Methods for Solving Fokker Planck

William Zheng

May 2019

1 Abstract

Stochastic differential equations (SDE)s are powerful and have numerous applications. SDEs like Black Scholes dictate how to price options while SDEs like Fokker-Planck are used from finance, to accelerator physics, to even neuroscience. In this project, we study Finite Difference Schemes to solve Fokker-Planck for (2+1) variables. We were able to implement and compare 5 stable finite difference schemes, both explicit and implicit, in terms of convergence of error in time and space on a test problem of a dampened harmonic oscillator. We determined the best algorithm in terms of accuracy and convergence and suggest its use as well as attempt to understand why it performed so well.

2 Derivation of Fokker-Planck

The Fokker-Planck Equation is the equation that governs the time evolution of the probability density of Stochastic Processes. It is a second order differential equation and is exact when the noise acting on the Brownian Particle is Gaussian White Noise [1]. It is also known as the Kolmogorov forward equation and can be derived from the master equation through Kramers–Moyal expansion[2].

Here we will derive the Fokker-Planck Equations adapted from [1]. First we need to derive the equations of motion for the probability density $\varrho(x,t,v)$ for finding the Brownian Particle in the interval $(x, x+dx)$ and $(v, v+dv)$ at time t over the realization for some random force $\xi(t)$. Then we average over this random force in order to obtain an equation for:

$$P(x,v,t) = \langle \varrho(x,v,t) \rangle \tag{1}$$

We will operate in the space of the coordinates $\mathbf{x} = (x,v)$ and say that the Brownian Particle is located in the infinitesimal area (dxdv) with probability $\varrho(x,v,t)$dxdv. Since the Brownian Particle needs to be present, if we integrate over the whole phase space we have:

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dv \varrho(x,v,t) = 1 \tag{2}$$

If we consider a finite volume V_0 and converse the probability of the Brownian Particle (i.e. it cannot be destroyed) then the probability contained in the volume V_0 must be due to the flow of probability through the surface S_0 surrounding V_0.

$$\int\int_{V_0} dxdv \frac{\partial}{\partial t}\varrho(x,v,t) = -\int_{S_0} \varrho(x,v,t)\dot{x} \cdot dS \tag{3}$$

Applying Gauss's Theorem to take the surface integral into a volume integral we have:

$$\int\int_{V_0} dxdv \frac{\partial}{\partial t}\varrho(x,v,t) = -\int\int_{V_0} dxdv \nabla \cdot (\dot{x}\varrho(x,v,t)) \tag{4}$$

Since we have V_0 as fixed and arbitrary we then can set the two integrands equal to each other to get the continuity equation:

$$\frac{\partial}{\partial t}\varrho(x,v,t) = -\nabla \cdot (\dot{x}\varrho(x,v,t)) = -\frac{\partial}{\partial x}(\dot{x}\varrho(x,v,t)) - \frac{\partial}{\partial v}(\dot{v}\varrho(x,v,t)) \tag{5}$$

With this continuity equation we can continue to solve for the probability flow of a Brownian Particle. We begin with the Langevin Equation governing the evolution of the particle. A particle moving through a potential V(x) has the Langevin equations:

$$\frac{dx}{dt} = v \tag{6}$$

$$\frac{dv}{dt} = -\frac{\gamma}{m}v + \frac{1}{m}F(x) + \frac{1}{m}\xi(t) \tag{7}$$

where the force is $F(x) = -V'(x)$. If we substitute these equations into the continuity equation we derived (4), then we get the expression:

$$\frac{\partial}{\partial t}\varrho(x,v,t) = -\frac{\partial}{\partial x}(v\varrho(x,v,t)) + \frac{\gamma}{m}\frac{\partial}{\partial v}(v\varrho(x,v,t)) \tag{8}$$

$$-\frac{1}{m}F(x)\frac{\partial}{\partial v}(\varrho(x,v,t)) - \frac{1}{m}\xi(t)\frac{\partial}{\partial v}\varrho(x,v,t) \tag{9}$$

$$= -L_0\varrho(x,v,t) - L_1(t)\varrho(x,v,t) \tag{10}$$

Where we have two differential operators L_0 and L_1 defined as:

$$L_0 = v\frac{\partial}{\partial x} - \frac{\gamma}{m} - \frac{\gamma}{m}v\frac{\partial}{\partial v} + \frac{1}{m}F(x)\frac{\partial}{\partial v} \tag{11}$$

$$L_1 = \frac{1}{m}\xi(t)\frac{\partial}{\partial v} \tag{12}$$

Since $\xi(t)$ represents a stochastic variable, we need to introduce an observable probability density that represents the average over this stochastic variable, $P(x,v,t) = \langle \xi(t) \rangle_\xi$.

If we let:

$$\varrho(t) = e^{-L_0 t}\sigma(t) \tag{13}$$

Then we have:

$$\varrho(t) = -e^{-L_0 t}L_t(t)e^{-L_0 t}\sigma(x,v,t) = -V(t)\sigma(x,v,t) \tag{14}$$

This has the solution

$$\sigma(t) = \exp - \int_0^t dt' V(t')\sigma(0) \tag{15}$$

Taking the average over the Gaussian Noise $\xi(t)$ or the rather the expected value, $\langle \sigma(t) \rangle_\xi$ is the characteristic function of the random variable $X(t) = i\int_0^t dt_1 V(t_1)$. X must be a Gaussian random variable with $\langle X(t) \rangle_\xi = 0$ and the variance of:

$$\langle X(t)^2 \rangle = \frac{1}{2}\int_0^t dt_1 \int_0^t dt_2 \langle V(t_1)V(t_2) \rangle \tag{16}$$

Since the characteristic function of a Gaussian r.v. $X(t)$ is known: $\exp(iX(t)) = \exp(i\mu_X - \langle X(t)^2 \rangle /2)$ we have:

$$\langle \sigma(t) \rangle = \exp\left(\frac{1}{2}\int_0^t dt_1 \int_0^t dt_2 \langle V(t_1)V(t_2) \rangle\right)\sigma(0) \tag{17}$$

