Recreating the Czirók One-Dimensional Self-Propelled Particle Model

CBE 448 Independent Project

Assimakis A. Kattis

Introduction

In this project, I have tried to recreate the models constructed in the first three chapters of Yates' paper [4] that have as primary objective to describe the behavior of locust group dynamics. I began by simulating discretized Brownian motion using the Euler-Maruyama integration method and continued by modelling an SDE that approximates the experimentally observed behavior of locust movements with data that was given in Section 2.1, using stochastic integrating techniques[1]. After recreating the behavior of this SDE along with the associated double-well potential, I computed, using Yates' methodology, the values for the switching times in the alignment of the locusts that were derived in Sections 2.7 and 2.8. In order to verify my results, I used Java to construct a simulation, similar to that performed by Yates, that measured the first passage and mean switching times for the double-well potential model used throughout the chapter.

In recreating the data found in Section 3, I used MATLAB[2] to simulate the Czirók Self-Propelled Particle (SPP) model that Yates uses as the primary model of locust behavior. I restricted my work to one dimension, and used the model to investigate the variation of particle alignment with Interaction Radius (IR) and α-weighing of the different particles in the model as seen in Section 3.1. In continuing, I also applied the Equation Free methods described in Section 2.9 to this model and managed to obtain graphs that related the alignment ζ with the Drift and Diffusion coefficients of an underlying SDE. In computing the values of these coefficients, I used two distinct methods. I firstly constructed artificial values for ζ in what Yates calls Short-Term Equation-Free Analysis both for the One-Dimensional SPP model and for the Full-Interaction SPP model. Then, I continued by using Long-Term Equation-Free Analysis by running simulations of the Czirók model over a prolonged period of time and sampling different values of ζ to construct the drift and potential. Finally, the two equation free methods were compared and their respective feasibilities discussed.

2 Stochastic Switching

2.1 Brownian Motion

In the this section, a method for modelling Brownian motion was demonstrated. I managed to recreate this by discretizing time in N timesteps over the unit interval and using the *randn* pseudorandom Gaussian generator MATLAB function to calculate each of the dW increments with the formula $dW = \sqrt{dt}X$ where $X \sim N(0,1)$. I obtained the graphs in Figure 1 by using the code *brownianPath.m* based on[1].

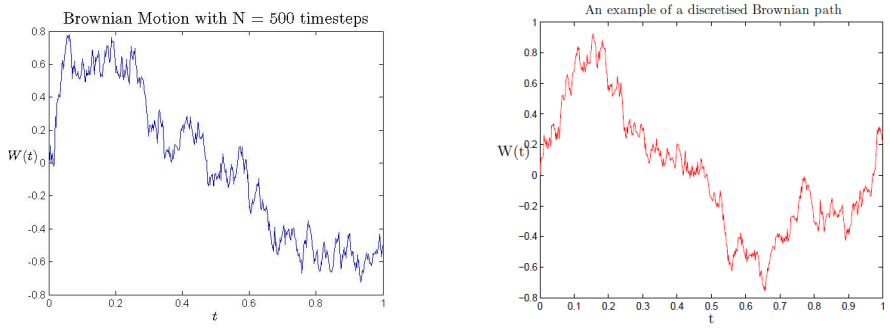


Figure 1: Brownian path generated using *brownianPath.m* compared to that found in [4].

2.3 An Application of the Euler Maruyama Method

In Section 2, Yates numerically solves an SDE with the potential $\phi(x) = 10(\frac{x^4}{4} - \frac{x^2}{2})$ by using the Euler-Maruyama method and demonstrates that its solution curve is similar to the instantaneous alignment Φ^t of the locusts in the experiment described. The periodic solution that he finds can be obtained by writing the equation in differential form with its drift and brownian terms visible.

$$dX(t) = f(X(t))dt + g(X(t))dW \tag{1}$$

The above are linked through the relation $\phi^{'}(x) = -\frac{f(x)}{D(x)}$ where D is the diffusion term and follows the relation $2D = g(x)^2$. In Yates' example, the SDE had values $f(x) = 10(x^3 - x)$ and $g(x) = \sqrt{2}$, since the diffusion constant had been set to 1. It should also be noted that in arriving at the numerical solution below, I used the initial condition $X(0) = 1$ that was also used in the paper. The code for the numerical solution below along with the potential used can be found in *myswitch.m* and is based on code found in [1].

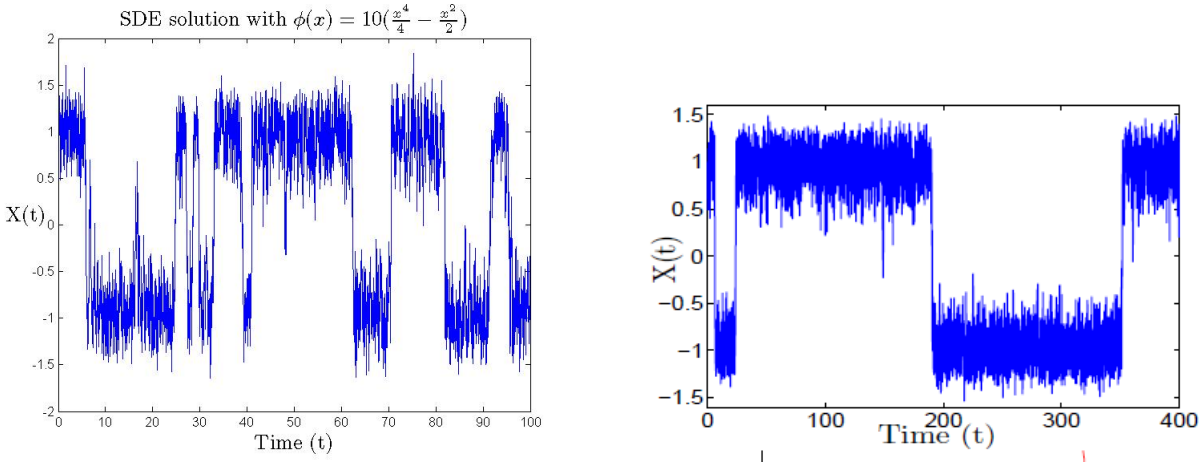


Figure 2: The solution to the Section 2 SDE alongside the corresponding solution in [4].

I observed a difference in repeated trials between my numerical solution and that observed in [4]. The numerical solution that I derived seems to have a more frequent switching frequency, something that can be observed from the decreased time that it takes the solution to move between -1 and $+1$, accentuated above since the axes used were of different scales. Since the drift and diffusion terms I used were the same, there should exist no discernable difference between the two. However, in changing the potential of my SDE, I observed a much greater similarity between Yates' SDE and that for $\phi_2(x) = 2.5\phi(x)$, whose solution graph is shown below next to the two different potentials used superimposed on the same axes.

