An Analysis of the Fokker–Planck Equation

Patrick McCaw, Michael Hopfinger, Peter Chtcheprov

November 2024

1 Abstract

This study explores the numerical solutions of the Fokker-Planck equation, focusing on stability, accuracy, and convergence under different discretization schemes and a custom-defined potential. The Fokker-Planck equation describes the time evolution of probability densities in systems influenced by drift and diffusion, necessitating robust numerical methods to accurately capture transient and steady-state behavior. We analyzed three numerical schemes: the central difference scheme, the upwind scheme, and the implicit scheme. Stability analysis and simulations revealed that the central difference scheme was conditionally stable and prone to spurious oscillations, making it impractical for most scenarios. The upwind scheme, while unconditionally stable, introduced numerical diffusion that smoothed fine features of the solution. The implicit scheme proved to be the most accurate, and successfully demonstrated the effects of varying diffusion constants, initial distributions, and potential functions.

2 Introduction

The Fokker–Planck Equation (FPE) is a stochastic partial differential equation describing the time evolution of a probability density. The FPE in its most general case for one spatial dimension takes this form:

$$\frac{\partial W(x,t)}{\partial t} = \frac{\partial}{\partial x}\left[-F(x)W(x,t)\right] + \frac{\partial^2}{\partial x^2}\left[D(x,t)W(x,t)\right] \tag{1}$$

The FPE was first discovered in 1914 by Adriaan Fokker studying electromagnetism and shortly after by Max Planck in 1917 who was studying quantum mechanical systems[1]. Most directly, the FPE can be considered a direct description of where one might expect to find a particle at a certain point in time given the influences of both advection and diffusion terms. Here we set out to run a computational study of the behavior of the FPE using a variety of initial conditions and numerical methods as well as gain some analytical intuition for the FPE.

3 Derivation

The derivation of the FPE is a long and complex process, but at its most basic level can be described in this way. Consider a particle subject to both drift from being in a potential field as well as Brownian random motion. From stochastic calculus, this fits the criteria to characterize $X(t)$ as an Itô Process, where $X(t)$ is a random variable that describes the position of the particle at time t.

$$X(t) = \int_0^t \mu\left[X(s), s\right] \mathrm{d}s + \int_0^t \sigma\left[X(s), s\right] \mathrm{d}B(s) \tag{2}$$

Since we are interested in how the particle evolves in time, we can take the derivative of $X(t)$:

$$\mathrm{d}X(t) = \mu\left[X(t), t\right] \mathrm{d}t + \sigma\left[X(t), t\right] \mathrm{d}B(t) \tag{3}$$

To solve, take an arbitrary function of compact support $f(X(t))$ and sub into Itô's Lemma,

$$\begin{aligned} \mathrm{d}f &= f_x \mathrm{d}X(t) + \frac{1}{2} f_{xx} \left[\mathrm{d}X(t)\right]^2 \\ &= f_x \left[\mu \mathrm{d}t + \sigma \mathrm{d}B(t)\right] + \frac{1}{2}\sigma^2 f_{xx} \mathrm{d}t \\ &= \left(\mu f_x + \frac{1}{2}\sigma^2 f_{xx}\right) \mathrm{d}t + f_x \sigma \mathrm{d}B(t) \end{aligned} \tag{4}$$

Where $\mu = \mu[X(t), t]$ and $\sigma = \sigma[(X(t), t]$ for compactness. We can then take the expectation of both sides:

$$\mathbb{E}(\mathrm{d}f) = \mathbb{E}\left[(\mu f_x + \frac{1}{2}\sigma^2 f_{xx})\mathrm{d}t + \sigma f_x \mathrm{d}B(t)\right] \tag{5}$$

The expectation of the Brownian is zero, so

$$\mathbb{E}(\mathrm{d}f) = \mathbb{E}\left[(\mu f_x + \frac{1}{2}\sigma^2 f_{xx})\mathrm{d}t + \sigma f_x \mathrm{d}B(t)\right] \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}(f) = \mathbb{E}(\mu f_x + \frac{1}{2}\sigma^2 f_{xx}) \tag{7}$$

By defining an arbitrary probability $p(x,t)$ and recalling the definition of expectation $\mathbb{E}(f) = \int_{-\infty}^{\infty} f(x)p(x,t)\mathrm{d}x$,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\int_{-\infty}^{\infty} f(x)p(x,t)\mathrm{d}x &= \int_{-\infty}^{\infty} (\mu f_x + \frac{1}{2}\sigma^2 f_{xx})p(x,t)\mathrm{d}x \\ &= \int_{-\infty}^{\infty} \mu(x,t) f_x p(x,t)\mathrm{d}x + \frac{1}{2}\int_{-\infty}^{\infty} \sigma^2(x,t) f_{xx}(x) p(x,t)\mathrm{d}x \end{aligned} \tag{8}$$

Recall that $f(x)$ is an arbitrary function and what we desire to solve for is $p(x,t)$ so we integrate by parts once on the first term and twice on the second,

$$\begin{aligned}\implies \int_{-\infty}^{\infty} f(x)\frac{\partial p(x,t)}{\partial t}\mathrm{d}x = &-\int_{-\infty}^{\infty} f(x)\frac{\partial}{\partial x}[\mu(x,t)p(x,t)]\mathrm{d}x \\ &+\frac{1}{2}\int_{-\infty}^{\infty} f(x)\frac{\partial^2}{\partial x^2}[\sigma^2(x,t)p(x,t)]\mathrm{d}x\end{aligned} \tag{9}$$

The integral is arbitrary:

$$\int_{-\infty}^{\infty} f(x)\left[\frac{\partial p(x,t)}{\partial t}+\frac{\partial}{\partial x}\left(\mu(x,t)p(x,t)\right)-\frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\sigma^2(x,t)p(x,t)\right)\right]\mathrm{d}x = 0 \tag{10}$$

$$\implies \frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left[\mu(x,t)p(x,t)\right]+\frac{1}{2}\frac{\partial^2}{\partial x^2}\left[\sigma^2(x,t)p(x,t)\right] \tag{11}$$

This is the most general form of the Fokker-Planck Equation[2].

4 Materials and Methods

To run numerical simulations of the FPE we used the Spyder IDE as part of the Anaconda suite running Python 3.11.