We can reduce this, since it is just a special case of a cumulant expansion, to:

$$\frac{1}{2}\int_0^t dt_1 \int_0^t dt_2 \langle V(t_1)V(t_2) \rangle = \frac{g}{m^2}\int_0^t dt_1 e^{L_0 t_1}\frac{\partial^2}{\partial v^2}e^{-L_0 t_1} \tag{18}$$

Therefore we have:

$$\frac{\partial}{\partial t}\langle \sigma(x,v,t) \rangle_\xi = \frac{g}{2m^2}e^{L_0 t_1}\frac{\partial^2}{\partial v^2}\langle \varrho(x,v,t) \rangle_\xi \tag{19}$$

Therefore for $\langle \varrho(x,v,t) \rangle_\xi$ we have:

$$\frac{\partial}{\partial t}\langle \varrho(x,v,t) \rangle_\xi = -L_0 \langle \varrho(x,v,t) \rangle_\xi + \frac{g}{2m^2}\frac{\partial^2}{\partial v^2}\langle \varrho(x,v,t) \rangle_\xi \tag{20}$$

so for the probability distribution evolution we have the following equation:

$$\frac{\partial}{\partial t}P(x,v,t) = -v\frac{\partial}{\partial x}P(x,v,t) - \frac{\partial}{\partial v}\left[\left(\frac{\gamma}{m}v - \frac{1}{m}F(x)\right)P(x,v,t)\right] + \frac{g}{2m^2}\frac{\partial^2}{\partial v^2}P(x,v,t) \tag{21}$$

A very cool idea that originates from this physics-like derivation was the idea of a continuity equation and also a definition for probability current! From the Fokker-Planck Equation as a continuity equation:

$$\frac{\partial}{\partial t}P(x,v,t) = -\nabla \cdot j \tag{22}$$

where $\nabla = e_x\frac{\partial}{\partial x} + e_v\frac{\partial}{\partial v}$ and the probability current is then:

$$j = e_x vP - e_v\left[\left(\frac{\gamma}{m}v - \frac{1}{m}F(x)\right)P - \frac{g}{2m^2}\frac{\partial}{\partial v}P\right] \tag{23}$$

3 Applications and Numerical Methods

There are various techniques used to solve the Fokker-Planck Equation which include Monte Carlo Methods [3] [4], Finite Element Method [5], Robust Finite Difference Methods which include the Fully Implicit Chang Cooper Method [6], operator splitting [7], and the Central Finite Differences and Alternating Directions Implicit Method [8]. Methods that have been recently introduced include Proximal Recursion [9].

Applications for solving the Fokker-Planck Equations include modelling single particle accelerator dynamics [10][11], pricing an option, and to characterize the spiking statistics of neurons receiving noisy synaptic input [12] with other applications in neuroscience.

For this project, we would like to numerically solve the Fokker-Planck Equation in (2+1) variables with our primary inspiration coming from stochastic beam dynamics within accelerators. An important problem within this field is particle motion under the influence of various noise sources. [13]. A stochastic differential equation governs this motion and its solutions are Markovian Diffusion Processes which can be described by the Fokker Planck Equation[10].

We would like to implement the method from [10], which is a fully implicit operator splitting method, and also implement an operator splitting method which utilizes Backward Euler and Crank-Nicholson Schemes as part of evaluating each operator [7]. We compare each method's error at different time steps on a test problem of a damped harmonic oscillator with strong diffusion and perform a convergence study. We perform an additional von Neumann Analysis on the fully implicit operator splitting method.

4 Description of the Numerical Methods

Following the general equations of motion studied in [10], we study equations of the form:

$$\frac{d}{ds}x_1 = x_2 \tag{24}$$

$$\frac{d}{ds}x_2 = -a_1(x_1) - a_2(x_1,x_2) + a_3(x_1)\eta_1 + a_4\eta_2 \tag{25}$$

We have η_1, η_2 as white noise Gaussian processes and $a_1(x_1)$ can be an arbitrary nonlinear potential (field), $a_2(x_1,x_2)$ can include van der Pol-like damping terms, $a_3(x_1)\eta_1$ describes random parameters and $a_4\eta_2$ represents an additive noise term.

The corresponding (Ito) Fokker-Planck Equation for the probability density reads:

$$\begin{aligned}\frac{\partial}{\partial s}p(x_1,x_2,s) = &-\frac{\partial}{\partial x_1}[x_2 \cdot p(x_1,x_2,s)] + \frac{\partial}{\partial x_2}[(a_1(x_1) + a_2(x_1,x_2)) \cdot p(x_1,x_2,s)] \\ &+ \frac{1}{2}\frac{\partial^2}{\partial x_2^2}[(a_3^2(x_1) + a_4^2) \cdot p(x_1,x_2,s)].\end{aligned}$$

This now can be evaluated in a few different ways. In [10] the previous equation could be written in the form of two fluxes, one in x_1 and one in x_2 such that:

$$\frac{\partial p}{\partial s} = \frac{\partial A}{\partial x_1} + \frac{\partial B}{\partial x_2}$$

This form implies that we should solve this with an operator splitting technique like the following. First we need to implicitly evaluate the x_2 derivative which takes on this form:

$$\frac{p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = \frac{p_{i,j}^{n}}{\Delta s} + \frac{F_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2} \tag{26}$$

Where we have $F_{i,j+\frac{1}{2}}$ defined as the following:

$$F_{i,j+\frac{1}{2}} = D\frac{p_{i,j+1} - p_{i,j}}{\Delta x_2} + [a_1(x_1) + a_2(x_1, x_2 + \Delta x_2)]\frac{p_{i,j+1} + p_{i,j}}{2} \tag{27}$$