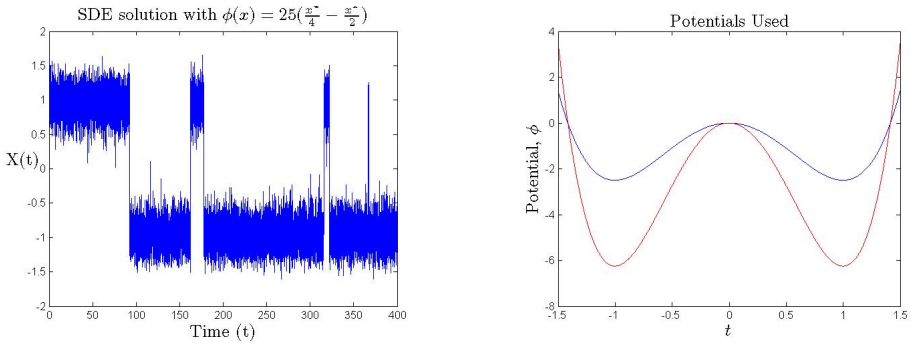


Figure 3: On the left is the solution graph of the accentuated potential, exhibiting similar behavior to that described by [4]. On the right are the two potentials superimposed on each other, with the steeper (red) line denoting the higher potential

2.8 Moments of First Passage Time

In this section, I verified Yates' analytical derivations of first passage time through computation and simulation. I calculated the times required based on the three different modes of approximation described in Section 1.8 of [4] that were made using the program *switchingtimes.m*, obtaining the same results in the integral computations.

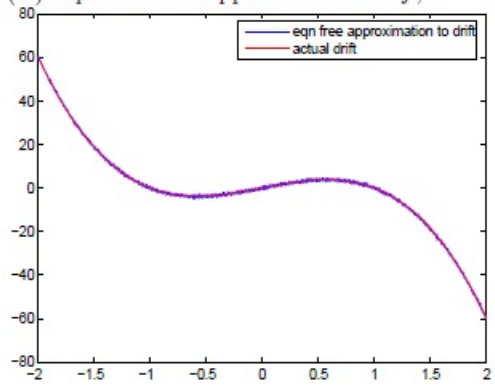
I also tried to simulate first passage time using Java, but it was so that the simulation took too much time to arrive at the solution within a reasonable bound of accuracy without having to alter the value of Δt. More specifically, I coded *Switch.java* and ran the simulation using different timesteps, all of which gave me different bounds on the error. The result derived with the smallest timestep and largest bumber of repeats is displayed below alongside the results from respective formulas, found in the same section.

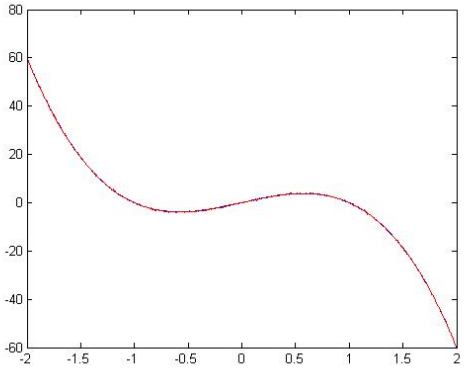
Method of Evaluation	**Mean Switching Time**
Average over 100,000 repeats with $\delta t = 0.0001$	3.096
Formula (2.8.4)	3.082
Formula (2.8.5)	3.313
Formula (2.8.6)	3.082

2.9 Equation-Free Analysis

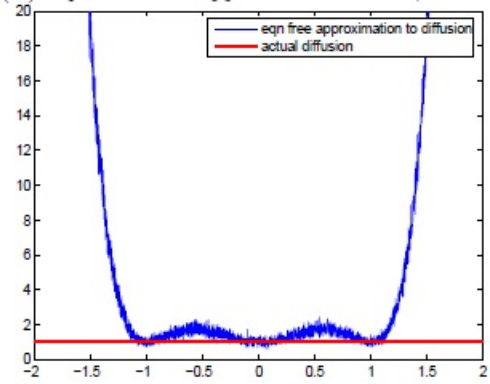
In this section, I simulated the SDE that Yates uses to demonstrate the efficacy of the equation-free method. In doing this, I followed a very similar strategy to that of Section 2.4 in [4], creating the script *eqnFreeExampleSectionTwo.m* that calculates the values of the SDE and which, using the equation-free techniques analyzed in this section, calculates approximations to the drift and diffusion terms. After doing so for the respective Δt values, it also numerically integrates their quotient using Euler's Method to arrive at values for the potential of the SDE. I have added more information on the exact methododology of the equation-free method in Section 3.

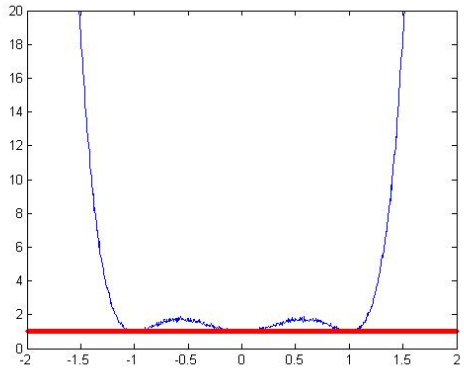
(A) Equation free approximation to f, $\delta t = 0.1$