Before any numerical simulations can take place we need to get the FPE to a form that is more convenient to solve numerically.
We begin the the FPE:

$$\frac{\partial W(x,t)}{\partial t} = \frac{\partial}{\partial x}\left[-F(x)W(x,t)\right]+D\frac{\partial^2}{\partial x^2}\left[W(x,t)\right],$$

and we know that:

$$\frac{\partial U}{\partial x} = -F(x)$$

Plugging in for $-F(x)$:

$$\frac{\partial W(x,t)}{\partial t} = \frac{\partial}{\partial x}\left[\frac{\partial U}{\partial x}W(x,t)\right]+D\frac{\partial^2}{\partial x^2}\left[W(x,t)\right],$$

With initial conditions:

$$U(x) = a_0 kT(a_4x^4+a_3x^3+a_2x^2+a_1x)$$

$$W(x=0,t=0)=0, W(x=1,t=0)=0.$$

In our case the constants are:

$$a_0 = 300, a_1 = -0.38, a_2 = 1.37, a_3 = -2, a_4 = 1.$$

For simplicity (b for Boltzmann):

$$b = \frac{1}{kT}$$

Now solving:

$$\frac{\partial W(x,t)}{\partial t} = b\frac{\partial}{\partial x}\left[\frac{\partial U}{\partial x}W(x,t)\right] + D\frac{\partial^2}{\partial x^2}\left[W(x,t)\right].$$

Applying the product rule to the first term on the right-hand side, we obtain:

$$\frac{\partial W(x,t)}{\partial t} = b\frac{\partial^2 U}{\partial x^2}W(x,t) + b\left(\frac{\partial U}{\partial x}\right)\left(\frac{\partial W(x,t)}{\partial x}\right) + D\frac{\partial^2 W(x,t)}{\partial x^2}. \tag{12}$$

In the following section we will provide methods and motivations on solving the FPE, beginning with seeing what the analytical form of the FPE can tell us and then exploring numerical techniques.

4.1 Analytical Exploration

To know if our numerical methods are effective, it is helpful to first analyze W under certain conditions. We first consider the case of zero diffusion $D = 0$, where the second derivative with respect to x drops out. The most basic thing we can do is observe the behavior of the potential function, which determines the force in the advection term. Physically, we expect to find the particle, and observe relatively large values of W, where the potential is lowest. Mathematically, we can solve for the extrema of the potential function – this is done trivially by simply finding the roots of its derivative. We find that the given potential has two minima at $x' = \{0.27, 0.77\}$, and a local maximum at $x' = 0.46$: These are the equilibrium positions at zero diffusion. Force is zero at these points, so (12) becomes simply

$$\frac{\partial W(x',t)}{\partial t} = -\frac{1}{kT}\left.\frac{\partial F}{\partial x}\right|_{x=x'} W(x',t) \tag{13}$$

which is a separable ODE. The RHS is simply a constant multiplied by W, so we can rewrite it as $-\kappa W(x',t)$. Dividing through by W and integrating gives us the time evolution at these critical points in a situation without diffusion:

$$\kappa \equiv \frac{1}{kT}\left.\frac{\partial F}{\partial x}\right|_{x=x'} \tag{14}$$

$$\frac{\partial W(x',t)}{\partial t} = -\kappa W(x',t) \tag{15}$$

$$\frac{\partial W(x',t)}{\partial t}\frac{1}{W(x',t)} = -\kappa \tag{16}$$

$$W(x',t) = Ae^{-\kappa t} \tag{17}$$

Recall that force is the negative derivative of potential, and thus κ has sign opposite to the concavity at x'. Thus, at the maximum $x = 0.46$, W exhibits

exponential decay. At the other two extrema, W exhibits exponential growth and blows up as we move forward in time. This is consistent with what we expect physically: in a situation without diffusion, the particle will be increasingly likely to be found at one of the potential minima, and less likely to be anywhere else. Regarding our numerical methods, then, we expect to see two spikes at $x = \{0.27, 0.77\}$ as we take small values of D. For the equation with diffusion, we can solve for the equilibrium distribution, what W will converge to after a relatively long period of time. Since we want the distribution where W no longer changes with time, we can set the time derivative, or the LHS of Eq. 12 to zero:

$$0 = b\frac{\partial^2 U}{\partial x^2}W(x,t) + b\left(\frac{\partial U}{\partial x}\right)\left(\frac{\partial W(x,t)}{\partial x}\right) + D\frac{\partial^2 W(x,t)}{\partial x^2}. \tag{18}$$

The solution then becomes

$$W = A\mathrm{e}^{-\frac{U(x)}{D}} \tag{19}$$

which is the form we expect W to converge to when we step through time numerically. If we plot this function with varying values of D, we find that it has two local maxima corresponding to the equilibrium positions $x = \{0.27, 0.77\}$ and a local minimum at $x = 0.46$. For higher D, the probability distribution is flatter, and for lower D, the two spikes that we predicted for the no diffusion case begin to appear. In analyzing the effectiveness of our numerical methods, we will compare our W at relatively large t to the known solution (19).

4.2 Numerical Methods

Solving PDE's numerically follows this general prescription:

1. Create a discrete grid of the variables involved in the PDE, in our case space and time.

2. Assign Boundary Conditions at points $(x_0, t_{0\ldots N})$ and $(x_{J+1}, t_{0\ldots N})$. Note, this will result in $J+2$ spatial points, J points for that are solved for and 2 points at each end that are used to store values for boundary conditions. Additionally, there are $N+1$ temporal points, N points we desire to know, and 1 point assigned for t_0 at the beginning of the algorithm.

3. Assign an Initial distribution for time t_0 that can be any arbitrary function (like a normal distribution) that meets the criteria to be a distribution

4. Choose a solver method

   (a) Explicit

   (b) Implicit

   (c) Crank Nicholson

   (d) Other methods also exist

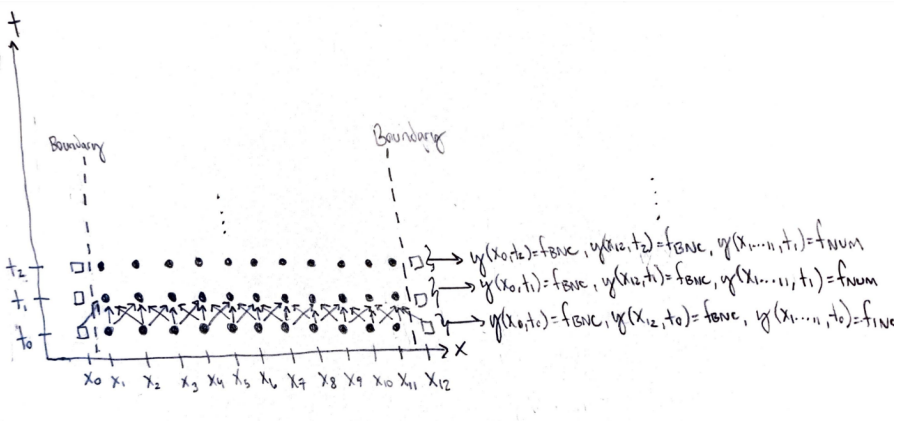

Figure 1: Visualization of Explicit Method for $J = 10$

5. Discretize the derivatives of the PDE based on whichever solver method you selected using techniques such as symmetric first derivatives, asymmetric first derivatives, and finite differencing for higher order derivatives. This will give you the prescription for your update steps.

