# Exploring Stochastic Differential Equations

Alan Chen

May 2022

## Contents



# 1 Introduction

In this report, we investigate and simulate the evolution of various continuous time processes governed by Itô SDEs. In §2, we look at the Ornstein Uhlenbeck Process. In §3, we look at Feller diffusion accompanied with a stopping time. In this section, we do some computations regarding the observables of the stopping time, which should outline similar computations for §4 and §5 which discuss and visualize SDEs based on a matrix completion problem with stopping conditions as well. Finally, in §6, we present some novel visualizations of the $n$ dimensional processes in the

previous sections by reducing them into lower dimensions using t-stochastic neighbor embedding and principal component analysis.

All code used for this project is linked in the appendix (§8).

### 1.1 Notation

$dB^{(n)}$ will denote standard Brownian motion in $n$-dimensions, with no exponent meaning in one dimension. Additionally, we define

$$\Delta B_k^{\sigma^2} = \sigma \mathcal{N}(0,1) \tag{1.1}$$

as the discrete approximation of an RV of standard Brownian motion. When representing this in $n$ dimensions, the $B$ will be bolded.

## 2 Ornstein-Uhlenbeck

We begin with some simple visualizations of the ubiquitous example of an SDE that incorporates a deterministic drift term and a stochastic noise term: the Ornstein-Uhlenbeck process.

**Definition 1.** (Ornstein-Uhlenbeck Process) The **Ornstein-Uhlenbeck Process** is the stochastic process governed by the solution $X$ to the Itô stochastic differential equation:

$$dX = -aXdt + dB. \tag{2.1}$$

Sometimes the equation is also seen with a $\sqrt{\frac{2}{\beta}}$ factor in front of $dB$. For this report, we will specifically consider $\beta = 2$ just to simplify the calculations a bit.

Of course, simulating this process and visualizing on a computer must be done through a discrete approximation of (2.1). By discretizing $[0, \infty)$ into time steps of size $h$, we can create an approximation $Y$ for $X$:

$$Y_{k+1} = Y_k - aY_k h + \Delta B_k^h, \quad k = 0, 1, 2 \ldots, \tag{2.2}$$

with $\Delta B_k^h$ defined as in (1.1).

With this simple recurrence relation in hand, we can very easily generate realizations of this stochastic process. Figure 1 displays one of these realizations for varying step sizes, 100 steps, initial condition $Y_0 = 0$, and drift coefficient $a = 1$.

In, for example, the left plot for $h = 0.1$, we can observe the drift term causing a progressive uplift of the process. However, with the other plots, we can see an equilibrium distribution forming, with a heavier concentration of points in the "middle" and less on the outside, suggesting a distribution similar to a Gaussian is the equilibrium distribution.

**Remark.** One can solve the **Fokker-Planck (Kolmogorov Forward) equation** to find the explicit form of the density for this equilibrium distribution (which is indeed a Gaussian). By coefficient matching, we know that (2.1) is a stochastic gradient flow with energy function

$$E(x) = \frac{1}{2} a x^2. \tag{2.3}$$

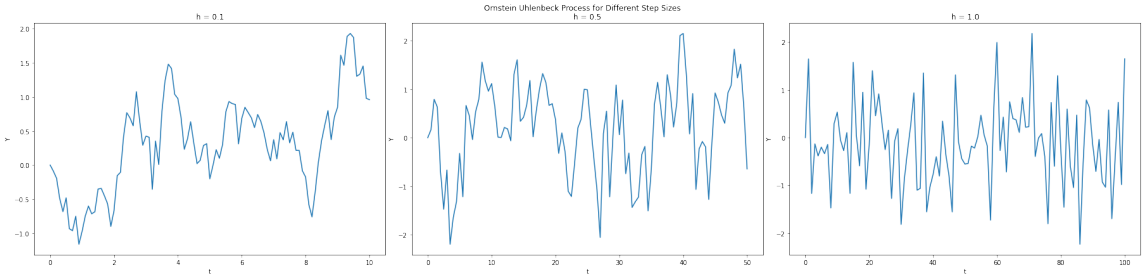



Figure 1: One set of realizations of Ornstein Uhlenbeck for different step sizes.

Recalling the Fokker Planck equation, we see that the equilibrium distribution of (2.1) (call it $p(x)$, as it is invariant over time) must satisfy

$$p_t = 0 = \frac{1}{2}\Delta p + \nabla \cdot (p \nabla E). \tag{2.4}$$

We showed that the Gibbs measure solves the Fokker-Planck equation, so we can conclude that our equilibrium distribution for (2.1) is

$$p(x) = \frac{e^{-2((1/2)ax^2)}}{Z_2} = \frac{e^{-ax^2}}{Z_2}, \quad Z_2 = \int_{\mathbb{R}} e^{-ax^2}, \tag{2.5}$$

or a Gaussian as suggested by Figure 1.