(B) Equation free approximation to D, $\delta t = 0.1$





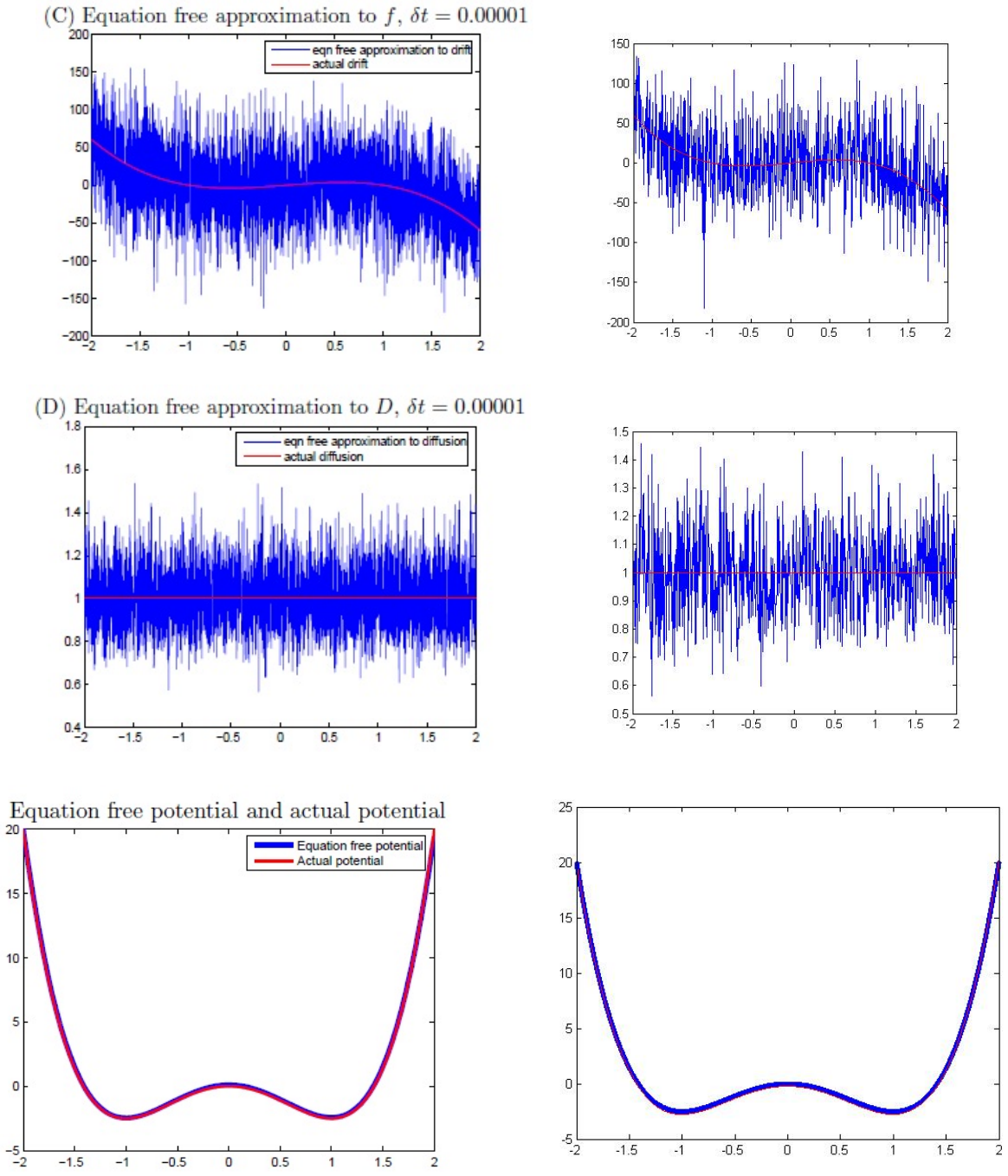


Figure 4: Above can be seen, on the left, the plots from [4] in comparison to those that I derived using the same equation-free techniques. The first four plots show the solutions for f and D respectively when $\delta t = 0.1$, while the next two show the same for $\delta t = 0.00001$. Finally, the derived potentials are also shown above, where the most accurate values for f and D found previously were numerically integrated to arrive at the potential. Similarly to Yates, I chose a constant of integration such that the graph of the analytical and the equation-free solution coincided. The blue here denotes the analytical while the red the equation-free solution.

3 Self Propelled Particle Models

3.1 Variation of Alignment with Interaction Radius

In this section, I built the core of the Czirók SPP model. I began by coding the scripts $Q.m$ and $G.m$ that modeled the respective functions defined for the model. I fixed the values of the two constants to those used by Yates throughout the paper at $\beta = 1$ and $\xi = 3$ [4] but allowed for the modification of the latter, with the $Q(\Delta t)$ function being defined for $\xi = 3\sqrt{\Delta t}$ as is mentioned later on in the paper. I did this because modifying the timestep in the simulation is something that becomes important when dealing with the equation free apects of the simulation. When creating the random variable $Q(x)$, I made use of the formula below for a uniform random variable defined between a and b while defining ξ with respect to time.

$$X = b + (b - a)U \tag{2}$$

where $U \sim U(0, 1)$ and $X \sim U(a, b)$ with $U(0, 1)$ being modelled by the *rand* function.

In terms of the model itself, I created a script *czirok.m* that emulated its behavior with all of the given parameters variable. I created it by firstly initializing the positions of the particles pseudorandomly uniformly on the space and using $Q(\Delta t)$ to assign random values to the initial velocities, since no specification was made by Yates in the paper on initial velocity. In checking whether these initial conditions made a difference to the overall behavior, I also checked for behavior with zero initial velocity, but did not find any serious differences between the two. The model was programmed to output the values of the positions and velocities of all the given points in array form while also returning a value for ζ at each given timestep.

I then investigated the values for $\zeta_{overall}$ that my simulation returned while varying r at increments of 0.5, where

$$\zeta_{overall} = \frac{\sum_{i=\frac{N}{2}+1}^{N} |\zeta(i)|}{\frac{N}{2}} \tag{3}$$

a measure of the overall alignment observed for a specific simulation over a discrete finite period of time[4]. I created a function *zetaOverall.m* that ran the simulation once at specified parameters and returned the associated $\zeta_{overall}$ value. I then ran this through the script $fig31.m$ for specified increments of r values and for a fixed number of repetitions. I initially performed the simulation 10 times for each r but realized that the uncertainty was too high to give me smooth enough results. After working at different values for the repetitions while keeping the values for the rest of the parameters as specified in Section 3.1, I observed that for an average of 500 repetitions at each point value the simulation provides data that is extremely accurate with respect to that observed in [4]. The two graphs are shown below for comparison.

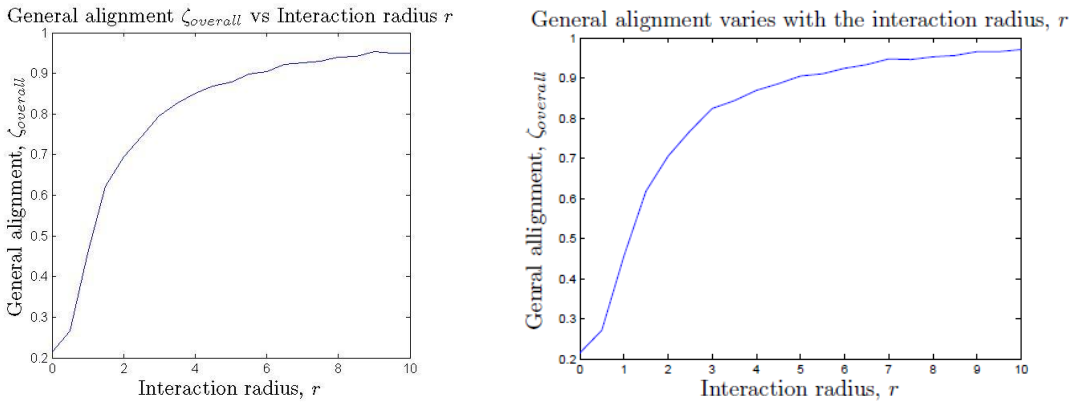


Figure 5: On the left is the plotted overall alignment calculated by averaging over 500 repetitions at different values of r using the script $fig31.m$ and on the right is the plot found in [4].

3.2 One-Dimensional Model

After analyzing the variation of $\zeta_{overall}$ with r, I looked at the extent to which the model could be improved by investigating the extent to which the influence of a particle's own velocity on itself affects the overall behavior of the model. Because this is unrealistic and due to the fact that the initial update equation for each particle always takes itself into account, a model where this is avoided needs to also be constructed. To do this, I noted the generalized one dimensional model defined by Yates in this section[4] which introduced a weighting parameter α to the velocity update equation. This parameter defined the extent to which a particle takes its own velocity into account. The generalized equation is shown below.

$$u_i(t+\Delta t) = u_i(t) + \Delta t[G(\frac{\alpha}{n(t)+\alpha}u_i(t) + \frac{1}{n(t)+\alpha}[\sum_{j\in \jmath^r} u_j(t) - u_i(t)]) - u_i(t)] + \Delta Q \tag{4}$$

The function n is defined with respect to the number of neighbors of particle i excluding itself.

$$n(t) = |\jmath^r| - 1 \tag{5}$$

It should also be noted that, in order to both more accurately simulate the results observed in the paper and deal with a specific corner case in a way that would make sense on a physical level, I decided to assume that, for $n(t) = 0$ and $\alpha = 0$, the update equation would be given by $u_i(t+\Delta t) = u_i(t) + \Delta Q$, since there would be no interactions with any neighboring particles.