6. Based on whichever solver method was chosen, loop through the current spatial points at that time and assign values for the next step

7. Repeat step 5 and 6 until $t = N$

Using the above language algorithm we embark on numerically solving Equation (12), beginning with the boundaries.

Boundary Conditions At every time step, the boundary conditions are updated as:

$$W(0, t_n) = f_{\text{BNC}}(0, W),$$
$$W(J+1, t_n) = f_{\text{BNC}}(J+1, W).$$

Initialization As before, we discretize the spatial domain into J points and the temporal domain into N points, with step sizes:

$$\Delta x = x_{j+1} - x_j, \quad \Delta t = t_{n+1} - t_n.$$

The initial condition $W(x, t = 0)$ is:

$$W(x, 0) = f_{\text{INC}}(x).$$

$$f_{\text{INC}}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$
$$\mu = 0.5, \sigma = 0.1$$

The boundary conditions are imposed at each time step:

$$W(0,t_n) = f_{\text{BNC}}(0,W), \quad W(J+1,t_n) = f_{\text{BNC}}(J+1,W).$$

Next we decide on the numerical approach. We explore two numerical methods: explicit and implicit. For the explicit method, we employ a staggered scheme. The first time step uses asymmetric finite difference approximations to handle the boundary conditions, while subsequent time steps use symmetric approximations. This leads to the following discretized FPE for the first time step. Notice that derivatives of U are not discretized because U is easily solved analytically:

$$\begin{aligned}\frac{W_j^{n+1} - W_j^n}{\Delta t} = b\left(\frac{\partial^2 U}{\partial x^2} W_j^n + \frac{\partial U}{\partial x}\left(\frac{W_{j+1}^n - W_{j-1}^n}{2\Delta x}\right)\right) \\ + D\left(\frac{W_{j+1}^n - 2W_j^n + W_{j-1}^n}{\Delta x^2}\right)\end{aligned} \tag{20}$$

For visual clarity, we group common terms of W in this way:

$$\begin{aligned}W_j^{n+1} = W_j^n + \left(-\frac{b\Delta t}{2\Delta x}\frac{\partial U}{\partial x} + D\frac{\Delta t}{\Delta x^2}\right) W_{j-1}^n \\ + \left(b\Delta t\frac{\partial^2 U}{\partial x^2} - 2D\frac{\Delta t}{\Delta x^2}\right) W_j^n + \left(\frac{b\Delta t}{2\Delta x}\frac{\partial U}{\partial x} + D\frac{\Delta t}{\Delta x^2}\right) W_{j+1}^n\end{aligned} \tag{21}$$

From this formula for the first time step of W we can easily see that it may be more efficient to condense these terms in the parentheses into parameters and simply work with these as opposed to large unwieldy formulas. For the remainder of this paper that is what we will do. Continuing with our explicit scheme we define two sets of parameters.

First Step Parameters
For the first time step, the coefficients are:

$$\alpha = D\frac{\Delta t}{\Delta x^2},$$

$$\beta(x) = -\frac{b\Delta t}{2\Delta x}\frac{\partial U}{\partial x} + \alpha,$$

$$\gamma(x) = b\Delta t\frac{\partial^2 U}{\partial x^2} - 2\alpha,$$

$$\delta(x) = \frac{b\Delta t}{2\Delta x}\frac{\partial U}{\partial x} + \alpha.$$

Subsequent Step Parameters
For further steps, the parameters are:

$$\epsilon(x) = 2\beta(x),$$

$$\zeta(x) = 2b\Delta t\frac{\partial F}{\partial x} - 4\alpha,$$
$$\eta(x) = 2\delta(x).$$

First Time Step: Using the coefficients $\beta(x)$, $\gamma(x)$, and $\delta(x)$, the update for the first time step is:

$$W_j^{(1)} = \beta(j\Delta x)W_{j-1}^{(0)} + \gamma(j\Delta x)W_j^{(0)} + \delta(j\Delta x)W_{j+1}^{(0)} + W_j^{(0)},$$

where $W^{(0)}$ represents the initial condition.

Subsequent Time Steps: For $n > 1$, the update rule uses the coefficients $\epsilon(x)$, $\zeta(x)$, and $\eta(x)$:

$$W_j^{(n+1)} = W_j^{(n)} + \epsilon(j\Delta x)W_{j-1}^{(n)} + \zeta(j\Delta x)W_j^{(n)} + \eta(j\Delta x)W_{j+1}^{(n)}.$$

4.3 Implicit Method

The discretization for the implicit method is extremely similar to the explicit step, except that we are now estimating the derivative at the $n+1$ timestep, and no longer need to take a symmetric time derivative. Thus, our discretized equation becomes

$$\begin{aligned}\frac{W_j^{n+1} - W_j^n}{\Delta t} = b\left(\frac{\partial^2 U}{\partial x^2}W_j^{n+1} + \frac{\partial U}{\partial x}\left(\frac{W_{j+1}^{n+1} - W_{j-1}^{n+1}}{2\Delta x}\right)\right)\\ +D\left(\frac{W_{j+1}^{n+1} - 2W_j^{n+1} + W_{j-1}^{n+1}}{\Delta x^2}\right)\end{aligned} \tag{22}$$

Notice that we now claim the point W_j^{n+1} can be found from information about the surrounding *subsequent* points at time $n+1$. This relationship is made clear by solving the above expression for W_j^{n+1}.

$$\begin{aligned}W_j^n = \ -b\Delta t\left(\frac{\partial^2 U}{\partial x^2}W_j^{n+1} + \frac{\partial U}{\partial x}\left(\frac{W_{j+1}^{n+1} - W_{j-1}^{n+1}}{2\Delta x}\right)\right)\\ -D\Delta t\left(\frac{W_{j+1}^{n+1} - 2W_j^{n+1} + W_{j-1}^{n+1}}{\Delta x^2}\right) + W_j^{n+1}\end{aligned} \tag{23}$$