Where we have $D = \frac{a_3^2(x_1)+a_4^4}{2}$ however, we view this term with caution as it may actually be incorrect. In order to then implicitly evaluate the x_1 derivative we perform the following:

$$\frac{p_{i,j}^{n+1} - p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = -x_2 \cdot \frac{p_{i+1,j}^{n+1} - p_{i-1,j}^{n+1}}{2\Delta x_1} \tag{28}$$

This is therefore in total the method described in [10]. If we use the same equations of motion, phrased a bit differently, we can split this into three operators [7]:

$$\frac{d}{ds}x_1 = x_2 \tag{29}$$

$$\frac{d}{ds}x_2 = -g(x_1, x_2, t) + f(t) \tag{30}$$

Where we have f(t) as some random Gaussian Noise where we have:

$$\langle f(t) \rangle = 0 \tag{31}$$

$$\langle f(t_1)f(t_2) \rangle = \frac{W_0}{2}\delta(t_1 - t_2) \tag{32}$$

Then the Fokker-Planck Equation for this system is given by [14].

$$\frac{W_0}{4}\frac{\partial^2 p}{\partial y^2} - \frac{\partial}{\partial y_2}g(y_1, y_2)p = \frac{\partial p}{\partial t} \tag{33}$$

Therefore if we split this operator up into three parts we get:

$$\frac{\partial p}{\partial t} = \mathcal{L}_1 p + \mathcal{L}_2 p + \mathcal{L}_3 p \tag{34}$$

$$\text{with} \begin{cases} p^{n+\frac{1}{3}} = \mathcal{U}_1(p^n) \\ p^{n+\frac{2}{3}} = \mathcal{U}_2(p^{n+\frac{1}{3}}) \\ p^{n+1} = \mathcal{U}_3(p^{n+\frac{2}{3}}) \end{cases} \tag{35}$$

Then in these cases we can replace $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ with different finite difference schemes that will represent the respective operators $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$.

Now in this project, we will attempt to implement the implicit 2-operator splitting finite difference technique from [10] and 3-operator splitting testing the accuracy of both explicit and implicit methods.

5 Implementation of the Numerical Methods

5.1 Implementation of fully implicit 2-operator splitting method

In order to implement the fully implicit 2-operator splitting method from [10] we try to rewrite the expression for the derivative of x_2 in terms of more manageable coefficients to easily see the tri-diagonal form:

This means we have a tri-diagonal expression for the first half step up into p. Since we define F as the following:

$$F_{i,j+\frac{1}{2}} = D\frac{p_{i,j+1} - p_{i,j}}{\Delta x_2} + [a_1(x_1) + a_2(x_1, x_2 + \Delta x_2)]\frac{p_{i,j+1} + p_{i,j}}{2} \tag{36}$$

At this point we also realize that C and D will be dependent on x_1 and x_2 therefore we have $D_{i,j}, C_{i,j}, E_{i,j}$. This will mean that our variables at (i,j) will correspond to evaluating $F_{i,j+\frac{1}{2}}$. We rewrite F as the following:

$$F_{i,j+\frac{1}{2}} = D_{i,j}\frac{p_{i,j+1} - p_{i,j}}{\Delta x_2} + C_{i,j}\frac{p_{i,j+1} - p_{i,j}}{2} \tag{37}$$

Where we define the following in order to make our calculations easier:

$$C_{i,j} = \frac{[a_1(x_1) + a_2(x_1, x_2 + \Delta x_2)]}{2} \tag{38}$$

$$E_{i,j} = \frac{D_{i,j}}{\Delta x_2} \tag{39}$$

Rewriting $F_{i,j+\frac{1}{2}}$ now in terms of our new coefficients we have:

$$F_{i,j+\frac{1}{2}} = E_{i,j} \cdot (p_{i,j+1} - p_{i,j}) + C_{i,j} \cdot (p_{i,j+1} + p_{i,j}) \tag{40}$$

$$F_{i,j+\frac{1}{2}} = (C_{i,j} + E_{i,j}) \cdot (p_{i,j+1}) - (E_{i,j} - C_{i,j}) \cdot (p_{i,j}) \tag{41}$$

$$\tag{42}$$

Therefore the terms for $F_{i,j-\frac{1}{2}}$.

$$F_{i,j-\frac{1}{2}} = (C_{i,j-1} + E_{i,j-1}) \cdot (p_{i,j}) - (E_{i,j-1} - C_{i,j-1}) \cdot (p_{i,j-1}) \tag{43}$$

Back to our expression for the x_2 derivative we need to calculate $F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}$.

$$F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}} = (C_{i,j} + E_{i,j}) \cdot p_{i,j+1} - [(E_{i,j} - C_{i,j}) \cdot p_{i,j} + (C_{i,j-1} + E_{i,j-1}) \cdot p_{i,j}] + (E_{i,j-1} - C_{i,j-1}) \cdot p_{i,j-1}$$

Therefore combining the two for the implicit derivative in the x_2 we get:

$$\frac{p_{i,j}^{n+\frac{1}{2}} - p_{i,j}^{n}}{\Delta s} = \frac{F_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2}$$

$$\frac{p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = \frac{p_{i,j}^{n}}{\Delta s} + \frac{F_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2}$$

$$\frac{p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = \frac{p_{i,j}^{n}}{\Delta s} + \frac{(C_{i,j} + E_{i,j}) \cdot p_{i,j+1}^{n+\frac{1}{2}} - [(E_{i,j} - C_{i,j}) \cdot p_{i,j}^{n+\frac{1}{2}} + (C_{i,j-1} + E_{i,j-1}) \cdot p_{i,j}^{n+\frac{1}{2}}] + (E_{i,j-1} - C_{i,j-1}) \cdot p_{i,j-1}^{n+\frac{1}{2}}}{\Delta x_2}$$

$$p_{i,j}^{n+\frac{1}{2}} = p_{i,j}^{n} + \Delta s\left(\frac{(C_{i,j} + E_{i,j}) \cdot p_{i,j+1}^{n+\frac{1}{2}} - [(E_{i,j} - C_{i,j}) + (C_{i,j-1} + E_{i,j-1})] \cdot p_{i,j}^{n+\frac{1}{2}} + (E_{i,j-1} - C_{i,j-1}) \cdot p_{i,j-1}^{n+\frac{1}{2}}}{\Delta x_2}\right)$$

