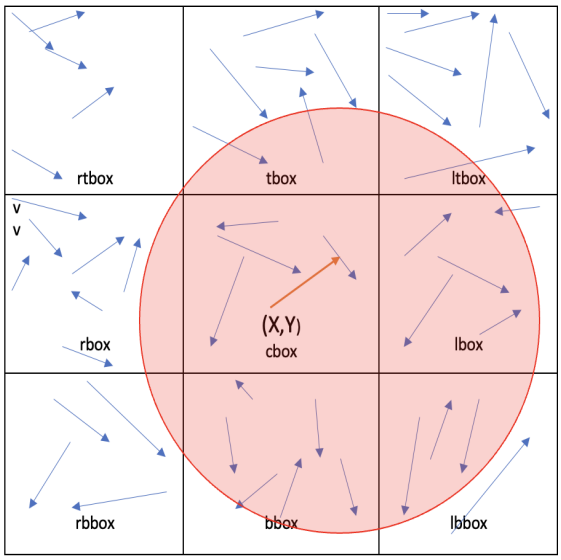
3.1.4 Running the Simulation

We are not including code as it is very long:
Here is a thorough explanation of the time loop:
This simulation loop is the main part of the program where the dynamics of the system are being simulated. It iterates over the time steps from 0 to TTIME. Description of the time loop:

1. nsamp - keeps track of the number of samples for calculating the average order parameter and other important parameter to be plotted later.
2. PHI - average order parameter.
3. PHI2 - the square of the average order parameter.
4. PHI4 - the fourth power of the average order parameter.
5. X and Y are the current positions of the swimmer i.
6. cbox is the box in which the swimmer i is located.
7. lbox, tbox, rbox, and bbox are the boxes that are adjacent to the current box in the left, top, right, and bottom directions, respectively.
8. ltbox, rtbox, lbbbox, and rbbox are the boxes that are adjacent to the current box in the diagonal directions.
9. list is an array that keeps track of the nearby swimmers within a distance of R.

Algorithm for finding the swimmers inside the circle of radius of 1 unit

Here is diagram showing a circle of radius 1 unit and all the swimmers that can be found inside of it.



Any general circle of influence around any i^{th} point can be thought of surrounded by these eight boxes. These 9 boxes include cbox containing the i^{th} particle.
Hence at last we have to check for the other particle coordinates which lie inside these nine boxes only and put them in circle equation to find if it is inside the circle and calculate the average.
All the indexes of swimmers inside these 9 boxes are stored inside the list [N], through this list we find out whether it's inside the circle of influence or not. This code will serve the task.

```
avgdx = cos(theta[i][0]); // x component
avgdy = sin(theta[i][0]); // y component

np = 1; // number of particles

for (j = 0; j < m; j++) // Loop over other swimmers in the list
{
    dr = sqrt(pow((x[i] - x[list[j]]), 2) + pow((y[i] - y[list[j]]), 2));
    if (dr > Lh)
        dr = L - dr; // corrected dr
    if (dr <= R)
    {
        avgdx += cos(theta[list[j]][0]);
        avgdy += sin(theta[list[j]][0]);
        np++; // # of swimmers inside the unit circle
    }
}

rantheta = (ran2(&iseed) - 0.5) * 2 * PI;
avgdx += ETA * np * cos(rantheta);
avgdy += ETA * np * sin(rantheta);

if (avgdy > 0)
    avgtheta = acos(avgdx / sqrt(pow(avgdx, 2) + pow(avgdy, 2)));
if (avgdy < 0)
    avgtheta = -acos(avgdx / sqrt(pow(avgdx, 2) + pow(avgdy, 2)));
if (avgdy == 0)
{
    if (avgdx > 0)
        avgtheta = 0;
    if (avgdx < 0)
        avgtheta = PI;
}

theta[i][1] = avgtheta;
```

We are iterating over the all the elements inside the list [N] and checking if x[list[i]] and y[list[i]] exist inside the circle of influence using this code piece

dr = sqrt(pow((x[i] - x[list[j]]), 2) + pow((y[i] - y[list[j]]), 2));

Now dr can be greater than L/2 to (if we are considering the corner cases) in such cases we have substract L from it.
After this is done we check whether dr $\geq$ R or not. if it's within the circle we calculate the average direction of the i^{th} swimmer this process is repeated for all N swimmers using a for loop.
We are doing the above process of calculating the average direction of all the particle at each time step.

3.1.5 Periodic Boundary Condition

There is loop over all the swimmers for there position update. Where we are letting all the particle move along their corresponding direction for each iteration of time. And also we are setting the periodic boundary condition, If a particle moves out of the box it is brought back to the inside of the box. Using this code

```
  for(i=0;i<N;i++) //Loop over swimmers for position update
{
  theta[i][0]=theta[i][1];
  x[i]=x[i]+V0*cos(theta[i][1])*dt;
  y[i]=y[i]+V0*sin(theta[i][1])*dt;

  if(x[i]>L)x[i]=x[i]-L; //PBC
  if(x[i]<0)x[i]=L+x[i]; //PBC
  if(y[i]>L)y[i]=y[i]-L; //PBC
  if(y[i]<0)y[i]=L+y[i]; //PBC

  X=x[i];
  Y=y[i];
  box=Y*L+X;

  BOX[box][NBOX[box]]=i;
  NBOX[box]++;
}

  if(time%100==0)
{
  vx=0.0;
  vy=0.0;
  for(i=0;i<N;i++)
    {
      vx+=V0*cos(theta[i][0]);
      vy+=V0*sin(theta[i][0]);
    }
  va=sqrt(pow(vx,2)+pow(vy,2))/(V0*N);
  fprintf(fp,"%ld   %lf\n",time,va);

  if(time>SAT_TIME)
    {
      nsamp++;
      PHI+=va;
      PHI2+=pow(va,2);
      PHI4+=pow(va,4);
    }//If Saturation Time
}
```

In this loop over all the swimmers we find if time%100 == 0, then we calculate the average velocity of the system and store that in time series file, moreover in the same if condition if time $\geq$ SAT_TIME, we calculate PHI, PHI2 and PHI4.
We are using gnuplot to visually represent the flocking nature of the Swimmers, I am not explaining the use of gnuplot.

3.1.6 Calculation of the final parameters

1. We calculate the order parameter, The order parameter in the Vicsek model measures the degree of alignment or coherence of the self-propelled particles in the system.

   $\phi = \frac{1}{N}\sum_{i=1}^{N} |\vec{v_i}|$ where $|\vec{v_i}|$ is the velocity i^{th} swimmer.

2. Binder Cumulant is defined as the ratio of the fourth moment of the order parameter to the square of the second moment. That's how the formula says actually but in reality we use it to define the critacal noise value.

   $G = 1 - \frac{\langle \phi^4 \rangle}{3\langle \phi^2 \rangle^2}$

3. Last parameter is χ (Susceptibility), At the critical point, the susceptibility is expected to diverge, indicating that the system becomes more and more sensitive to changes in the noise parameter as the phase transition is approached.