As before, we can split this equation up based on spatial points and assign variables based on the coefficients:

$$\begin{aligned}W_j^n = \left(\frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \frac{D\Delta t}{\Delta x^2}\right)W_{j-1}^{n+1} + \left(-b\Delta t\frac{\partial^2 U}{\partial x^2} + \frac{2D\Delta t}{\Delta x^2} + 1\right)W_j^{n+1}\\ + \left(-\frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \frac{D\Delta t}{\Delta x^2}\right)W_{j+1}^{n+1}\end{aligned} \tag{24}$$

$$\alpha \equiv \frac{D\Delta t}{\Delta x^2} \tag{25}$$

$$\beta \equiv \frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \frac{D\Delta t}{\Delta x^2} = \frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \alpha \tag{26}$$

$$\gamma \equiv -b\Delta t\frac{\partial^2 U}{\partial x^2} + \frac{2D\Delta t}{\Delta x^2} + 1 = -b\Delta t\frac{\partial^2 U}{\partial x^2} + 2\alpha + 1 \tag{27}$$

$$\delta \equiv -\frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \frac{D\Delta t}{\Delta x^2} = -\frac{\partial U}{\partial x}\frac{b\Delta t}{2\Delta x} - \alpha \tag{28}$$

Our update step thus becomes

$$W_j^n = \beta W_{j-1}^{n+1} + \gamma W_j^{n+1} + \delta W_{j+1}^{n+1} \tag{29}$$

This expression does not look the particularly useful; after all, we have no information about W at time $n+1$. However, we can rewrite this expression in terms of the vectors W^n and W^{n+1}, and express Eq. 29 as a matrix transformation:

$$\begin{bmatrix} W_0^n \\ W_1^n \\ W_2^n \\ \vdots \\ W_{J+1}^n \end{bmatrix} = \begin{bmatrix} \gamma & \delta & 0 & 0 & 0 & 0 \\ \beta & \gamma & \delta & 0 & 0 & 0 \\ 0 & \beta & \gamma & \delta & 0 & 0 \\ 0 & 0 & \beta & \ddots & \ddots & \vdots \\ & & & \ddots & \ddots & \delta \\ 0 & 0 & 0 & \cdots & \beta & \gamma \end{bmatrix} \begin{bmatrix} W_0^{n+1} \\ W_1^{n+1} \\ W_2^{n+1} \\ \vdots \\ W_{J+1}^{n+1} \end{bmatrix} \tag{30}$$

In this way, you can see the elements $W_j^n + 1$ are hit with the appropriate γ, while the W_{j-1} elements correspond to β and W_{j+1} to δ, as the update step dictates. The individual variables of the matrix vary along the diagonals, but the matrix as a whole is constant and tridiagonal. Rewriting the matrix as A,

$$W^n = \mathbf{A}W^{n+1} \tag{31}$$

Where W represents the vector at all points in space. We can left multiply by the inverse of $\mathbf{A}$ to solve for the $n+1$ timestep:

$$\mathbf{A^{-1}}W^n = \mathbf{A^{-1}}\mathbf{A}W^{n+1} \tag{32}$$

$$W^{n+1} = \mathbf{A^{-1}}W^n \tag{33}$$

We must also enforce the boundary condition at each step, this is done in the same manner as the explicit method.

5 Results

Our study of the Fokker-Planck equation focused on the numerical stability and accuracy of three discretization schemes: the central difference scheme, the upwind scheme, and the implicit scheme. The equation requires careful handling of stability and accuracy due to its advective and diffusive components.

5.1 Explicit Method

5.1.1 Central Difference Scheme

We initially implemented the central difference scheme for spatial discretization due to its second-order accuracy. However, stability analysis revealed significant limitations for this method in time-dependent simulations. Using the Von Neumann stability analysis, we found that the central difference scheme was conditionally stable, requiring strict adherence to the Courant-Friedrichs-Lewy (CFL) condition in order to prevent instability. Specifically, the CFL condition imposed an upper limit on the time step based on the spatial grid size and the diffusion coefficient:

$$\Delta t < \frac{(\Delta x)^2}{2D} \tag{34}$$

In simulations where the diffusion coefficient was small, the CFL condition mandated impractically small time steps. This constraint not only increased computational cost but also failed to eliminate numerical instability completely in scenarios involving sharp gradients or rapid changes in the probability density. Numerical results showed oscillations and exponential growth of errors, making the central difference scheme impractical for our purposes. The following figures show the results of this method with 100 support points, and the initial condition being a Gaussian with $\sigma = 0.1$ centered at 0.5.

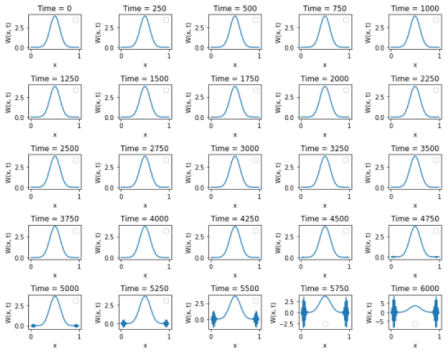

Figure 2: Time evolution for $D = 1$, $\Delta t = 0.1\mu$s

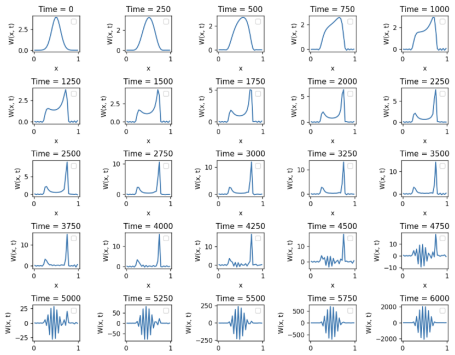

Figure 3: Time evolution for $D = 0.1$, $\Delta t = 0.1$ms

5.1.2 Upwind Scheme

To overcome the instability of the central difference scheme, we implemented the upwind scheme. Unlike the central difference scheme, the upwind scheme is stable under the CFL condition for advective-dominant problems. This scheme has the stability condition

$$F\frac{\Delta t}{\Delta x} < 1 \tag{35}$$

The upwind scheme is a numerical method used to approximate solutions to partial differential equations, particularly those involving advective or convective terms. It is a first-order accurate finite difference method designed to handle situations where the direction of flow significantly impacts the solution. This method incorporates numerical dissipation to maintain stability, making it especially useful for problems where sharp gradients or steep fronts can lead to instability in other schemes, such as the central difference method. The upwind scheme accounts for the direction of drift in the system. For a one-dimensional advection-diffusion equation, the method chooses the direction based on the sign of F(x): if the sign of F(x) is positive, then we would take the finite difference from the right. If the sign of F(x) is negative, then we would take the finite difference from the left.