Therefore in Tri-diagonal Form we get:

$$-p_{i,j}^{n} = -p_{i,j}^{n+\frac{1}{2}} + \frac{\Delta s \cdot (C_{i,j} + E_{i,j}) \cdot p_{i,j+1}^{n+\frac{1}{2}}}{\Delta x_2} + \frac{-\Delta s \cdot [(E_{i,j} - C_{i,j}) + (C_{i,j-1} + E_{i,j-1})] \cdot p_{i,j}^{n+\frac{1}{2}}}{\Delta x_2} + \frac{\Delta s \cdot (E_{i,j-1} - C_{i,j-1}) \cdot p_{i,j-1}^{n+\frac{1}{2}}}{\Delta x_2}$$

This is under the assumption that $E_{i,j} = E_{i,j-1}$. We will also set $C_{i,j} = C_{i,j-1}$ This will prevent the method from "losing" probability and suffer constantly increasing/decreasing values. These variables are then used in the code as part of this project.

5.2 Implementation of the 3-operator splitting method

To implement the 3-operator splitting method, we need to choose methods that can replace $\mathcal{U}_1$,$\mathcal{U}_2$,$\mathcal{U}_3$ with different finite difference schemes that will represent the respective operators $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$.

For implementation we use:

$$\begin{cases} \mathcal{L}_1 = \text{Implicit Euler, Explicit Euler} \\ \mathcal{L}_2 = \text{Implicit Euler} \\ \mathcal{L}_3 = \text{Backward Euler, Crank-Nicolson} \end{cases}$$

These methods were also implemented assuming zero boundary conditions.

We only use the explicit method for the $\mathcal{L}_2$ because we were not able to get it to reproduce the Implicit Euler method on the $\mathcal{L}_2$. We also tried an explicit method for the $\mathcal{L}_3$ step, however, we were also unable to reproduce the accuarcy that Backward Euler and Crank-Nicolson achieved. We attribute this possible to boundary conditions we were unable to phrase properly.

6 Testing the Numerical Methods

6.1 Test Problem: Dampened Harmonic Oscillator

The first example that we will use is the damped harmonic oscillator with strong diffusion where $a_1(x_1) = Kx_1$, $a_2(x_1, x_2) = \gamma x_2$, $a_3(x_1) = 0$, and $a_4 = \sigma$. Where we set $K = 1$, $\gamma = 2.1$, $\sigma = 0.8$, a 81 by 81 grid, $\Delta x_1 = \Delta x_2 = 0.1, \Delta s = \frac{\pi}{1000}$ where we use the exact solution for the Harmonic Oscillator at $s = 0.95$. A note, we set $D = 0.8$, with reference to [10] which is the same as $W_0 = 3.2$ [7] and simulate out 800 time steps. (We note that s and t both represent the same quantity time).

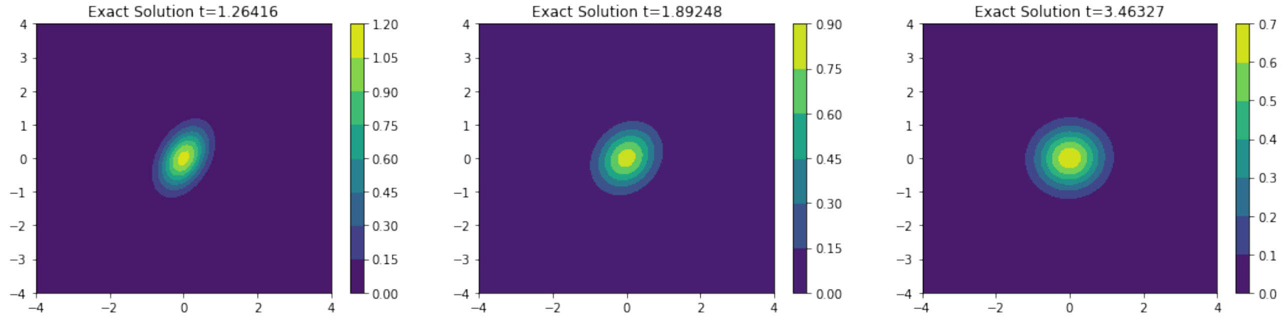

Figure 1: In this figure we show the PDF over the phase space (x_1,x_2) for the dampened harmonic oscillator.

We denote the methods as the following. The 2-operator implicit method is 2OI, the Implicit ($\mathcal{L}_1$), Implicit ($\mathcal{L}_2$), Crank-Nicolson ($\mathcal{L}_3$) Scheme is IICN, the Implicit ($\mathcal{L}_1$), Implicit ($\mathcal{L}_2$), Backwards Euler ($\mathcal{L}_3$) Method is IIBE, the Explicit ($\mathcal{L}_1$), Implicit ($\mathcal{L}_2$), Crank-Nicolson ($\mathcal{L}_3$) Method is denoted EICN, and the Explicit ($\mathcal{L}_1$), Implicit ($\mathcal{L}_2$), Backwards Euler ($\mathcal{L}_3$) Method is EIBE. We do the follow comparisons of the error of these methods after 800 time steps where we define error as $||e|| = \sqrt{\frac{1}{N}(\sum(\rho_{i,j}^{num} - \rho_{i,j}^{exact}))}$ where N represents the total amount of data points present.

Numerical Method	**Error at 800 time steps**
IICN	0.000450273
IIBE	0.000468833
EICN	0.000256599
EIBE	0.000333529
2OI	0.007862812

Figure 2: In this figure we show the error of the method after simulating 800 time steps, we find that the best performing algorithm is EICN and the worst performing algorithm is 2OI by an order of magnitude with respect to the rest of the algorithms.