   $\chi = \langle \phi^2 \rangle - \langle \phi \rangle^2$

Finally after the time loop finishes we have ϕ with us, from that we can calculate the other to parameters easily.

```
  for(i=0;i<N;i++) //Loop over swimmers for position update
{
  theta[i][0]=theta[i][1];
  x[i]=x[i]+V0*cos(theta[i][1])*dt;
  y[i]=y[i]+V0*sin(theta[i][1])*dt;

  if(x[i]>L)x[i]=x[i]-L; //PBC
  if(x[i]<0)x[i]=L+x[i]; //PBC
  if(y[i]>L)y[i]=y[i]-L; //PBC
  if(y[i]<0)y[i]=L+y[i]; //PBC

  X=x[i];
  Y=y[i];
  box=Y*L+X;

  BOX[box][NBOX[box]]=i;
  NBOX[box]++;
}

  if(time%100==0)
{
  vx=0.0;
  vy=0.0;
  for(i=0;i<N;i++)
    {
      vx+=V0*cos(theta[i][0]);
      vy+=V0*sin(theta[i][0]);
    }
  va=sqrt(pow(vx,2)+pow(vy,2))/(V0*N);
  fprintf(fp,"%ld    %lf\n",time,va);

  if(time>SAT_TIME)
    {
      nsamp++;
      PHI+=va;
      PHI2+=pow(va,2);
      PHI4+=pow(va,4);
    }//If Saturation Time
}
```

Chapter 4

Results and Discussions

4.1 Constant Noise

In this part of the simulation we are running our simulation with constant value of noise.
We are taking η = 0.0, 0.1, 0.2, ..., 1.0 and running our simulation to get values of corresponding parameter.

Afterwards we are plotting the variation of theses parameter vs noise, we are plotting ϕ vs η, G vs η and χ vs η.

These are simulation parameters:

1. Simulation Box: L = 32, N = 2×32×32, ρ = 2 unit^{-2}, R = 1 unit, v_0 = 0.3, Saturation time = 100000, Time = 100000, Total time = 200000
2. η is constant and takes values 0.0, 0.1, 0.2, ..., 1.0.

Here are some snapshot of the swimmers.