Simulations using the upwind scheme produced smooth and stable solutions across a range of parameter settings. For instance, in the case of a potential well, the upwind scheme successfully captured the gradual redistribution of probability density without introducing oscillations or instability.

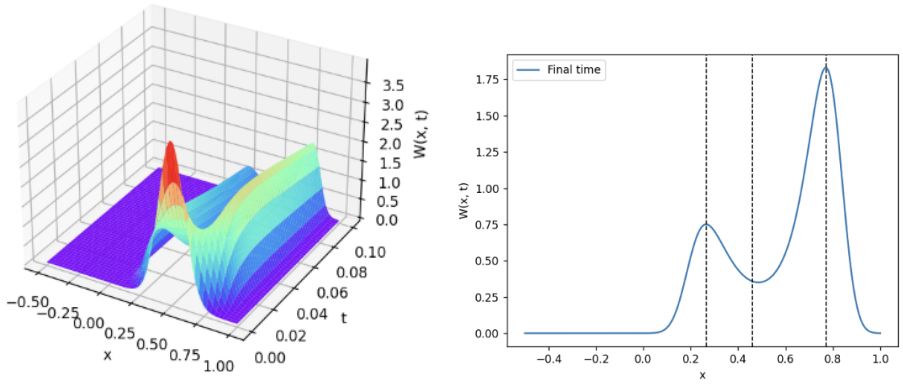

Figure 4: Time evolution and equilibrium positions for $D = 1$, $\Delta t = 0.1\mu$s

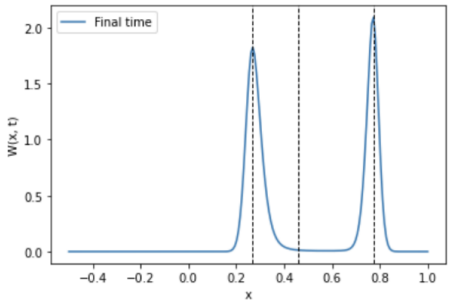

Figure 5: Equilibrium distribution for $D = 0.1$, $\Delta t = 0.1$ms

5.2 Implicit Method

To further improve stability and accuracy, we implemented the implicit scheme, which is known for being unconditionally stable. Unlike the explicit schemes, the implicit method allows larger time steps without compromising stability, making it particularly well-suited for diffusive systems. Numerical results from the implicit scheme demonstrated exceptional stability across all varying parameters. The scheme produced accurate and smooth solutions that closely matched the expected theoretical distributions. One drawback of the implicit scheme was its computational cost per time step, as solving the resulting linear system at each iteration required more resources than the explicit methods. However, the stability of the implicit scheme allowed us to use much larger time steps, reducing the total number of steps needed and offsetting the higher per-step cost. Once more, we run the program at varying values of D, with the Gaussian initial condition. We notice that the equilibrium distribution lines up with the potential function, in that the x-values at the minima of the poten-

tial correspond to maxima in the probability distribution, and vice versa. This matches with our predictions from section 4.1, and also makes sense physically: the particle seeks areas of low potential, so it is most likely to be found there. Note that W evolves very quickly, so taking the distribution at $t = 1$ seconds is generally sufficient to estimate the equilibrium distribution. Unless otherwise specified, the timestep used was 10^{-4} seconds with 100 support points across x. This allows sufficient resolution in our case, but one advantage of the implicit method is that it does not suffer from the Δt constraint of the explicit step, and thus does increasing J is much less computationally expensive. Note also that for the 3D plots, the z axis represents W, and is missing a normalization constant. Our first trial, with $D = 1$, is shown below:

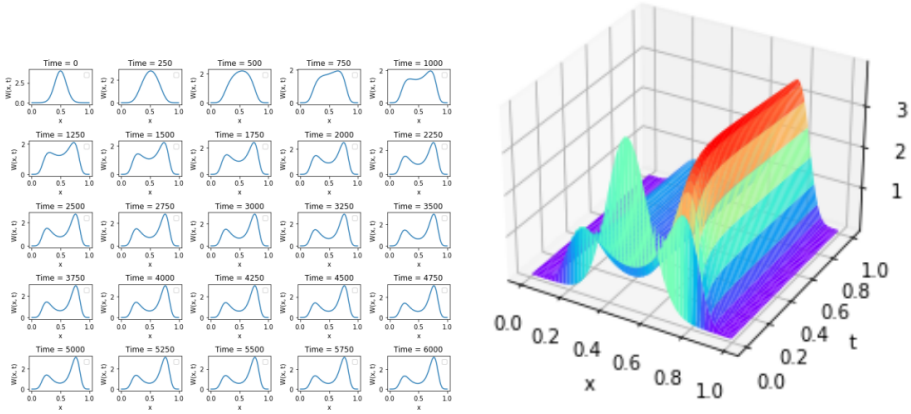

Figure 6: Time evolution and visualization of W, for $D = 1$

The image on the left shows that W becomes relatively stable at just a few tenths of a second.

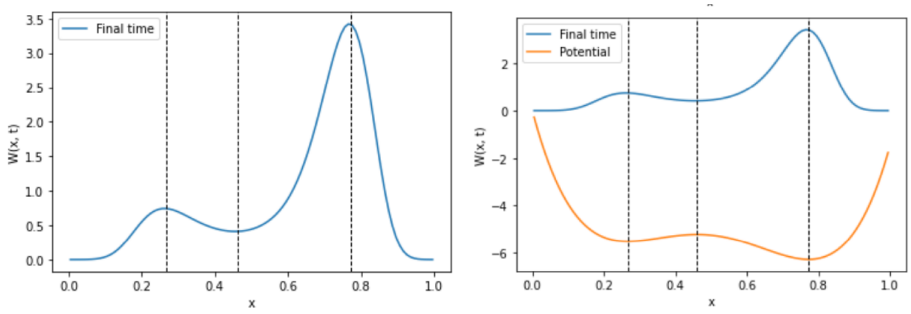

Figure 7: Comparison of the equilibrium position with the potential function. The vertical lines represent the extrema points discussed in Section 4.1

This trial produced the expected results. W has two maxima corresponding to the potential minima, and goes to zero at the boundaries. Now we want to see how the system reacts to different diffusion constants.