6.2 Error Convergence: Space

We now perform error convergence measurements all of our algorithms.

Figure 3: In this figure we show the convergence of error for the 2OI method is not even 1. It seems like this algorithm does quite poorly.

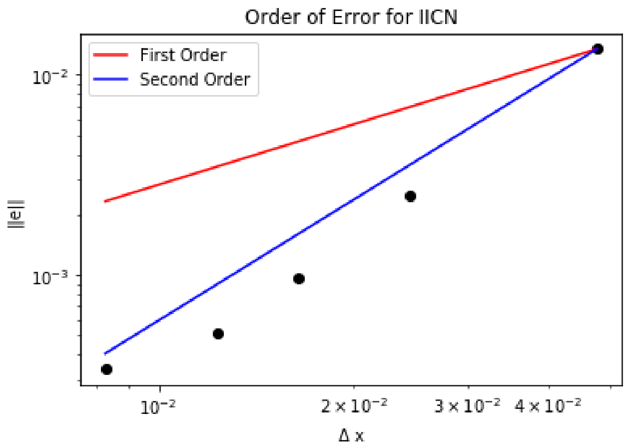

Figure 4: In this figure we show the convergence of error for the IICN method is around 2 and appears to show small signs that the error will stop improving as quickly as we decrease Δx.

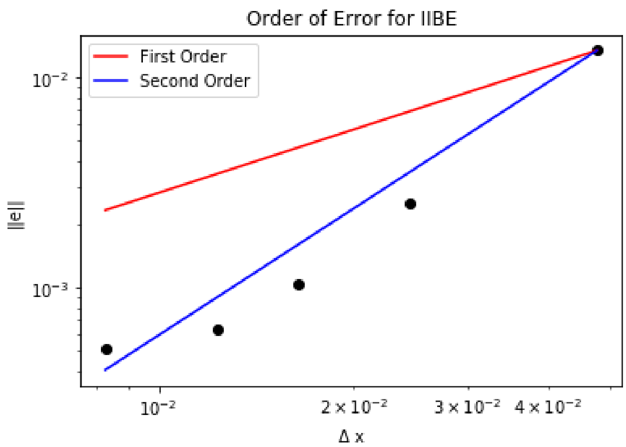

Figure 5: In this figure we show the convergence of error for the IIBE method is around 2 and appears to show large signs that the error will stop improving quickly as we decrease Δx.

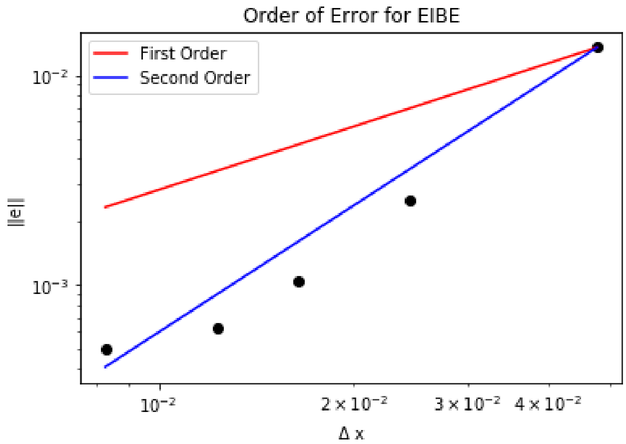

Figure 6: In this figure we show the convergence of error for the EIBE method is around 2 and appears to show large signs that the error will stop improving quickly as we decrease Δx.

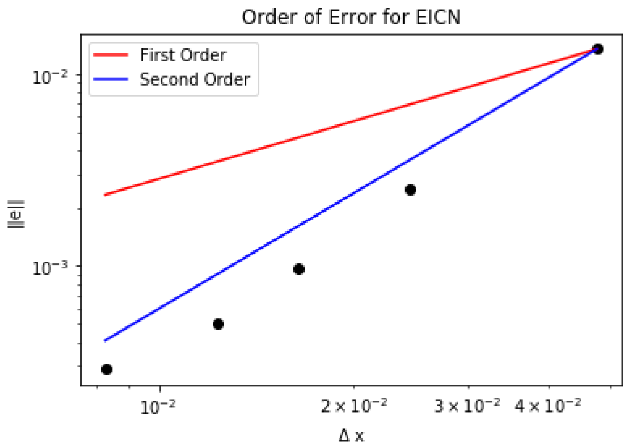

Figure 7: In this figure we show the convergence of error for the EICN method is around 2 and appears to show small signs that the error will stop improving quickly as we decrease Δx.

6.3 Error Convergence: Time

We also were able to test convergence with time and for the 2OI method, decreasing the step size seemed to barely improve the error where it wasn't converging with any appreciable order. However, we were able to achieve first order convergence in error with all the three operator methods. This is what we expect since our methods are limited by steps that are first order in time.

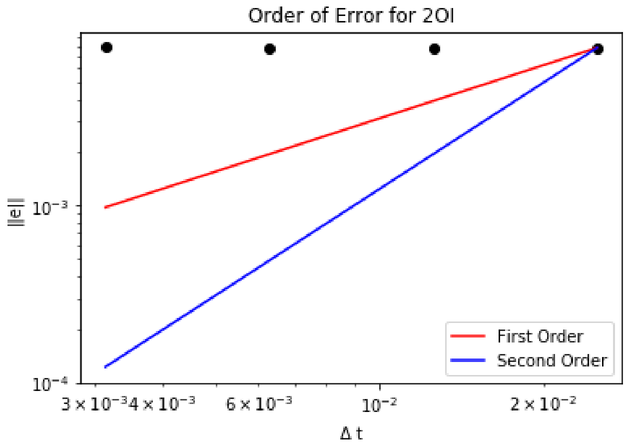

Figure 8: In this figure we show the convergence of error for the 2OI method is not even 1. It seems like this algorithm does quite poorly.