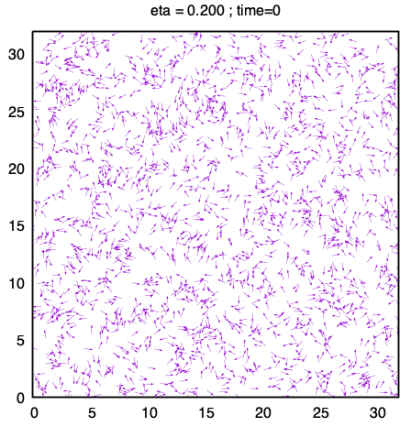


Figure 4.1: t = 0

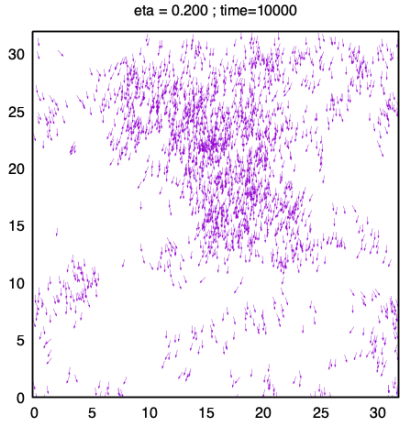
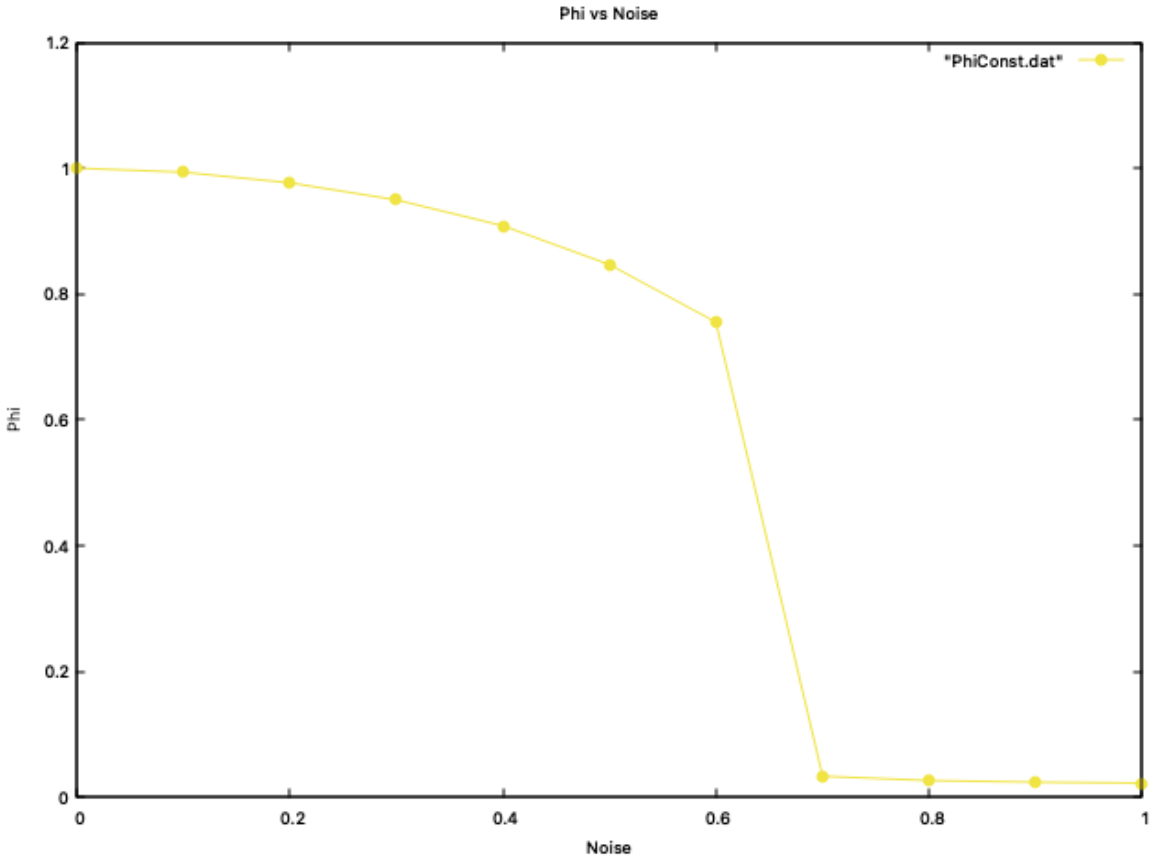
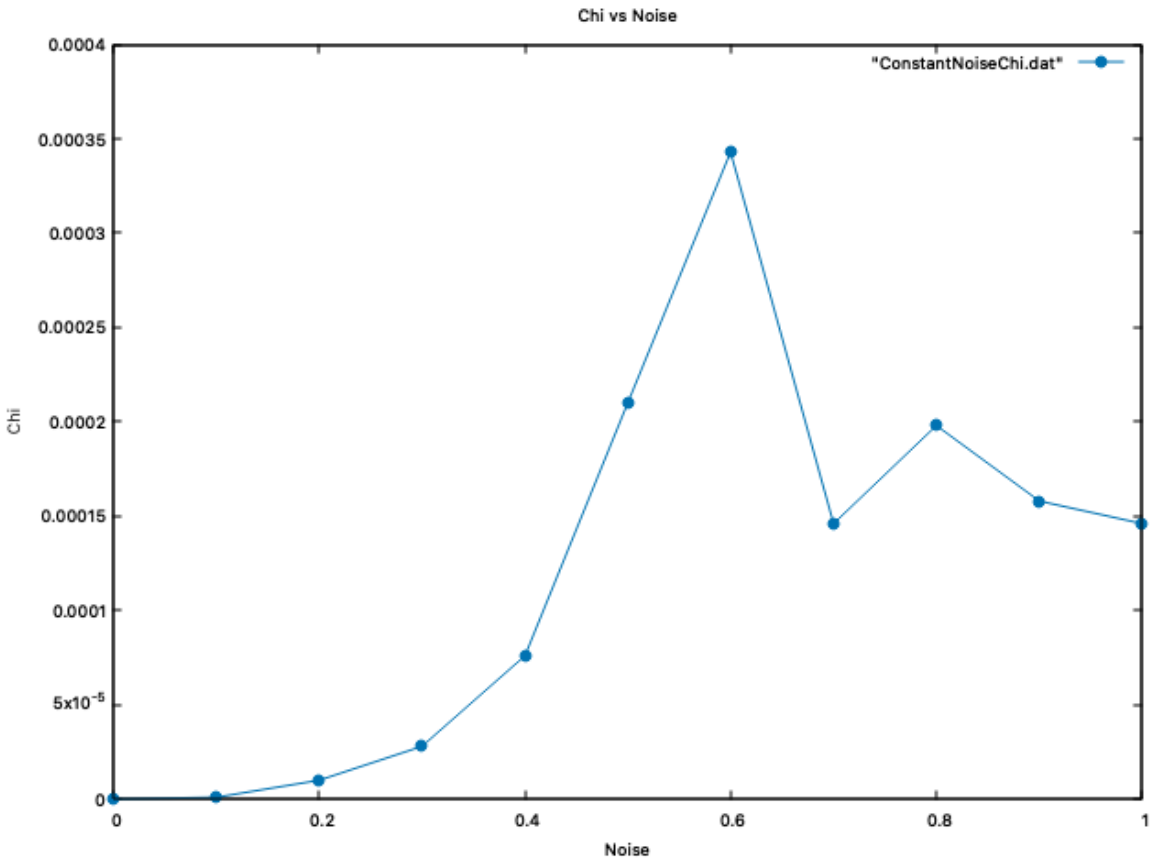
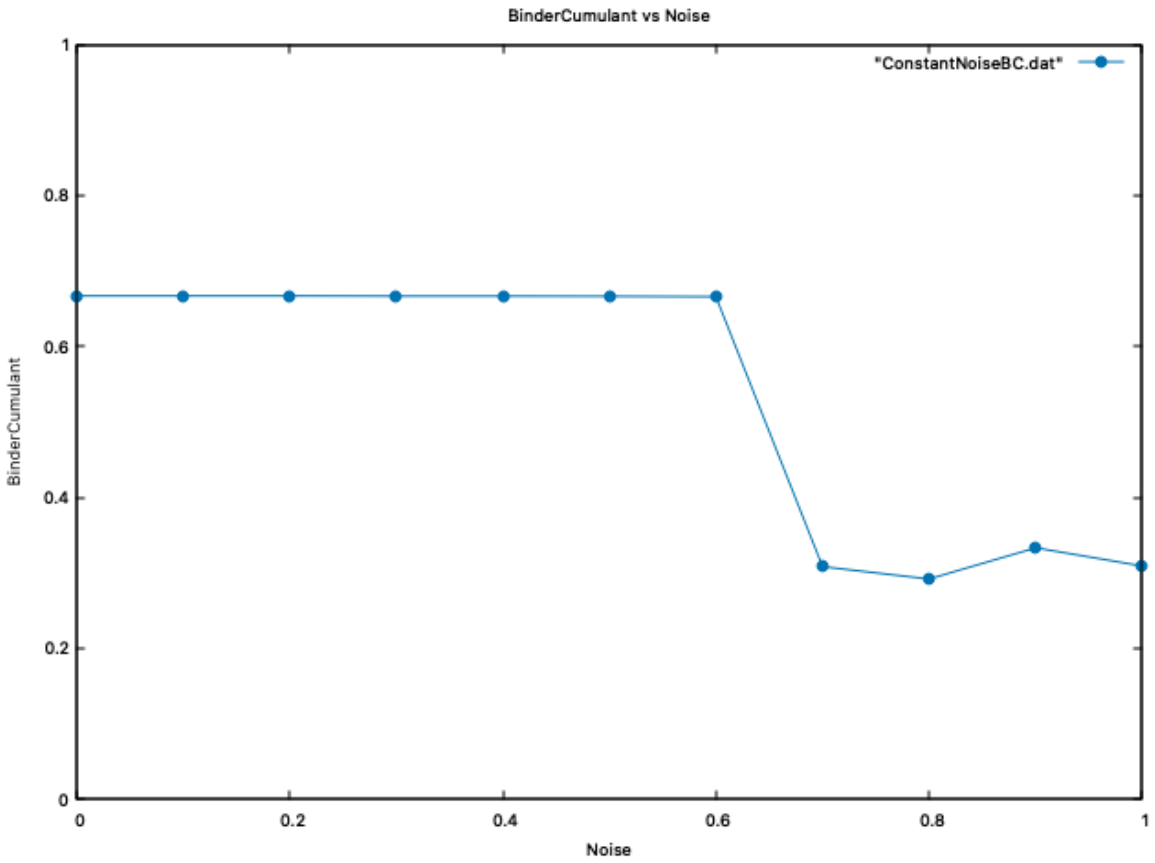


Figure 4.2: t = 10,000

Now I am attaching all the plots that we have plotted.







4.1.1 Discussing about the results

Order Parameter

We are seeing that order parameter drops sharply to almost zero value at $\eta = 0.7$, there is phase transition from ordered to disordered phase, However, the order parameter is not always well-defined at the critical point due to the divergence of the correlation length. In practice, the critical noise level is often estimated by measuring the susceptibility of the system to changes in the noise parameter, as described in my previous response. The susceptibility is expected to diverge at the critical point, allowing for the determination of the critical noise level.

Susceptibility

In the χ vs η plot we see a peak at $\eta = 0.7$ which shows that this is a 2^{nd} order-disorder phase transition, The susceptibility is expected to diverge at the critical point, allowing for the determination of the critical noise level hence we are able to find that $\eta_c = 0.7$ or somewhat near to this.

Binder Cumulant

In the plot of G vs η, we find out that at the critical noise level where the system undergoes the phase transition from the ordered phase to the disordered phase, Binder

cumulants are expected to have universal values, regardless of system size or other parameters. This is a result of the pronounced scaling behavior near the phase transition.

4.2 Time-Dependent Noise

In this part of the simulation we are running our simulation with time-dependent value of noise.
We are taking η(t) and running our simulation to get values of corresponding parameter.