Based on this refined model, I created two new functions, *czirokAlpha.m* and *zetaOverallAlpha.m* that generalized the previous model by adding the α parameter to the simulation and calculating the ζ and $\zeta_{overall}$ values respectively for a simulation. Then, I created the script *alphaIR.m* which calculated the change in $\zeta_{overall}$ as a function of α and r, creating a surface plot of the results, displayed below. In order to arrive at more accurate values for $\zeta_{overall}$, I decided to repeat its calculation at each value of α and r 100 times and to plot the mean of the results. The smoothness of the plot and the lack of deviation across α agrees with Yates' conclusion that this parameter does not substantially influence the long term behavior of the model[4].

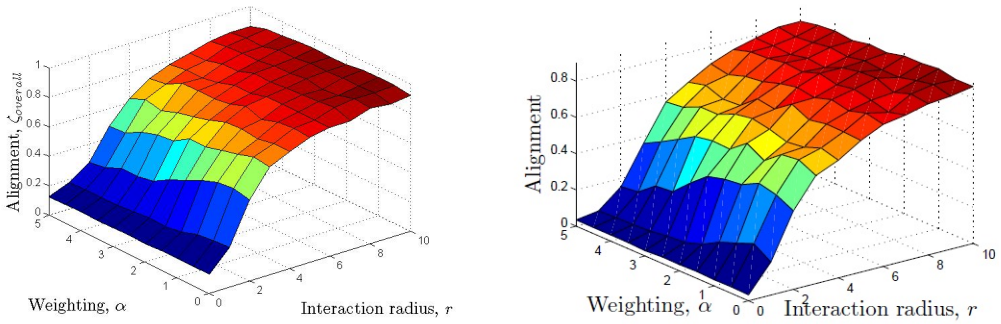


Figure 6: Above on the left is the surface plot I obtained by running the simulation with *czirokAlpha.m* 100 times for each value of α and r and on the right is the same plot as seen in [4].

3.3 Application of the Equation-Free Method to the One-Dimensional Model

Using the model that I built in *czirok.m* (with $\alpha = 1$) I used the initial conditions specified by Yates to simulate the behavior of ζ with respect to time. I simply let the above script run with $r = 4$, $dt = 1$ (which implies $\xi = 3$) and for 30 particles on a length 90 domain, giving a value of $\rho = \frac{1}{3}$ for the density. In running the simulation, I used the script $figure34.m$, which executes *czirok.m* for the above parameters over a time period $T = 2000$ and subsequently plots the values of $\zeta(t)$ and $x_i(t)$ versus time. The results that I got from this mirror those observed in [4], also shown here for comparison.

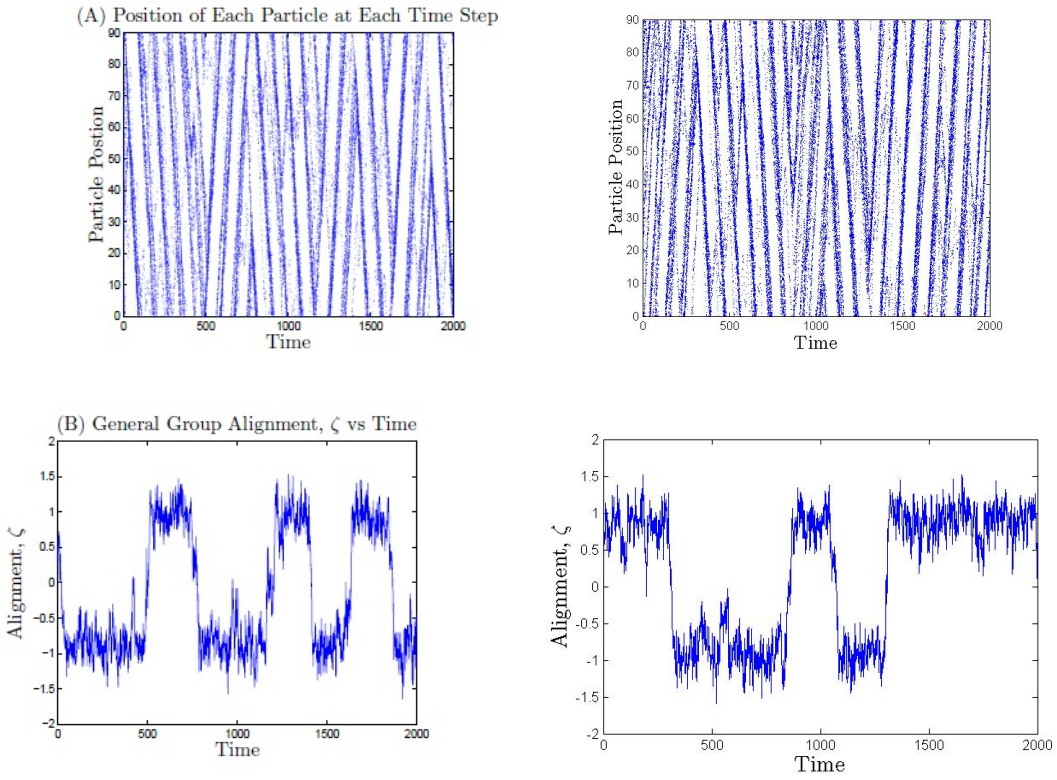


Figure 7: On the left are the plots found in [4] and on the right are the same graphs that I obtained through the script $figure34.m$. The graphs on the top model the positions of the particles at each time step, showing a definite alignment in the positionis with time, while the bottom graphs model the value of ζ with time, clearly demonstrating its periodic nature. These graphs draw direct links between the behavior of the Czirók model and of the SDE modelled in Section 2.

The next graph that I generated involved a minor modification of the second figure above. I changed the initial conditions in the model so that the SDE I was modelling would have a density $\rho = 1$, showing what the qualitative difference is in increasing the density from $\rho = \frac{1}{3}$. This change involved setting the parameter $N = 90$. The graph I obtained is shown below, alongside that found in[4] for comparison.

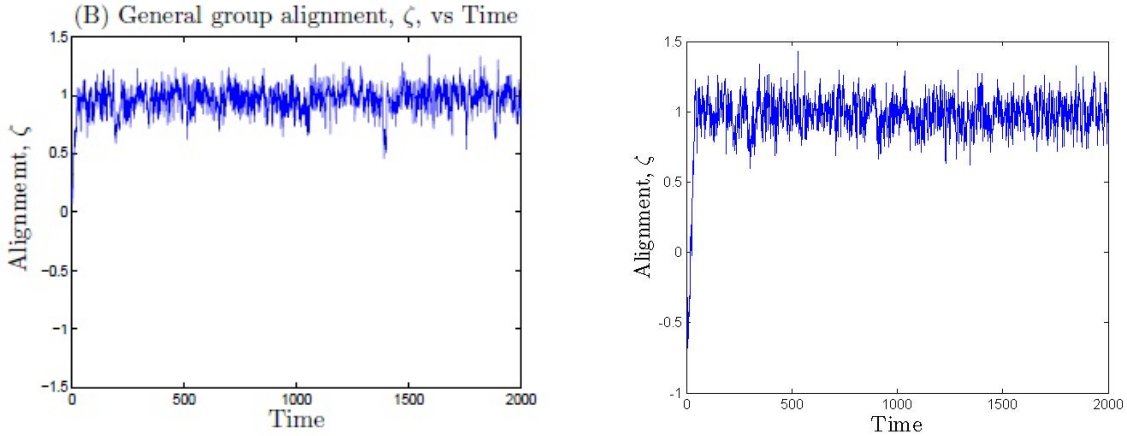


Figure 8: On the left is the graph obtained by Yates in setting the density of particles to $\rho = 1$, while on the right is the corresponding graph that I obtained from my model. Both graphs exhibit no switiching in the values of ζ, which remains relatively constant in both of the simulations.