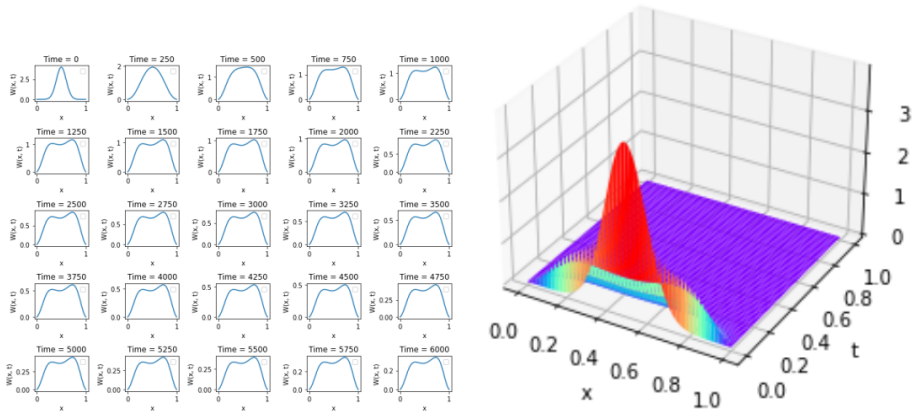

Figure 8: Time evolution and visualization of W, for $D = 5$

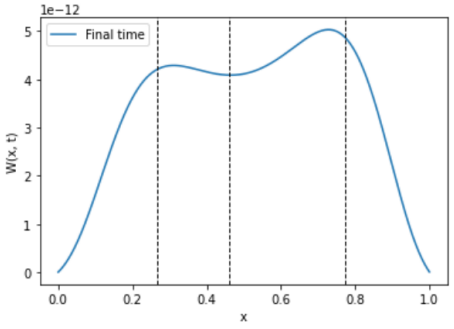

Figure 9: Equilibrium distribution for $D = 5$

For a larger diffusion constant of 5, the system evolves significantly more rapidly and we end up with a flatter equilibrium distribution, as predicted. The extrema still line up with the potential, but not quite as well, even when a lower time step is taken.

From our earlier analysis of the given equilibrium distribution, we expect lower values of diffusion to exhibit sharper spikes at the equilibrium positions. Our implicit method demonstrates this phenomena.

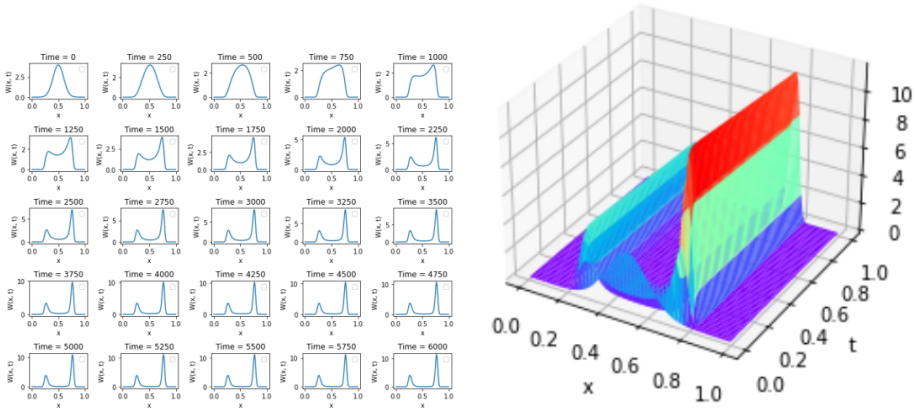

Figure 10: Time evolution and visualization of W, for $D = 0.1$

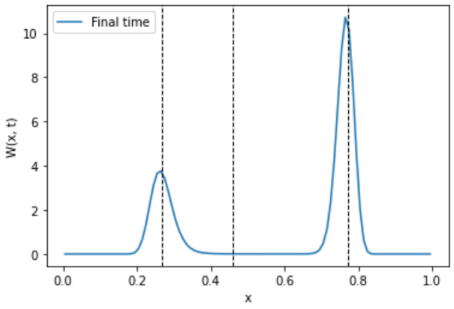

Figure 11: Equilibrium distribution for $D = 0.1$

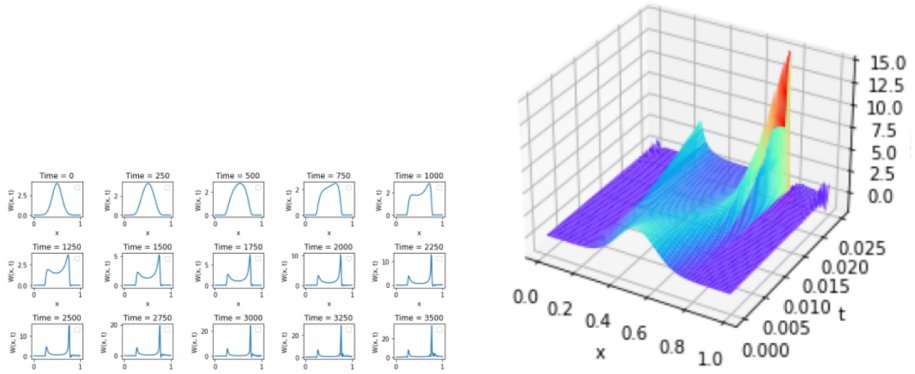

Figure 12: Time evolution and visualization of W, for $D = 0.01$

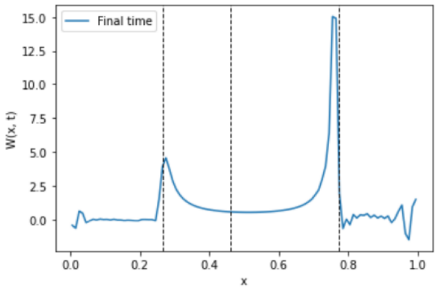

Figure 13: Probability distribution for $D = 0.1$ at $t = 0.025$

Down to about $D = 0.1$, our solutions are stable and match what we expect. Lower values of diffusion, however, are unstable with our method, as illustrated by Figures 12 and 13. For a small value of t, they seem to take the proper form, but beyond this they blow up, and the oscillations near the boundaries grow larger in both the positive and negative direction. We know this is incorrect because it is impossible to have a negative probability. The instability here comes from our handling of the boundary conditions and continually setting $x = 0$ and $x = 1$ to zero. At very low diffusion, this causes the ghost zones to over correct and returns increasingly large errors. A more precise computer could remedy this.

Now that we have confidence in the implicit method, we can change the initial distribution to see how that affects both the time evolution of the probability and the equilibrium distribution. We will use $D = 1$, $\Delta t = 0.1$ms and $J = 100$.