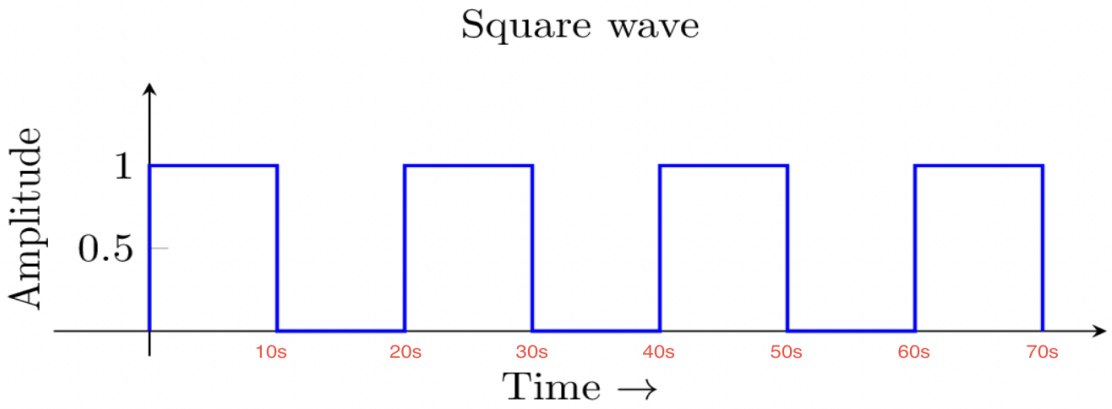
Afterwards we are plotting the variation of theses parameter vs noise, we are plotting ϕ vs η, G vs η and χ vs η.

These are simulation parameters:

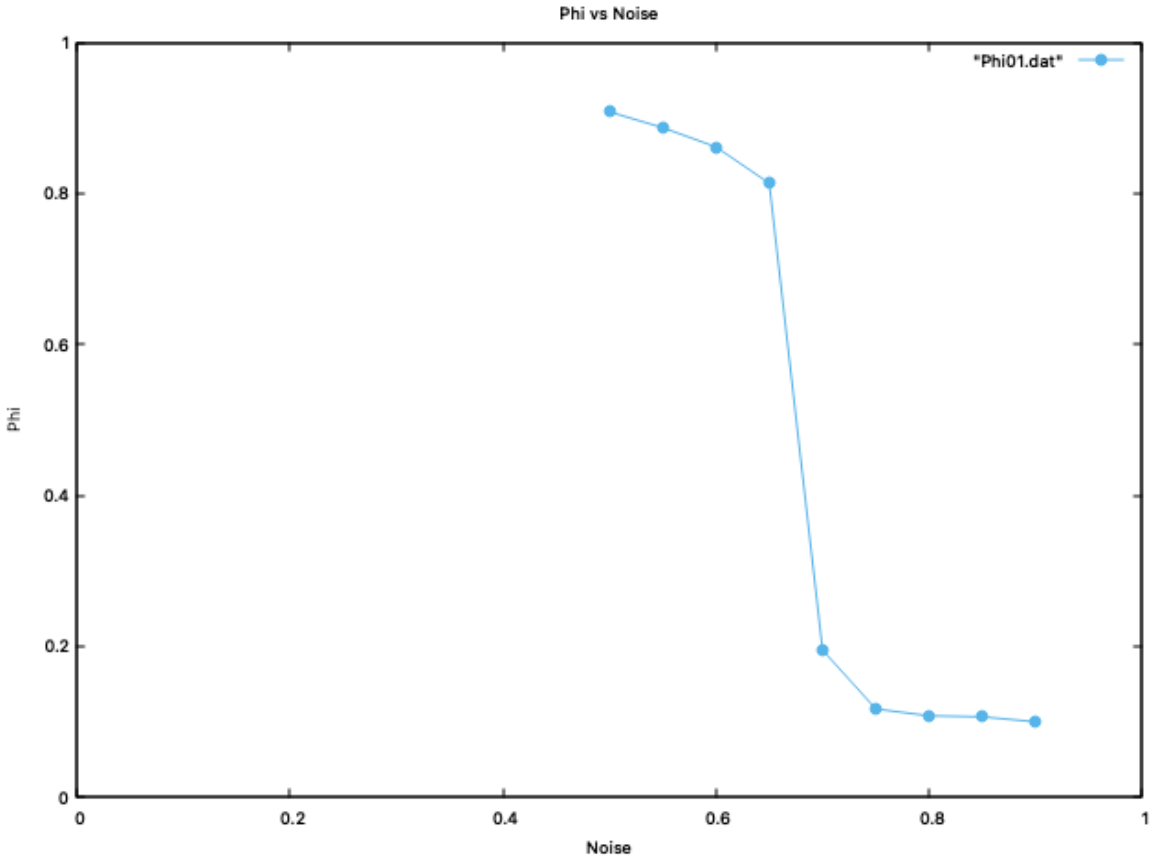
1. Simulation Box: L = 32, N = 2×32×32, ρ = 2 unit^{-2}, R = 1 unit, v_0 = 0.3, Saturation time = 100000, Time = 100000, Total time = 200000

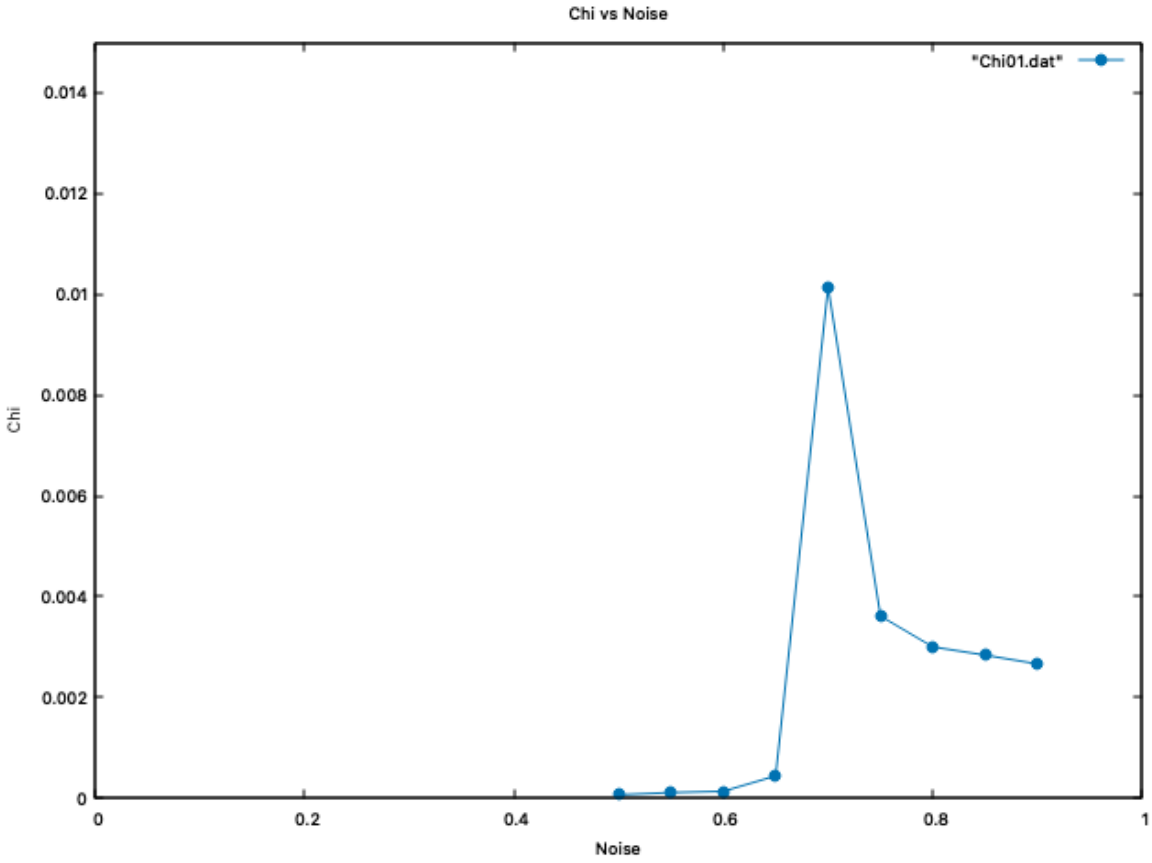
4.2.1 Square Wave Function $\eta \in [0, 1]$