However, it should be noted that, unlike the behavior that was observed by Yates in setting ρ to 1, namely that there is never any switching[4], I noticed that in my model, the SDE does undergo switching, but this happens much less than for the case of $\rho = \frac{1}{3}$. I have included a graph of this instance below.

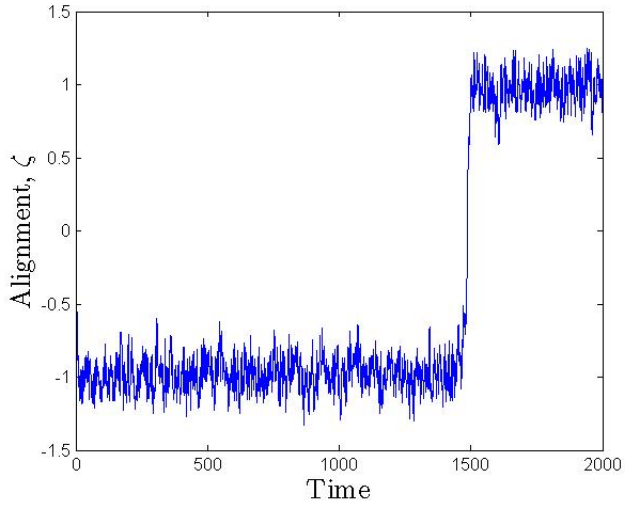


Figure 9: Above can be seen an instant where my SDE modelled with $\rho = 1$ undergoes switching between its two potential wells in contrast to the analogous behavior described in [4].

These graphs indicate that the behavior of ζ can be modelled by an SDE of the form of (1), with a double well potential $\phi(x)$. In assuming this, equation free techniques can be utilized to find this potential and model the behavior of the SDE.

In order to do this, I followed Yates' technique in generating artificial values for the positions and velocities for each of the particles such that the respective value for ζ can be fixed from the beginning (see footnote on p.37 of [4]). I used values of $N = 30$ and $L = 90$ for the model and, by fixing ζ, I randomly distributed the points in the domain. Then, I set their velocities to equal ζ, adding $\frac{N}{2}$ pseudorandomly generated noise terms to each of the first $\frac{N}{2}$ particles and subtracted the same noise terms from the rest. This ensured that ζ for this initial state was fixed and its derivative was then able to be computed. The theoretical basis for this method lies in the following limits, derived in Section 2.9 of [4].

$$f(X) = \lim_{\delta x \to 0} \left\langle \frac{X(t+\delta t) - X(t)}{\delta t} \right\rangle \tag{6}$$

$$2D = g(X)^2 = \lim_{\delta x \to 0} \left\langle \frac{[X(t+\delta t) - X(t)]^2}{\delta t} \right\rangle \tag{7}$$

However, I had to choose the values for δt with care, depending on whether I was evaluating f or D. This was because, as is explained in Section 2.9 of [4], although the above limits hold, we cannot numerically compute them, and therefore the values for δt matter most when we are trying to approximate the drift and diffusion terms. More specifically, a larger value for δt would ensure that the noise terms of the diffusion matter less in the overall computation, and therefore this is ideal for an approximation to the drift f. Similarly, smaller values for δt would allow for a much more accurate determination of the diffusion.

I created a function called *eqnfreeShort.m* that took as inputs a value for ζ, the parameters that defined the simulation, namely L, N, r and δt, and a parameter *repeats*, which set the number of times that the simulation would be repeated in going from t to $t + \delta t$. I also added an internal variable that I called *calibration*, which had as its purpose to cut the timestep δt into smaller dt values so that the simulation would be more refined and accurate. I added this value because I noticed that its introduction allowed for a much better simulation of D. I set it to 10 after experimenting with different values. Using these variables, the function constructed an artificial starting point for the simulation which had a known ζ value by using

the aforementioned construction. After the completion of each simulation, I stored the derived value of ζ in an array, using this along with equations (6) and (7) to derive the values for f and g in the end, which were then returned.

I also created a script, *czirokeqnfreeShort.m*, that allowed for the modification of the values in the previous function and, using these as tools, I was able to simulate values for the drift and diffusion terms of the model. I found that choosing $\delta t = 1$ with *calibration* set to 1 allowed for a smooth approximation to the drift, plotted below. The number of repeats was set to 100 and I sampled values for ζ between -1.5 and 1.5 at increments of 0.005 in both graphs below. It should be noted, however, that in calculating the drift D, I used a value of $\delta t = 0.01$ alongside a calibration of 10 in order to get the required graph. In trying with larger timesteps, I noticed that the graph of D became very affected by the values of f, spiralling upwards on the sides. This is in line with the behavior outlined by [4] in Section 2.9.