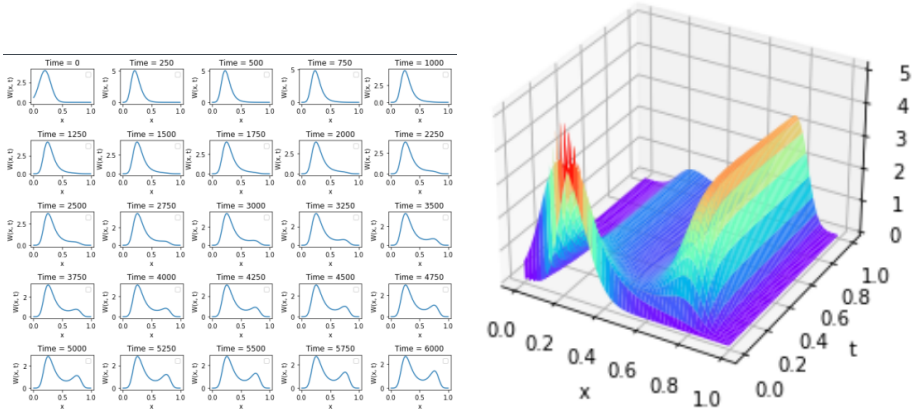

Figure 14: Time evolution and visualization of W, for $\mu = 0.2$, $\sigma = 0.1$

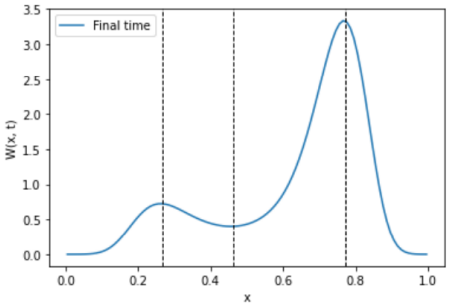

Figure 15: Equilibrium distribution for for $\mu = 0.2$, $\sigma = 0.1$

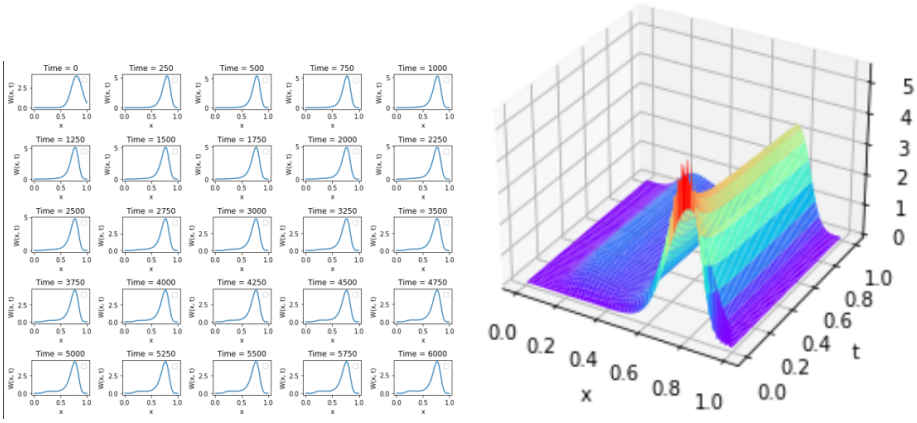

Figure 16: Time evolution and visualization of W, for $\mu = 0.8$, $\sigma = 0.1$

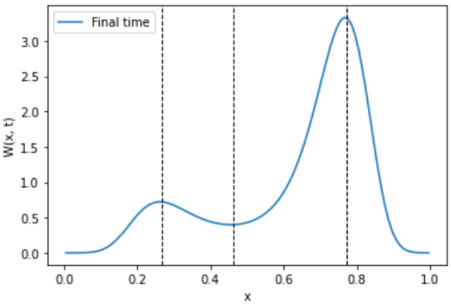

Figure 17: Equilibrium distribution for for $\mu = 0.8$, $\sigma = 0.1$

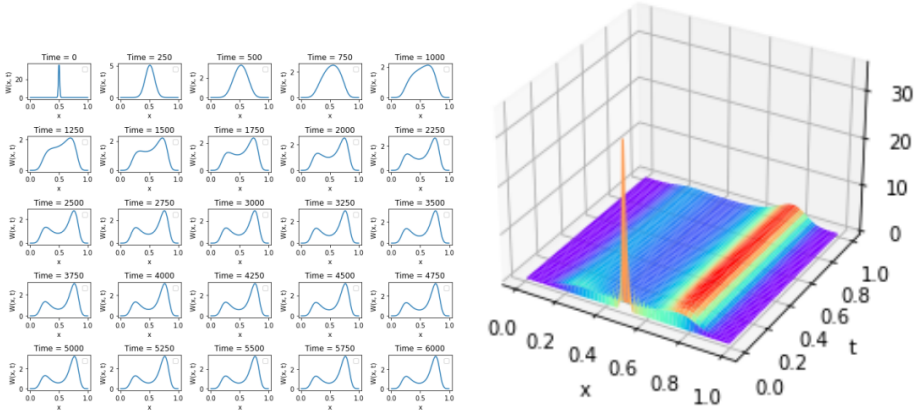

Figure 18: Time evolution and visualization of W, for $\mu = 0.5$, $\sigma = 0.01$

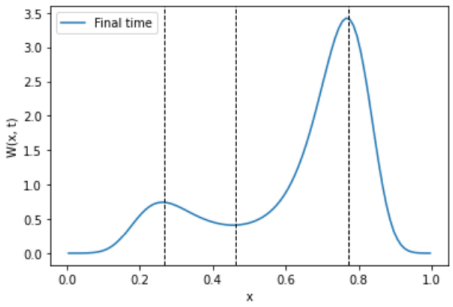

Figure 19: Equilibrium distribution for for $\mu = 0.5$, $\sigma = 0.01$

The initial distribution affects the early timesteps of W, however, the numerical solution consistently converged to the equilibrium distribution defined by U(x). This demonstrated the robustness of the implicit scheme in handling different starting points and its ability to resolve transient dynamics accurately while ensuring convergence to the correct steady state.

Overall, the implicit scheme is much preferable to either explicit method. The results demonstrate much less instability, and the lack of restrictions on Δt in regards to Δx mean that higher resolution could be achieved if necessary. Our implicit method is an effective and accurate representation of the probability distribution of a particle in a potential well as the situation is evolved forward in time.