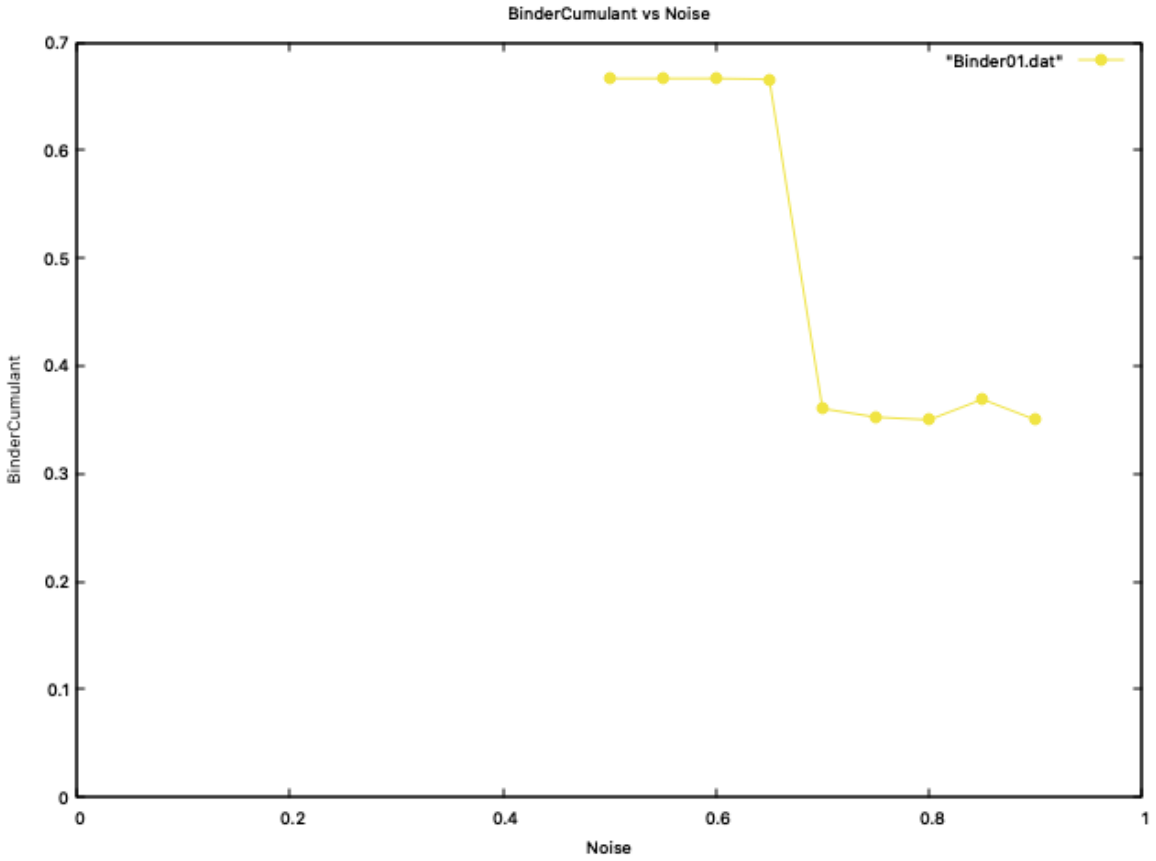
We have out timeperiod as 20s in the simulation for this case.
Here is the plot of the function with time



Now i am attaching all the plots. We are plotting around η = 0.55, 0.60, 0.65, 0.70, 0.75, 0.8, 0.85 and 0.90


Phi vs Noise
"Phi01.dat"
Phi
Noise
0
0.2
0.4
0.6
0.8
1


Chi vs Noise
"Chi01.dat"
Chi
Noise
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.2
0.4
0.6
0.8
1



4.2.2 Discussing about the results

Order Parameter

In this case to we are seeing that order parameter drops sharply to almost zero value at $\eta \sim 0.7$, there is a little change in critical noise, but in this case order parameter is not approaching zero sharply.

Susceptibility

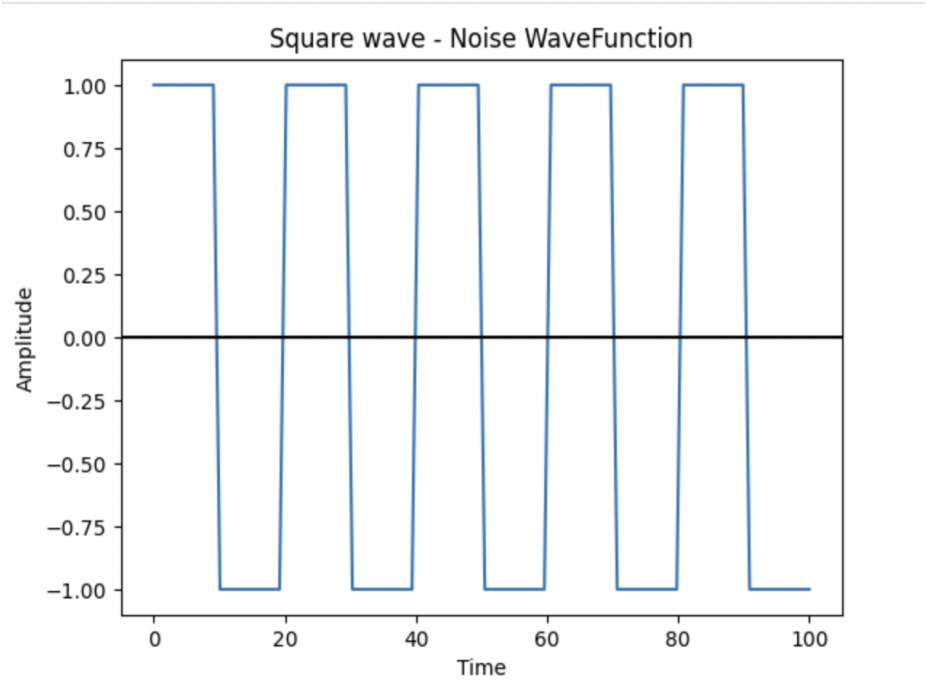
In this case susceptibility does diverge but not very shaprly.It is somewhat more compressed, but the nature of the plot remains the same.