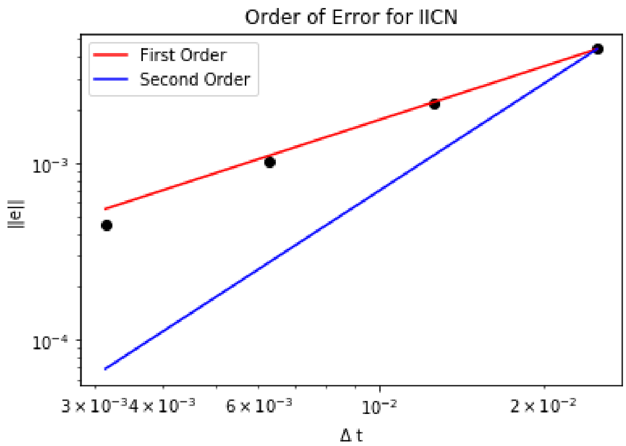

Figure 9: In this figure we show the convergence of error for the IICN method with respect to time is order 1.

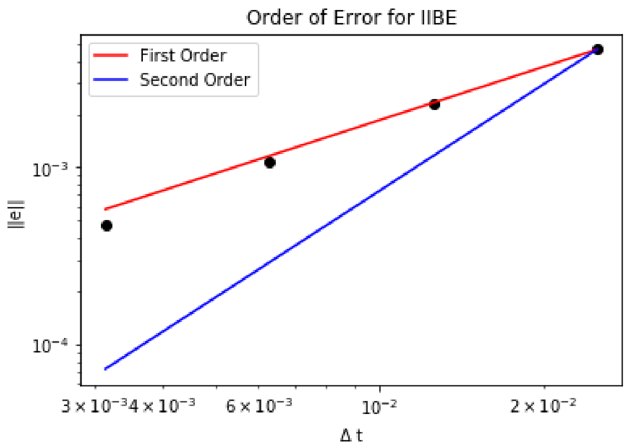

Figure 10: In this figure we show the convergence of error for the IIBE method with respect to time is order 1.

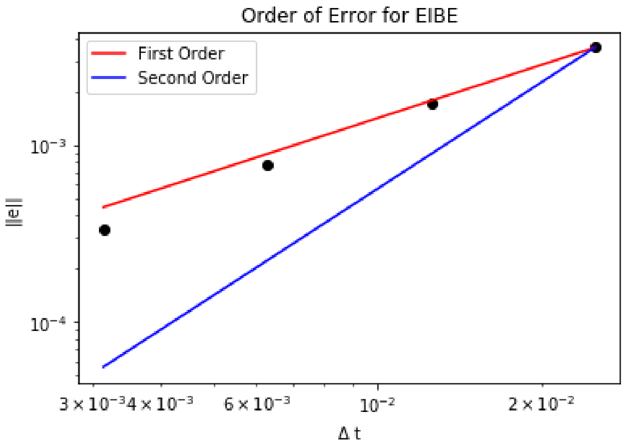

Figure 11: In this figure we show the convergence of error for the EIBE method with respect to time is order 1.

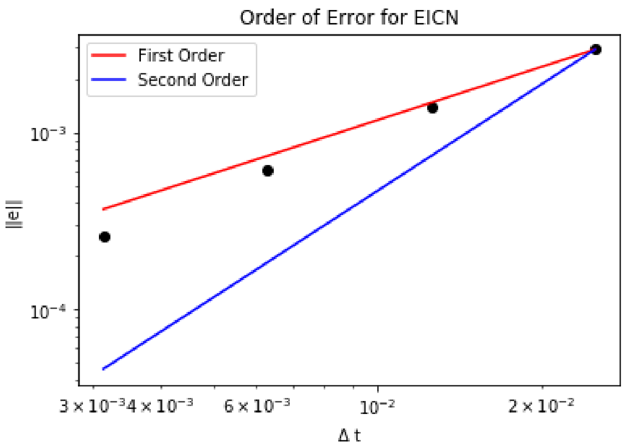

Figure 12: In this figure we show the convergence of error for the EICN method with respect to time is order 1.

7 Stability of the Numerical Methods

7.1 Stability of fully implicit 2-operator splitting method

Although it was mentioned in [6] that some implicit methods for solving the Fokker-Planck Equation were unstable, with guidance from [11] we were able to show that the 2OI (2-Operator Implicit Method) has unconditional stability in both steps by Von-Neumann Analysis.

For the first step, x_2 derivative, we have the following:

$$\frac{p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = \frac{p_{i,j}^{n}}{\Delta s} + \frac{F_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - F_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2}$$

$$F_{i,j+\frac{1}{2}} = D\frac{p_{i,j+1} - p_{i,j}}{\Delta x_2} + [a_1(x_1) + a_2(x_1, x_2 + \Delta x_2)]\frac{p_{i,j+1} + p_{i,j}}{2}$$

We will consider $p_{i,j}^n = g(\xi)^n e^{i\xi j\Delta x}$ and we will approximate $a \approx [a_1(x_1) + a_2(x_1, x_2 + \Delta x_2)] \approx [a_1(x_1) + a_2(x_1, x_2 - \Delta x_2)]$

$$F_{i,j+\frac{1}{2}} = D\frac{p_{i,j+1} - p_{i,j}}{\Delta x_2} + a\frac{p_{i,j+1} + p_{i,j}}{2}$$