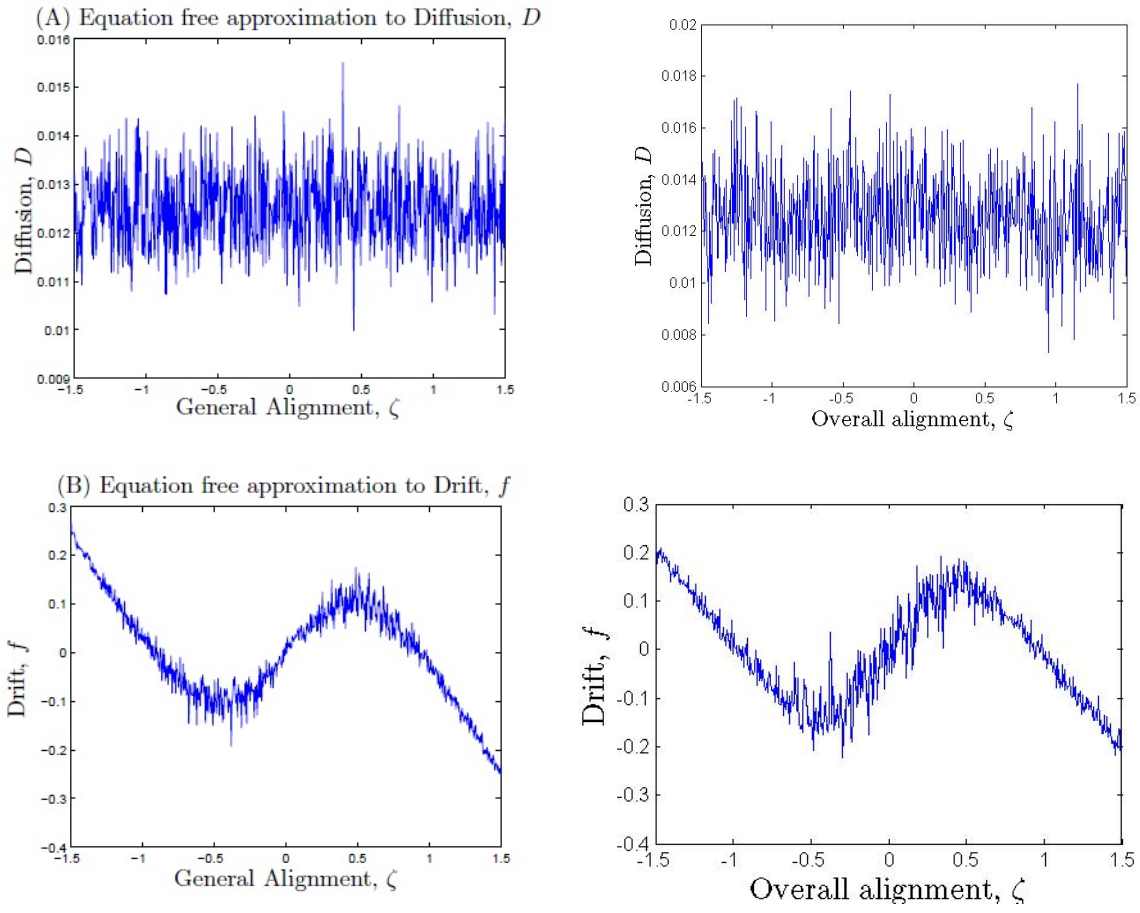


Figure 10: On the top are the approximations to the diffusion coefficient D found through short term equation free methods by Yates in[4] (left) and through my script *czirokeqnfreeShort.m* (right). On the bottom can also be seen the respective approximations to the drift of the assumed underlying SDE.

After calculating the approximations to the drift and the potential above, I was able to integrate them to arrive at the value for the SDE's potential. I followed Yates' methodology in doing this (Section 2.9 of [4]) by using the exact relation between drift, the diffusion constant and the potential, shown below.

$$-\phi'(x) = \frac{f(x)}{D(x)} \tag{8}$$

Using the discrete values that I had derived for f and D, I used Euler's method to numerically integrate the above expression to arrive at the potential of the assumed SDE, using the same timestep as before, namely $\Delta\zeta = 0.005$.

Finally, I also used the equation given in [4], Section 2.6, to compute the Stationary Probability Distribution curve of the derived potential, shown below. This was plotted against a normalized histogram[3] of data for ζ obtained by running the simulation over a long period of time, with $\Delta t = 1$ and $T = 100000$. Unfortunately, I was unable to simulate the behavior of the model for a larger value of T given constraints in the runtime of my program. However, the qualitative similarities can still be observed. In normalizing the histogram, I used the script *histnorm.m*[3] alongside the values for ζ obtained over the course of the simulation.

$$P_s(z) = \frac{exp[-\phi(z)]}{\int_{\mathbb{R}} exp[-\phi(z)]dz} \tag{9}$$

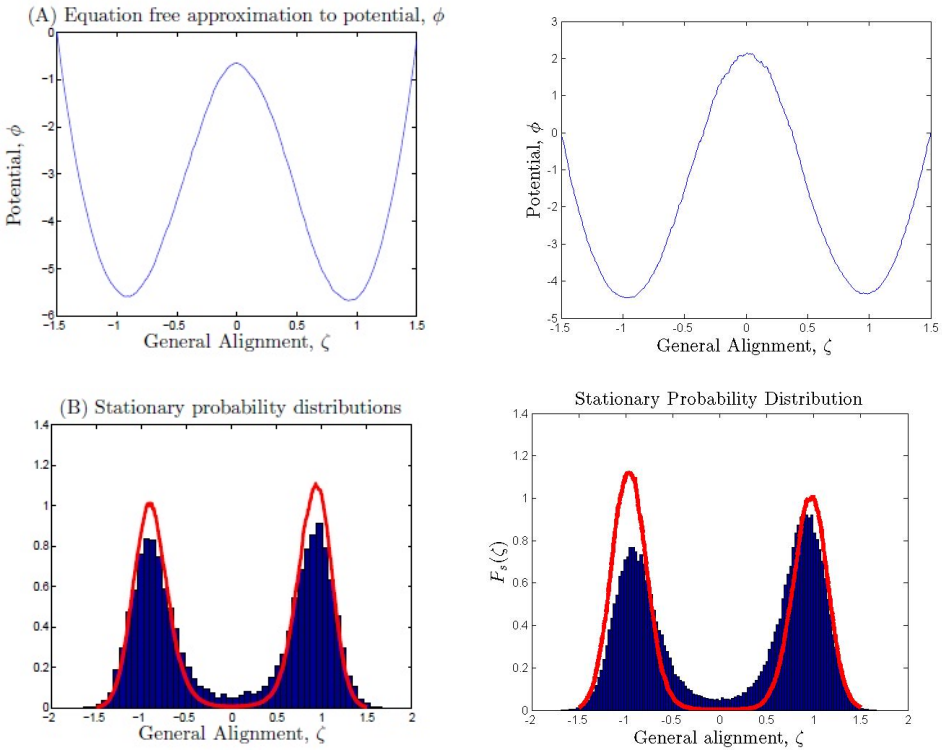


Figure 11: On top are the equation free potentials derived by Yates (left) and by using the above methodology (right). Notice the substantially higher potential that I found around zero. On the bottom can be seen the equivalent Stationary Probability Distribution (red) plotted against a histogram of data derived from allowing the simulation to run for a long time.

3.4 Application of the Equation-Free Method to the Full-Interaction SPP Model

In this section, I recreated a specific model that Yates simulates and compared my derivations to their corresponding analytical solutions. In the full interaction SPP model, the value for the radius is changed to equal $\frac{L}{2}$ so that all of the particles interact with each other throughout their movement. This leads to interesting effects that can be modelled by an analytical solution that Yates derived explicitly for the Potential and the Stationary Probability Distribution. They are shown below.

$$\phi(\zeta) = \frac{12N\beta}{\xi^2(1+\beta)}(\zeta^2 - 2|\zeta|) \tag{10}$$

$$P_s(\zeta) = C^{-1}exp[-\phi(\zeta)] \tag{11}$$

where $C = \int_{-\infty}^{+\infty} exp[-\phi(\zeta)]d\zeta$.

Using the above as benchmark solutions, I applied the equation free techniques outlined in the previous chapter to the same model, with the only difference that I had to take $r = \frac{L}{2}$ and $\alpha = 0$. After running some initial simulations, I decided to repeat the simulations 100000 times with a timestep of $\delta t = 1$ for calculating the drift and $\delta t = 10^{-5}$ for the diffusion. The final values for each are shown on the two graphs below and can be seen to be fairly accurate. I ran the simulations using the script $FIsecscript.m$ and the function $FIsec.m$.