Binder Cumulant

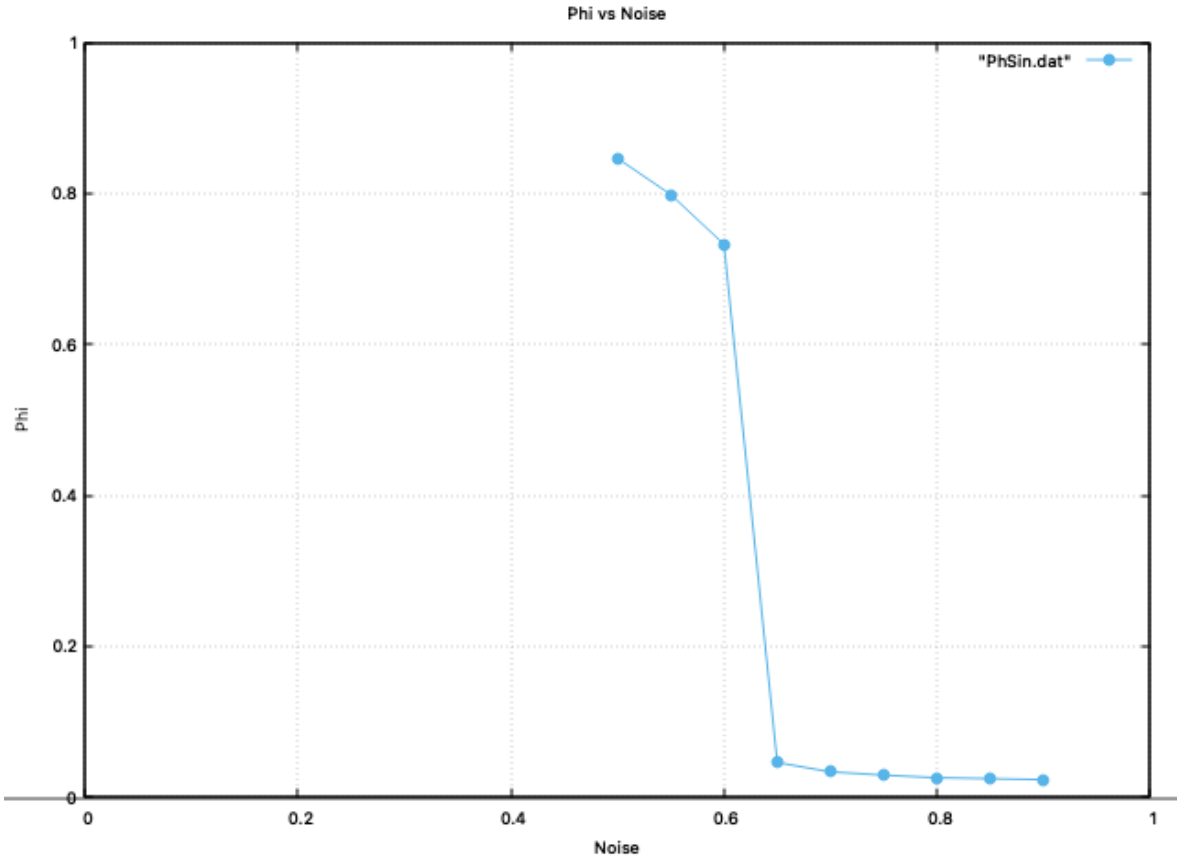
In the plot of G vs η, We see that the critical noise is shifted little bit towards right side ie, $\eta \sim 0.7$, Nature of the plot is still the same.

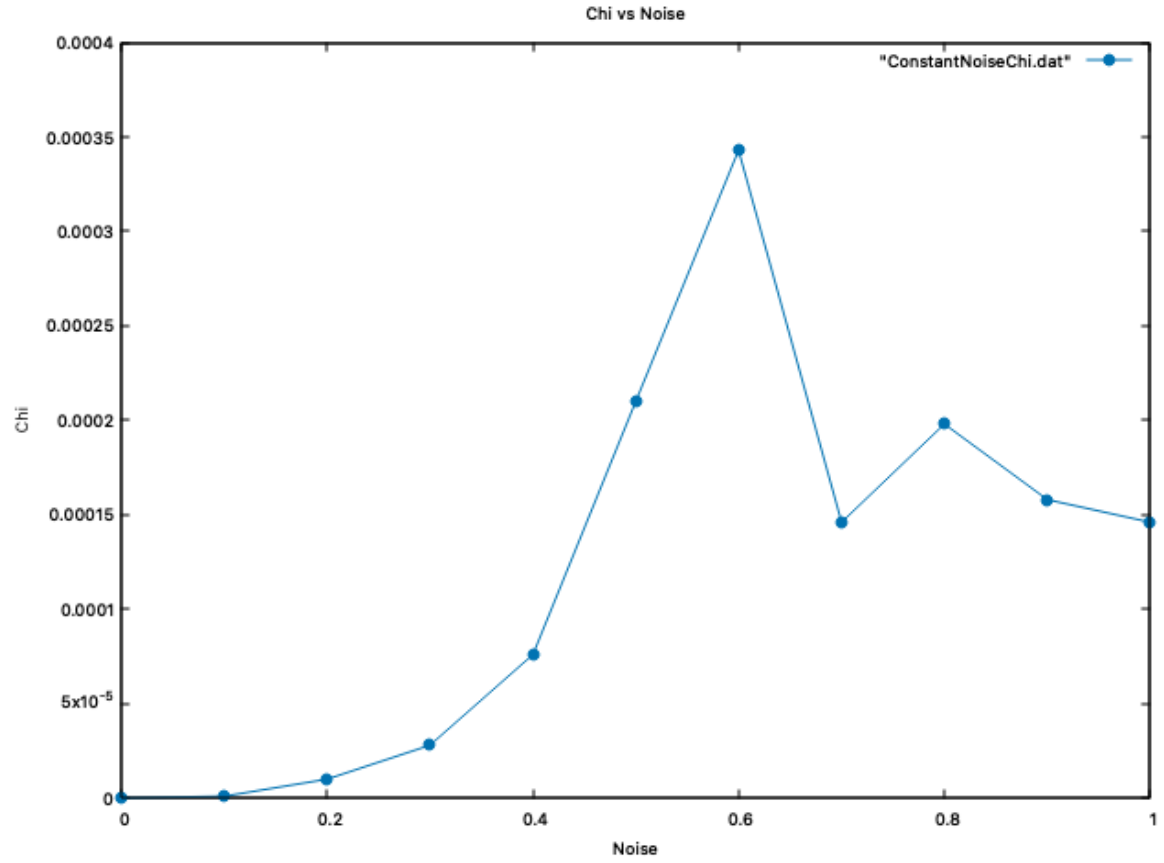
4.3 Sinusoidal Square Wave Function $\eta \in [-1, 1]$