$$\frac{p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = \frac{p_{i,j}^{n}}{\Delta s} + \frac{D\frac{p_{i,j+1}^{n+\frac{1}{2}} - p_{i,j}^{n+\frac{1}{2}}}{\Delta x_2} + a\frac{p_{i,j+1}^{n+\frac{1}{2}} + p_{i,j}^{n+\frac{1}{2}}}{2} - [D\frac{p_{i,j}^{n+\frac{1}{2}} - p_{i,j-1}^{n+\frac{1}{2}}}{\Delta x_2} + a\frac{p_{i,j}^{n+\frac{1}{2}} + p_{i,j-1}^{n+\frac{1}{2}}}{2}]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{D\frac{e^{\xi i\Delta x}-1}{\Delta x_2} + a\frac{e^{\xi i\Delta x}+1}{2} - [D\frac{1-e^{-\xi i\Delta x}}{\Delta x_2} + a\frac{1+e^{-\xi i\Delta x}}{2}]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{D[\frac{e^{\xi i\Delta x}-1}{\Delta x_2} - \frac{1-e^{-\xi i\Delta x}}{\Delta x_2}] + a[\frac{e^{\xi i\Delta x}+1}{2} - \frac{1+e^{-\xi i\Delta x}}{2}]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{D[\frac{e^{\xi i\Delta x}-2+e^{-\xi i\Delta x}}{\Delta x_2}] + a[\frac{e^{\xi i\Delta x}-e^{-\xi i\Delta x}}{2}]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{D[\frac{-2+2\cos(\xi\Delta x)}{\Delta x_2}] + a[\frac{i\sin(\xi\Delta x)}{1}]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{D[\frac{-2+2-4\sin^2(\frac{\xi\Delta x}{2})}{\Delta x_2}] + a[i\sin(\xi\Delta x)]}{\Delta x_2}$$

$$\frac{1}{\Delta s} = \frac{g(\xi)^{-\frac{1}{2}}}{\Delta s} + \frac{-4D}{\Delta x_2^2}\sin^2(\frac{\xi\Delta x}{2}) + \frac{1}{\Delta x_2}a[i\sin(\xi\Delta x)]$$

$$1 = g(\xi)^{-\frac{1}{2}} + \frac{-4D\Delta s}{\Delta x_2^2}\sin^2(\frac{\xi\Delta x}{2}) + \frac{\Delta s}{\Delta x_2}a[i\sin(\xi\Delta x)]$$

$$1 + \frac{4D\Delta s}{\Delta x_2^2}\sin^2(\frac{\xi\Delta x}{2}) - \frac{\Delta s}{\Delta x_2}a[i\sin(\xi\Delta x)] = g(\xi)^{-\frac{1}{2}}$$

Therefore if we consider a half step as a whole step we have:

$$g(\xi) = \frac{1}{1 + \frac{4D\Delta s}{\Delta x_2^2}\sin^2(\frac{\xi\Delta x}{2}) - \frac{\Delta s}{\Delta x_2}a[i\sin(\xi\Delta x)]}$$

For the second step, x_1 derivative, we have:
Now we will consider $p_{i,j}^n = g(\xi)^n e^{i\xi j\Delta x}$

$$\frac{p_{i,j}^{n+1} - p_{i,j}^{n+\frac{1}{2}}}{\Delta s} = -x_2 \cdot \frac{p_{i+1,j}^{n+1} - p_{i-1,j}^{n+1}}{2\Delta x_1}$$

$$\frac{g(\xi)^{n+1}e^{i\xi j\Delta x} - g(\xi)^{n+\frac{1}{2}}e^{i\xi j\Delta x}}{\Delta s} = -x_2 \cdot \frac{g(\xi)^{n+1}e^{(i)\xi(j+1)\Delta x} - g(\xi)^{n+1}e^{(i)\xi(j-1)\Delta x}}{2\Delta x}$$

$$\frac{1 - g(\xi)^{-\frac{1}{2}}}{\Delta s} = -x_2 \cdot \frac{e^{\xi i\Delta x} - e^{-\xi i\Delta x}}{2\Delta x}$$

Using $e^{\xi j\Delta x} = \cos(\xi j\Delta x) + i\sin(\xi j\Delta x)$ we get:

$$\frac{1 - g(\xi)^{-\frac{1}{2}}}{\Delta s} = -x_2 \cdot \frac{\cos(\xi\Delta x) + i\sin(\xi\Delta x) - (\cos(\xi\Delta x) - i\sin(\xi\Delta x))}{2\Delta x}$$

$$\frac{1 - g(\xi)^{-\frac{1}{2}}}{\Delta s} = -x_2 \cdot \frac{2i\sin(\xi\Delta x)}{2\Delta x}$$

$$1 - g(\xi)^{-\frac{1}{2}} = -x_2 \cdot \frac{\Delta s}{\Delta x}i\sin(\xi\Delta x)$$

$$g(\xi)^{-\frac{1}{2}} = 1 + x_2 \cdot \frac{\Delta s}{\Delta x}i\sin(\xi\Delta x)$$

If we consider the fraction half step as a full half step we will have:

$$g(\xi) = \frac{1}{1 + x_2 \cdot \frac{\Delta s}{\Delta x}i\sin(\xi\Delta x)}$$

Therefore as claimed by [11] our expressions for g show that:

$$|g| \leq 1 \tag{44}$$

The claim is also that the scheme is second order accurate in Δx, Δv, and Δt [11].

7.2 Stability of Forwards/Explicit Euler Method

We were able to show in class [15] that the Forward Euler method is Lax-Richtmeyer stable and has strong stability which by the Lax Equivalence Theorem shows that the linear method is convergent since it is Lax-Richtmyer Stable. This method is first order in time and second order in space since we use second order centered difference in space for the discretization.

7.3 Stability of Backwards/Implicit Euler Method

Backwards Euler is known as an L-Stable method where the absolute Stability Regions for backwards Euler consists of the entire left half plane. Therefore as a method it is very stable. This method is first order in time and second order in space.

7.4 Stability of Crank Nicolson Method

For the Crank-Nicholson Method we know that the trapezoidal method is absolutely stable so we can choose any Δt [16]. We also know that these method is second order accurate in both time and space.