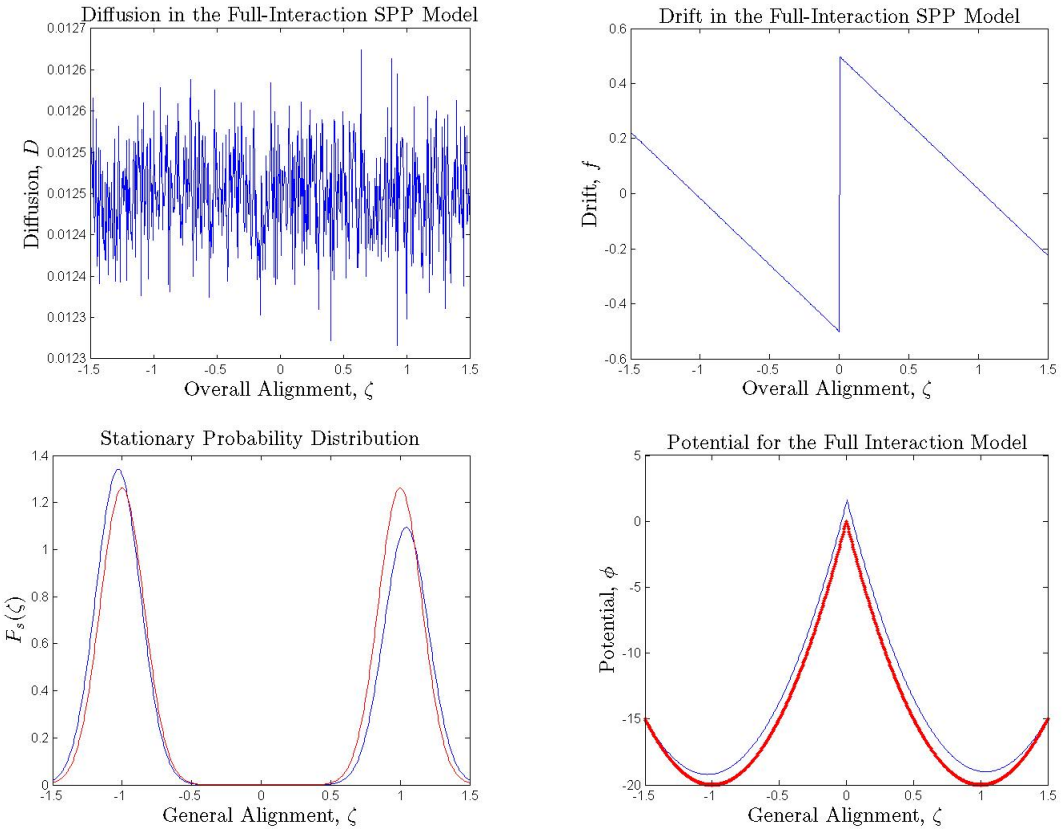


Figure 12: From top to bottom can be seen the derived drift term, diffusion coefficient, stationary probability distribuition, and potential. It should be noted that the last two also have the analytical solution expected plotted on the graph for comparison. I decided to choose a starting point for the potential which would allow for a good comparable analysis between it and the analytical solution, opting for $\phi(-1.5) = -15$. The main reasons for error here seem to stem from the integration of the parameters. I used Euler's Method to integrate the points with $d\zeta = 0.005$ but was unfortunately unable to refine this timestep, since doing so would make the simulation take a prohibitive amount of time.

3.5 Long Term Equation-Free Analysis

In this section, I recreated the derivations of the drift and diffusion without the artificial fixing of ζ. As explained by Yates, due to the artificial nature of setting specific values for each of the particles in a simulation, it is more desirable to try and model the behavior of ζ without having to alter the values of position and velocity. In this section, Yates overcomes the difficulty in generating explicit values for ζ by running a simulation of the Czirók model over a long time, obtaining data for ζ, X and U at each timestep, which was chosen as $\Delta t = 1$. Using this data, he samples different values for ζ with 'natural' values for position and velocity, progressing through the rest of the simulation in the same fashion as in Section 3.4.

In order to recreate this method, I ran the same simulation of the model, but I chose to do so only for 100000 timesteps, as opposed to Yates' 1000000. I did this mainly due to the prohibitive timeframe associated with running the simulation for more time. In light of this, deviations between the two models in this section should be expected, especially in terms of accuracy of values in the fringes of the ζ range. This is because the simulation rarely allows for values of $|\zeta| > 1.2$, and thus it is much harder to sample those points at high accuracy. This is a problem that I observed in my results, especially with respect to the diffusion coefficient. The parameters that I used in the long term simulation were the same as Yates' and as in previous sections, with $\Delta t = 1$. The full set of values included a ζ array and two 30 by 100000 matrices X and U, corresponding to the position and velocity.

I created the script *sortNsampleRep.m* to sort the simulation data set and sample it at discrete increments of ζ, using these values to run the model a set number of times and thus compute the drift and diffusion terms using equation-free techniques. I managed to do this by first discretizing the ζ interval, using an increment value of 0.005 into equally sized domains. This was decided on based on a tradeoff between computability and accuracy, since smaller values made the scripts too slow. Then, I used the *histc* function to define and construct an array that contained the index of the domain in which each of the ζ values belonged, where the domains were numbered in increasing order. Using this, I then chose each domain individually and, by using the $find$ function on the previous array, created an array of all ζ values it included and sampled it. Then, I proceeded to run the simulation for each of the sampled points for a set number of times. To do this, I created the function *eqnfree.m* which took as inputs the ζ, X and U values and ran the simulation from there. By taking averages in the end, I was able to obtain approximations to the long-term drift and diffusion. These are shown below alongside the corresponding solutions in Yates' paper.

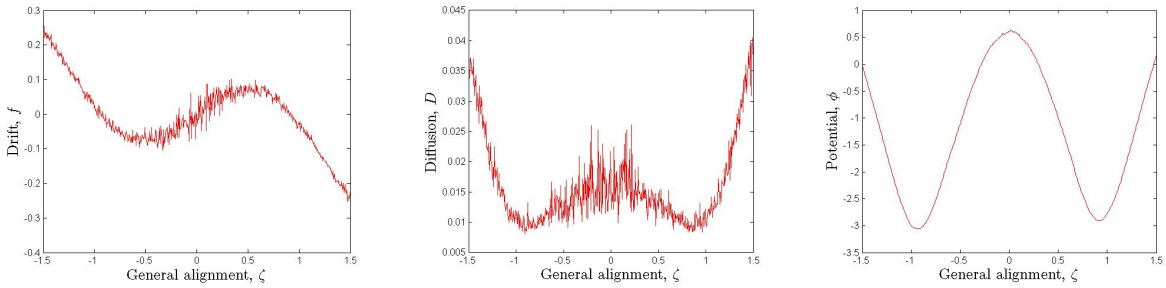
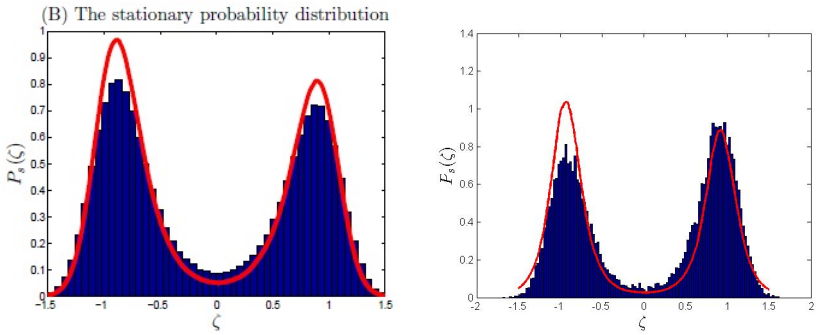


Figure 13: Above can be seen the plots derived for the long-term equation-free analysis using the script *sortNsampleRep.m* and the function *eqnfree.m*.