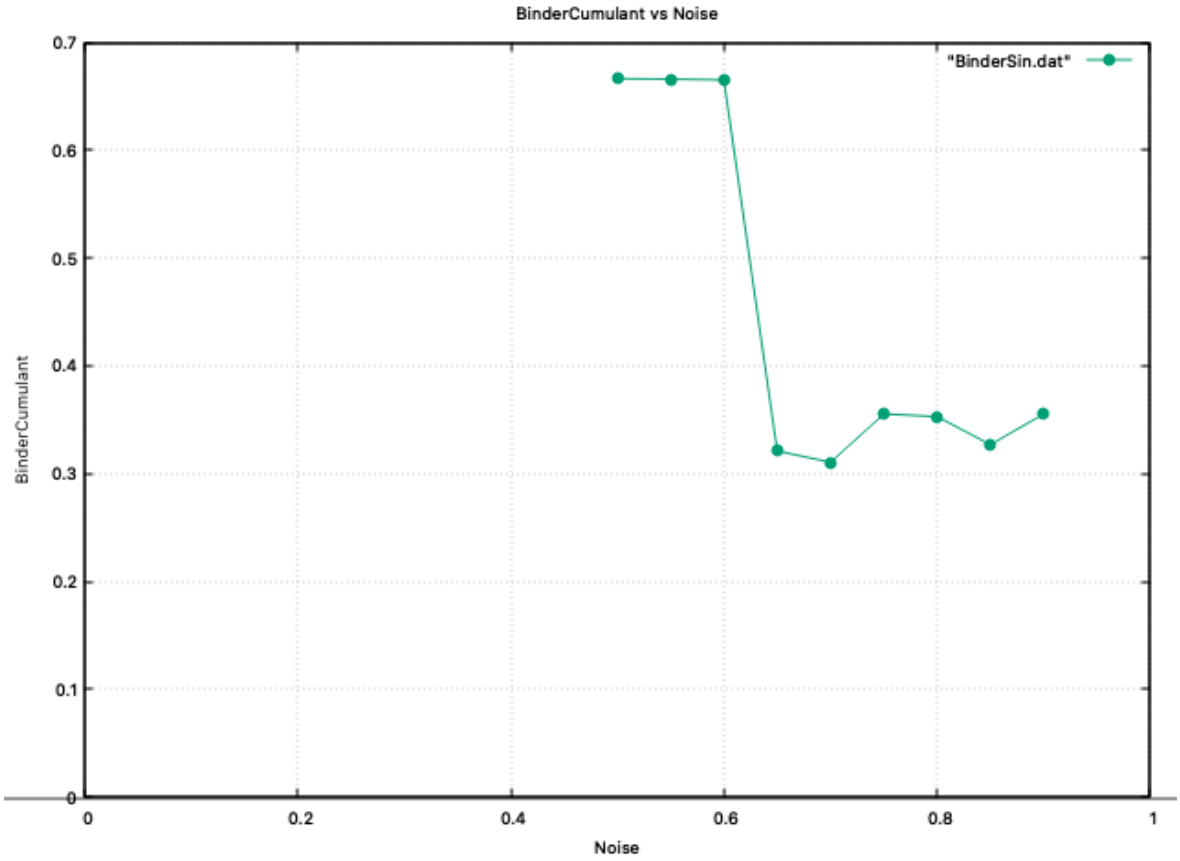
Here is the plot of the wave-function,



Here are all the plot that we get after running the simulation.


Phi vs Noise
"PhSin.dat"
Phi
Noise
0
0.2
0.4
0.6
0.8
1


Chi vs Noise
"ConstantNoiseChi.dat"
Chi
Noise
0
0.0001
0.00015
0.0002
0.00025
0.0003
0.00035
0.0004
5x10^-5
0.2
0.4
0.6
0.8
1



4.3.1 Discussing about the results

Order Parameter

In this case to0, we are seeing that order parameter drops sharply to almost zero value at $\eta \sim 0.7$, there is a little change in critical noise, but in this case order parameter is not approaching zero sharply.

Susceptibility

In this case susceptibility does diverge but not very sharply.It is somewhat more compressed, but the nature of the plot remains the same.

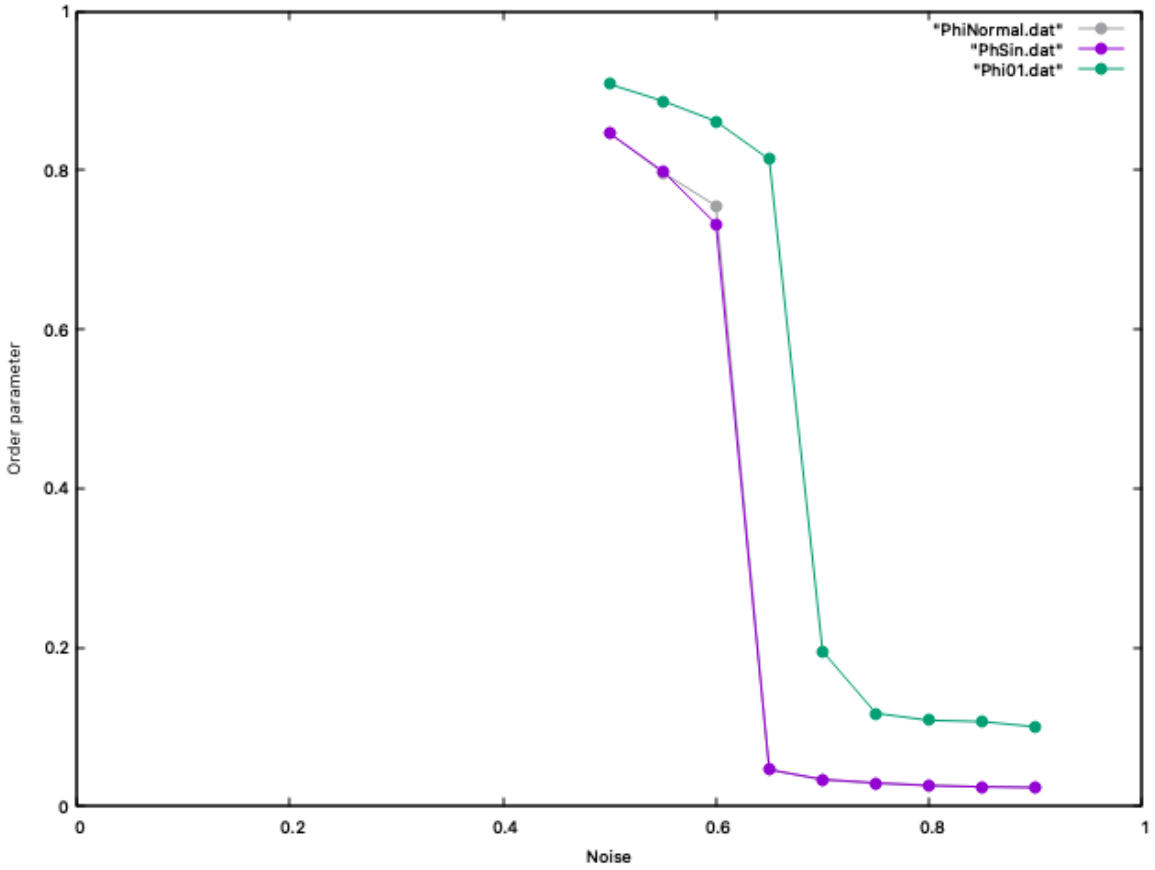
Binder Cumulant

In the plot of G vs η, We see that the critical noise is shifted little bit towards right side ie, $\eta \sim 0.7$, Nature of the plot is still the same.