5.3 New Potential

To study how W operates in the presence of a different potential, we generate our own:

$$F(x) = -[\sinh(x-2) + \ln\cosh(x-2) + 1.5x] \tag{36}$$

$$U = \cosh(x-2) + \tanh(x-2) + 1.5 \tag{37}$$

$$\frac{\partial U}{\partial x} = \sinh(x-2) + \operatorname{sech}(x-2) \tag{38}$$

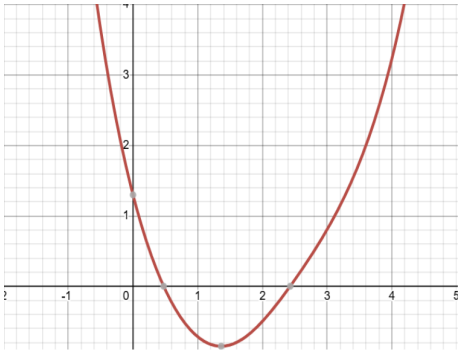

Figure 20: New potential as a function of x

Running our implicit method on this function yields the expected results: the maximum of W corresponds to the potential minimum. The following trials used $J = 100$, $\Delta t = 0.1$ms, and $\mu = 0.5$, $\sigma = 0.1$ for the initial Gaussian.

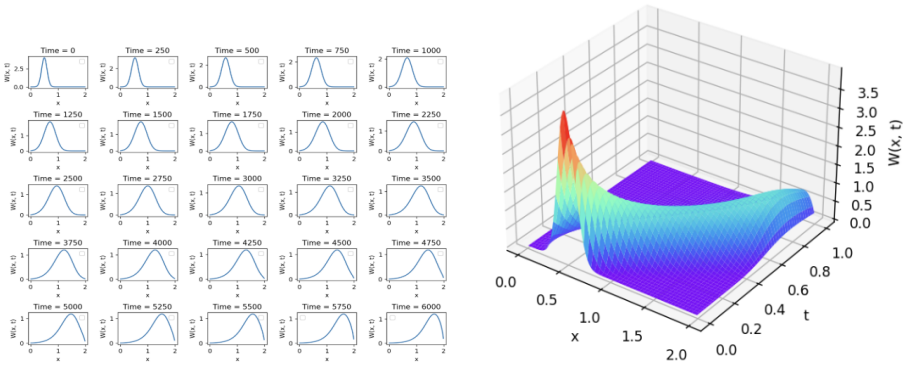

Figure 21: Time evolution and visualization of W, for $D = 0.1$

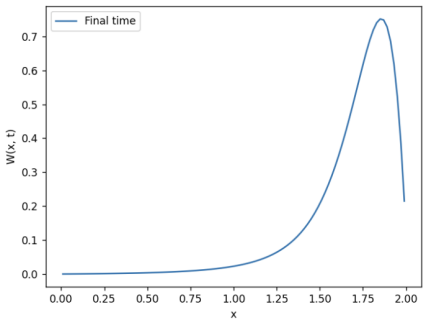

Figure 22: Equilibrium distribution for for $D = 0.1$

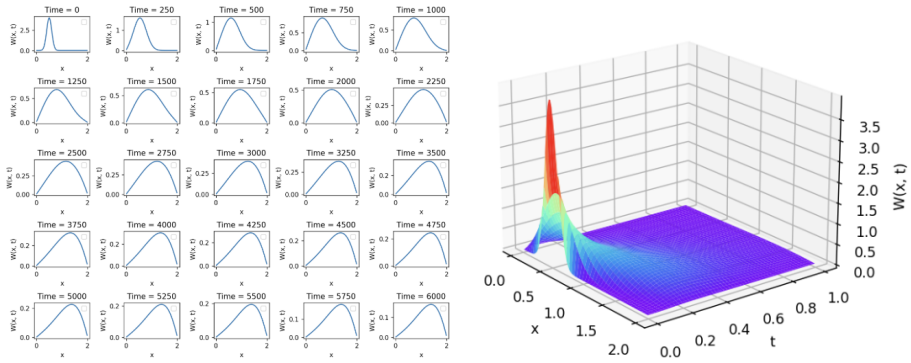

Figure 23: Time evolution and visualization of W, for $D = 1$

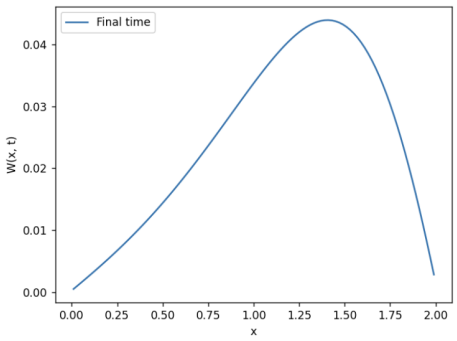

Figure 24: Equilibrium distribution for for $D = 1$

As expected, the larger diffusion constant leads to a flatter equilibrium distribution and a more rapid evolution with respect to time. W converges such that its

maximum at equilibrium matches the potential minimum. Our implicit method is thus an accurate analysis of our case of Fokker-Planck equation, regardless of the potential governing the advection term.

6 Conclusion

In this paper, we derived the general form of the Fokker-Planck Equation (FPE), conducted an analytical exploration of its properties, and implemented numerical solutions for various potential functions $U(x)$. To achieve this, we employed both explicit and implicit numerical methods, analyzing their respective advantages and disadvantages with particular emphasis on numerical stability and computational efficiency.

However, the question may linger, why is this Fokker-Planck Equation significant? In essence, its utility lies in its versatility. The FPE is an exceptionally general partial differential equation with applications spanning numerous fields, including physics, finance, and beyond. While one derivation of the FPE stems from physical principles, it can also be derived completely independently through stochastic modeling and statistics, underscoring its prevalence. From plasma physics to Wall Street, mastering the FPE and its numerical solutions provides valuable tools for advancing research and applications across a broad spectrum of disciplines.

References

[1] Wikipedia. (2024, October 4). *Fokker–Planck equation*. Retrieved from `https://en.wikipedia.org/wiki/Fokker%E2%80%93Planck_equation`

[2] García-Palacios, J. L. (2004, May 1). *Introduction to the theory of stochastic processes and Brownian motion problems* [Lecture Notes].

[3] Heitsch, F. (2023, August 20). *PHYS 332: Notes* [Class Notes].

[4] Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (2002). *Numerical Recipes in C* (2nd ed.). Cambridge University Press.

[5] Wikipedia. (2024, September 28). *Von Neumann stability analysis*. Retrieved from `https://en.wikipedia.org/wiki/Von_Neumann_stability_analysis`