8 Discussion

8.1 Analysis

We find that after comparing the 2OI, IICN, IIBE, EICN, EIBE algorithms 2OI seems to be pretty poor in our implementation in comparison to the rest in terms of accuracy level, being an order of magnitude off, and in terms of convergence in space, where 2OI couldn't achieve even first order convergence while the other methods easily achieved second order convergence. This could be because we didn't implement proper boundary conditions as they were not specified in [10] since we are supposed to achieve second order convergence in space and a high level of accuracy. However, we do show that all the methods are stable and that we are able to achieve second order convergence and very good accuracy with respect to results achieved in [7] for the IICN, IIBE, EICN, EIBE algorithms. If we were to suggest using some of these methods, we would suggest EICN, since EICN had the lowest error at 800 time steps and 2nd order convergence in space with little sign of slowing down. IICN also appeared to have 2nd order convergence in space with little sign of slowing down while the other methods seemed to be slowing down as we increased Δx as if they were going to stop second order convergence with respect to space. We remark that since the two algorithms in question used Backwards Euler, maybe that is the reason why. Future work can be done to elucidate this. We also notice that the algorithms that used the explicit method for the $\mathcal{L}_1$ operator achieved some of the best accuracy scores.

8.2 Difficulties and Nuances

We found that [10] and [11] contain a fair share of discrepancies, but for a paper the same age as me, I don't fault them. If we had the full description of the algorithm, maybe we would be able to achieve better convergence. Otherwise, our implementation took very long overall since it was so hard to decipher. [7] was much more straight forward and helped me realize that a term on the Dampened Harmonic Oscillator Expression was the diffusive term which corrected a major mistake in the 2OI algorithm. We also noticed that the operators $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ could be performed in different orders in order to increase accuracy. The best combination of the orders was $\mathcal{L}_3$, $\mathcal{L}_1$, $\mathcal{L}_2$ which decrease the error at the 800th time step by half in comparison to the normal order!

9 Conclusion

In this project, we were able study Finite Difference Schemes to solve Fokker-Planck for (2+1) variables. We were able to implement and compare 5 stable finite difference schemes, both explicit and implicit, in terms of convergence in space and accuracy on a test problem of a dampened harmonic oscillator. We found that the EICN scheme is the most accurate in solving the test problem and displays the best convergence characteristics. While we weren't able to achieve the best 2OI convergence and accuracy values from [10], we were able to implement it successfully and prove its stability. We were able to achieve first order convergence in time for the majority of our methods except for the 2OI method. Overall, we were able to perform a comparison across these 5 schemes for solving Fokker-Planck, determine the best scheme, and elucidate possible areas of future study.

10 Appendix

The Jupyter Notebooks are arranged as follows:

1. Numerical Solution for Fokker Planck Equations in Accelerators: Description and Code of the 2OI
2. Fokker Planck Implementation: Implementation of the 2OI and Analysis
3. Fokker Planck Three Operators: Code for IICN, IIBE, EIBE, EICN
4. Testing Convergence Three Operators: Analysis for IICN, IIBE, EIBE, EICN

The module FokkerPlanck.py includes the implementation code for IICN, IIBE, EIBE, and EICN and comments!

References

[1] Kap Chapter 7. "Brownian Motion : Fokker-Planck Equation". In: ().

[2] Wikipedia. "Fokker–Planck$_{e}$*quation*". In: ().

[3] J Varje. "Monte Carlo method and High Performance Computing for solving Fokker-Planck equation of minority plasma particles". In: (2015). DOI: 10.1017/S0022377815000203. arXiv: arXiv:1510.06221v1.

[4] K Kikuchi, M Yoshida, T Maekawa, et al. "Metropolis Monte Carlo method as a numerical technique to solve the Fokker-Planck equation". In: 185.3 (1991), pp. 335–338.

[5] Pankaj Kumar and S. Narayanan. "Solution of Fokker-Planck equation by finite element and finite difference methods for nonlinear systems". In: *Sadhana* 31.4 (Aug. 2006), pp. 445–461. ISSN: 0973-7677. DOI: 10.1007/BF02716786. URL: https://doi.org/10.1007/BF02716786.

[6] Brian T Park and Vahe Petrosian. "Fokker-Planck Equations of Stochastic Acceleration : A Study of Numerical Methods". In: (1995).

[7] F Schmidt and C Lamarque. "Computation of the solutions of the Fokker – Planck equation for one and two DOF systems". In: 14 (2009), pp. 529–542. DOI: 10.1016/j.cnsns.2007.09.004.

[8] L Pichler, A Masud, and L A Bergman. "Numerical Solution of the Fokker – Planck Equation by Finite Difference and Finite Element Methods — A Comparative Study". In: (2013). DOI: 10.1007/978-94-007-5134-7.

[9] Kenneth F Caluya, Abhishek Halder, and O C Nov. "Proximal Recursion for Solving the Fokker-Planck Equation". In: *arXiv* 0.3 (2018). arXiv: arXiv:1809.10844v2.

[10] M P Zorzano, H Mais, D- Hamburg, et al. "NUMERICAL SOLUTION FOR FOKKER-PLANCK EQUATIONS IN ACCELERATORS". In: 1.5 (1998), pp. 1825–1827.

[11] M P Zorzano, H Mais, and L Vazquez. "Numerical solution of two dimensional Fokker-Planck equations". In: 98 (1999), pp. 109–117.

[12] Robert Rosenbaum. "A Diffusion Approximation and Numerical Methods for Adaptive Neuron Models with Stochastic Inputs". In: 10.April (2016), pp. 1–20. DOI: 10.3389/fncom.2016.00039.

[13] Anuschka Pauluhn. "Stochastic beam dynamics in storage rings". PhD thesis. Hamburg U., 1993. URL: http://www-library.desy.de/cgi-bin/showprep.pl?DESY-93-198.

[14] Risken H. *The Fokker-Planck Equation. Methods of solution and applications.* Vol. Springer S. 2nd Edition. Berlin, 1989.

[15] Mandli. *10 Hyperbolic-1*. 2019.

[16] Mandli. *09 parabolic*. 2019.