4.4 Comparisons and Discussions

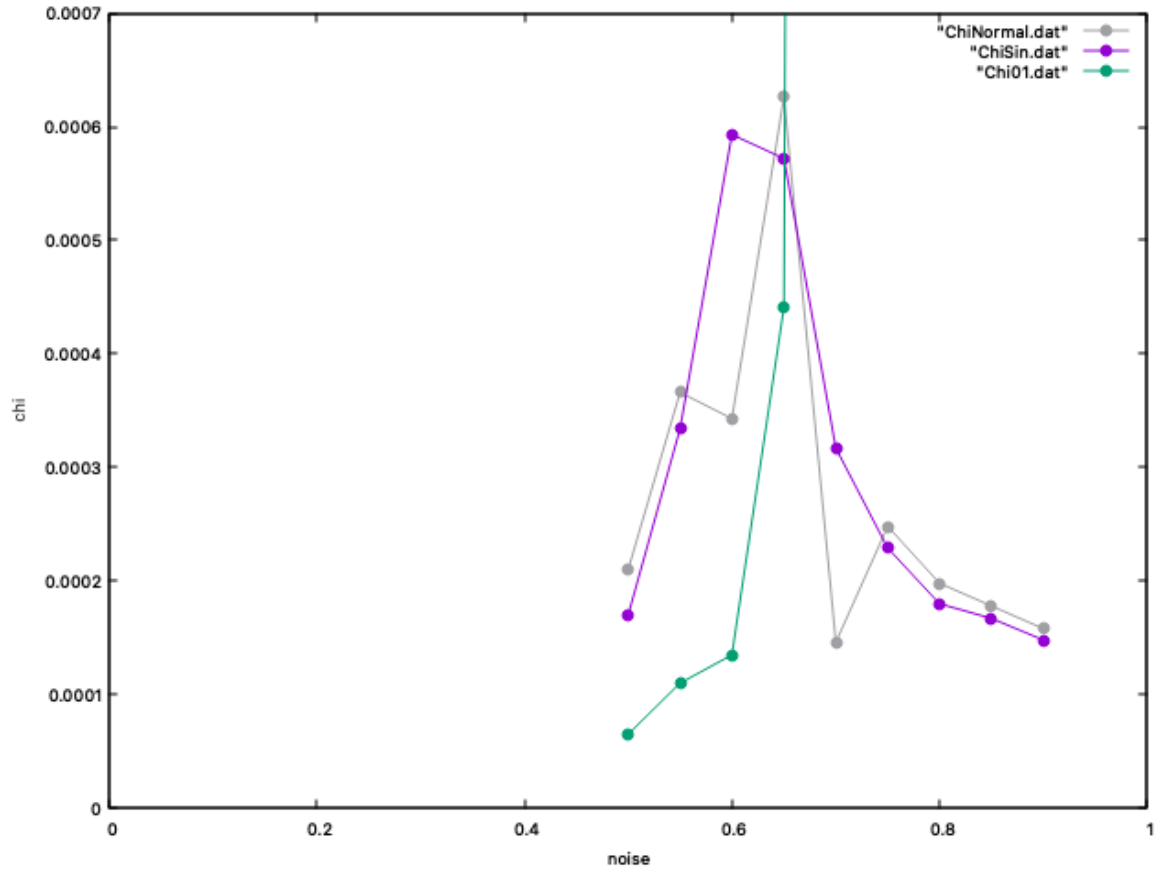
4.4.1 Order Parameter

Here we have plotted all plots at once for previous three cases:



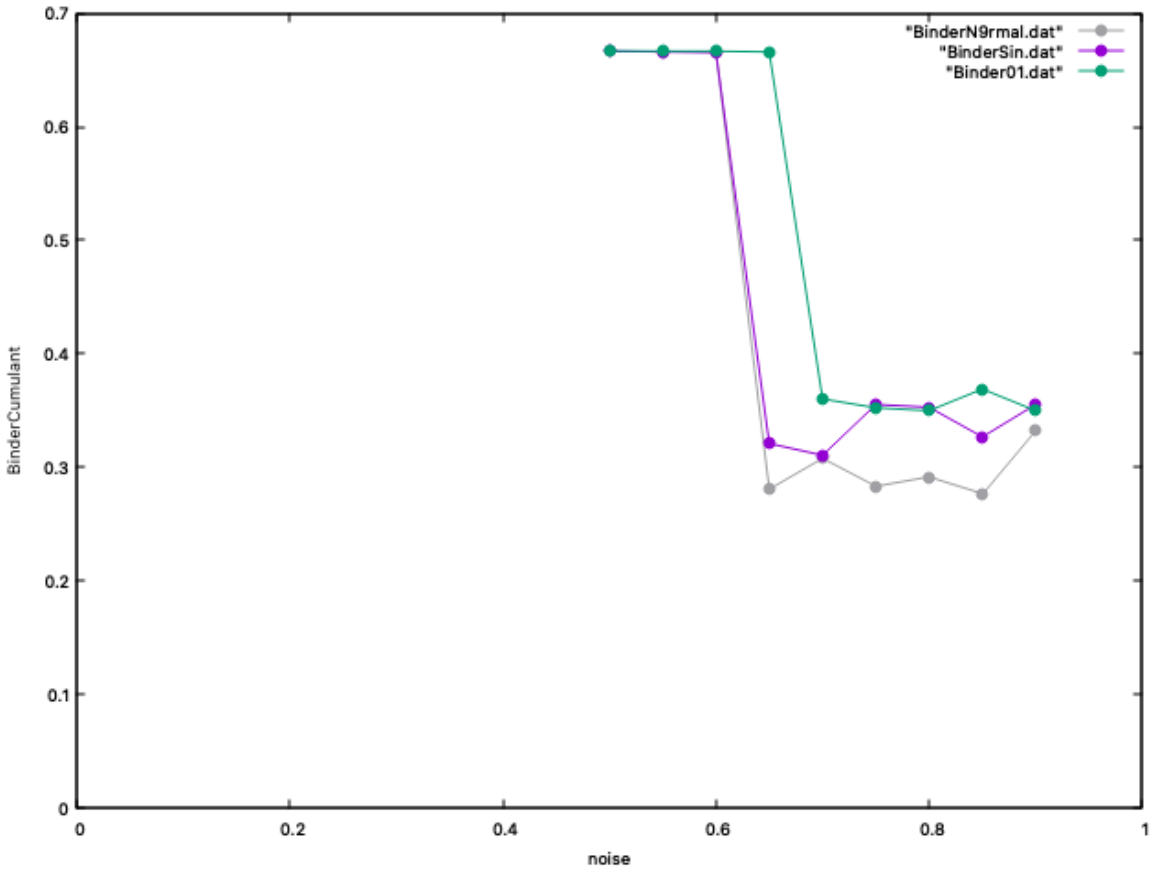
4.4.2 Susceptibility

Here I have plotted all plots of susceptibility for previous three cases:



4.4.3 Binder Cumulant

Here I have plotted all the plots of G vs Noise in a single plot:



4.4.4 SUMMARY

When we combined all the plots together we find out that there is not very much difference in Sinusoidal Time Dependent Noise and Constant Noise, Chi parameter in case of Sinusoidal Diverges much sharply in this case,

Then we have Square-Wave Function it has slightly shifted critical value of noise, but overall the shape of all the plots remains the same.
There are some fluctuations in plots of parameter this is due to the small size of the system and small numbers of swimmers too.

4.4.5 REFERENCES

1. .T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett., 1995,

2. Epl, 135 (2021) 48003, Adhikari, Santra, Effect of trapping perturbation on the collective dynamics of self propelled particles.

3. Ginneli, Epl-jst (2016), The Physics of Vicsek model.

4. Vicsek T, Cziro k A, Ben-Jacob E, Cohen I, Shochet O. 1995. Phys. Rev. Lett. 75:1226–29