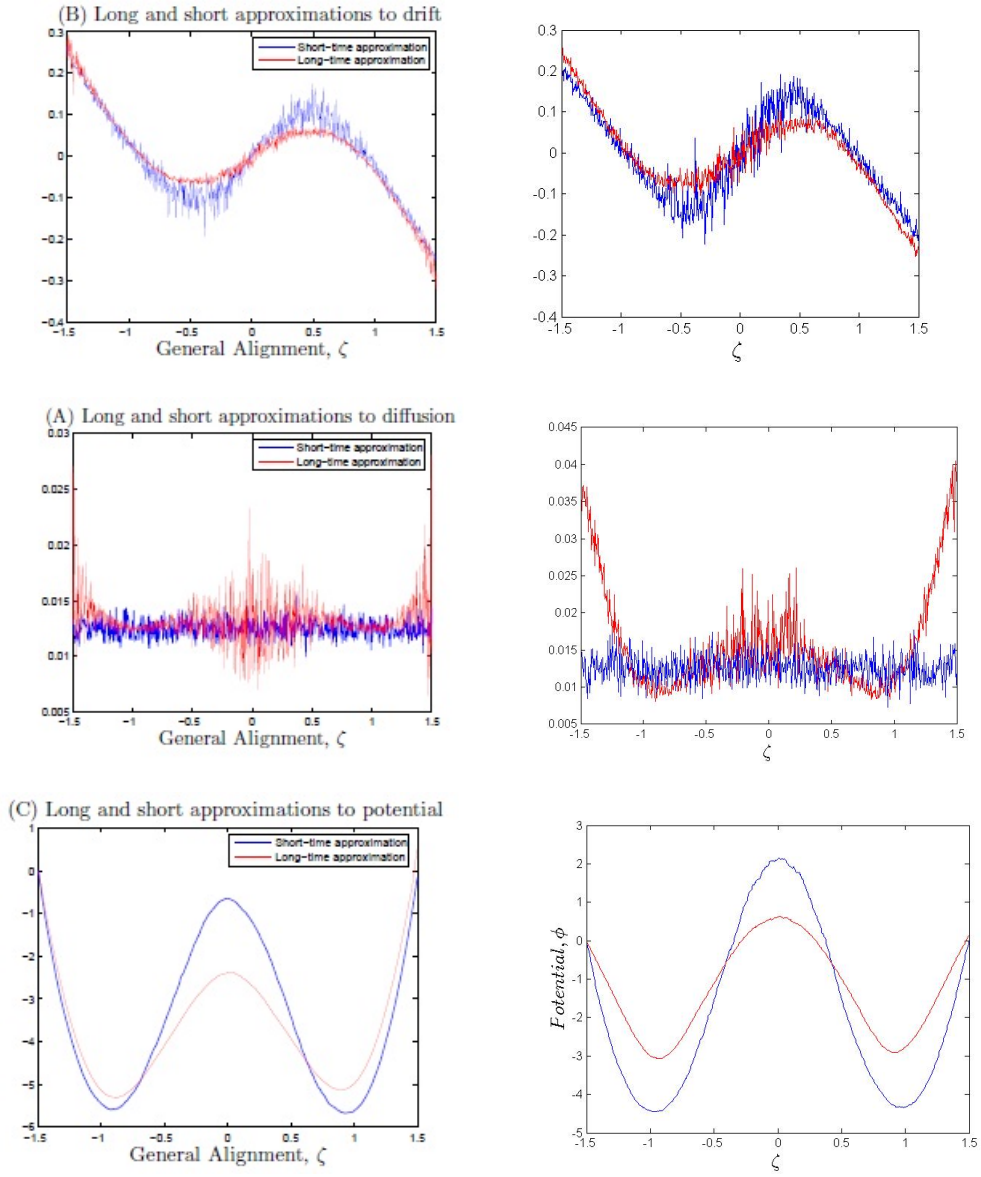


Figure 14: In this figure are plotted the contrasts between the long-term (red) and short-term (blue) equation-free methods. The graphs found in the paper are shown on the left while those that I constructed are on the right. The final two figures show the calculated potentials for the assumed underlying SDEs of the model for each method of analysis. Histogram plotted using [3].

In the data observed above, the most striking difference can be seen for the diffusion, which diverges rapidly for $\zeta > |1.2|$. I attributed this to the fact that I did not let the initial simulation run for enough time, and therefore the sampling at the areas outside of the two basins of potential did not provide an accurate enough indication of the actual diffusion. However, another source of this error could be that the timestep for the approximation to the diffusion was too high. Indeed, I encountered problems when bringing the timestep below 10^{-3} as the values for the diffusion, although substantially more uniform, began to increase in value by an order of magnitude for each power of 10 that I refined my timestep. I tried cutting up the timestep to equally spaced sub-timesteps of vaying orders (as was done with the *calibration* variable previously) but this did not help me refine my results. Therefore, this could have also contributed in the stark deviation from Yates' results.

4 Conclusions

In using the described simulation techniques, I was able to recreate most of the data that Yates provided and to a level of accuracy that retained the qualitative behaviors of each model. The biggest problems in the modelling that I encountered were two and involved both the way that I approached the level of accuracy and also how I used numerics in computing the final solutions. These, alongside comments on how I attempted to deal with them, are briefly described below.

To begin with, I had an issue with the potentials that were derived because, albeit smooth, their values at the middle peak were much higher than anticipated in a way that would seriously affect the switching times of the models. Indeed, when using Yates' derived analytical approximations to find the initial switching times for my potentials, I got values that were orders of magnitude above what I had expected. In order to test out whether the potential that I was using indeed had larger switching times than Yates' (and therefore was a different model), I performed a simulation using *czirok.m* similar to the one coded for in *Switch.java*. I found that the simulated switching times were of the same order of magnitude as those discovered by Yates and that the issue lay in the way that I had computed the potential. In looking for my source of error, I assumed that the problem was created in my approximate values for $\delta\zeta$, so I decided to calculate the potential for the SDE modelled in Section 2 by varying the values of δx to see what a simpler model would behave like under such conditions. I discovered that this also gave me a higher potential around the values for zero. Therefore, I attribute this error to the numeric intergration that I performed in finding the potentials, since this seems to have overestimated the value for the integral.

Finally, my simulation of the long-term equation-free analysis suffered from the serious drawback that it did not have enough points in the edges, and so sampling over repeated times had little effect in approximating the average changes in ζ. This problem can be resolved by sampling over more points, as was done in [4]. Furthermore, as noted in the previous section, I encountered a problem when minimizing the time-step of the simulation, which could also have resulted in the inaccuracies observed. This is particularly plausible because in the areas where the diffusion diverges, we have a large value for $|f|$, which influences greatly the behavior of diffusion as described in Section 2.9 of [4].

References

[1] Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

[2] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[3] Arturo Serrano. Normalized histogram. `http://www.mathworks.com/matlabcentral/fileexchange/22802-normalized-histogram`, May 2013.

[4] Christian A Yates. *On the dynamics and evolution of self-propelled particle models*. PhD thesis, University of Oxford, 2007.