Alternatively, using a clever ansatz of a Gaussian with mean $m(t)$ and variance $\rho(t)$ inspired by the solution to the ODE $\dot{x} = -ax$ combined with applications of Itô's Lemma/Calculus, we can arrive at the same conclusion as above.

# 3 Feller Diffusion

We now turn to another commonly studied stochastic differential equation.

**Definition 2.** (Feller Diffusion Process) The Feller Diffusion Process is described by the solution $Z$ to the Itô stochastic differential equation:

$$dZ = \sqrt{Z} dB, \quad Z_0 > 0. \tag{3.1}$$

Notice the strictness of the condition on $Z_0$, as if $Z_0 = 0$ the process would become degenerate and non-interesting.

We can observe the Feller diffusion for some number of iterations, but we note that (3.1) no longer becomes well defined if $Z_t \leq 0$. So, we are motivated to define the following stopping time and terminate the process after it occurs.

**Definition 3.** (Extinction Time) Formally, **extinction time** of Feller diffusion process is a stopping time $T$ defined as

$$T = \min\{t \geq 0 : Z_t \leq 0\}. \tag{3.2}$$

Before proceeding into discrete approximation and simulation, we can figure out some things about the extinction time to get a better feel for it. First, we will consider a boundary/exit time problem. Namely we wish to answer the problem: given two barriers 0 and $b$, what is the probability that $Z$ hits 0 first?

We recall that for a simple Brownian motion process, the probability of this occurring is $\frac{b-1}{b}$. We will show that even though the amplitude of the noise is now variant on the location in space, the probability is the same through first step analysis.

Let $f(x) = \mathbb{P}(Z \text{ hits } 0 \text{ before } b | Z_0 = x)$. Using first step analysis, we can see that

$$f(x) = \mathbb{E}[f(x + \sqrt{x}B_{\Delta t})], \quad \lim_{x \to 0^+} f(x) = 1 \text{ and } f(b) = 0. \tag{3.3}$$

By Taylor expansion and dropping the higher order terms, we find that

$$f(x) = f(x) + f'(x)\mathbb{E}(\sqrt{x}B_{\Delta t}) + \frac{1}{2}f''(x)\mathbb{E}[(\sqrt{x}B_{\Delta t})^2]. \tag{3.4}$$

Computing these expectations and rearranging while being careful to note that $x$ cannot be 0 gives

$$f''(x) = 0, \tag{3.5}$$

which is exactly the same ODE in the simple Brownian motion case. We can solve and plug in the initial conditions to find that $f(x) = -\frac{1}{b} + 1$, or that

$$f(x) = \frac{b-1}{b}. \tag{3.6}$$

**Proposition 3.1.** *Given a Feller diffusion process with initial state $Z_0 = 1$ and two boundaries at $x = 0$ and $x = b$,*

$$\mathbb{P}(\text{Z hits } 0 \text{ before } b) = \frac{b-1}{b}. \tag{3.7}$$

The reason why we introduced this problem is for the next step: using the formula in (3.7), we can prove that the stopping time in (3.2) is finite with probability 1.

**Proposition 3.2.** *Given a Feller diffusion process with initial state $Z_0 = 1$ and $T$ as defined in (3.2),*

$$\mathbb{P}(T < \infty) = 1, \tag{3.8}$$

*or that $T$ is finite with probability 1.*

*Proof.* Consider the path space $\Omega$ of realizations of the Feller diffusion process, and define $\mathcal{A}$ as the subset of paths in this path space that have finite $T$.

$$\mathcal{A} = \{\omega \in \Omega : T(\omega) < \infty\}.$$

Let $\mathcal{A}_b$ be the event where the process hits 0 before an arbitary right barrier $b > 1$. First, we note that this is a subset of $\mathcal{A}$, so we have that $\mathbb{P}(\mathcal{A}) \geq \mathbb{P}(\mathcal{A}_b)$. But, from (3.7), we know that $\mathbb{P}(\mathcal{A}_b) = \frac{b-1}{b}$. So, we have that

$$\mathbb{P}(\mathcal{A}) \geq \frac{b-1}{b}.$$

Taking the limit as $b \to \infty$, since the right barrier doesn't actually exist, we can find that $\mathbb{P}(\mathcal{A}) \geq 1$. But, by the definition of a probability, we know that $\mathbb{P}(\mathcal{A}) \leq 1$. Since we have found that the probability is bounded both above and below by 1, we can conclude that $\mathbb{P}(\mathcal{A}) = \mathbb{P}(T < \infty) = 1$. □

Since $T$ is always finite, we can then try to calculate $\mathbb{E}(T)$. Similar to the calculation to derive Proposition 3.1, we can use first step analysis (I believe this is also possible through continuous time martingales, but I am not confident enough in my knowledge of the theory to use it).

*Derivation.* Consider, again, the two barrier setup as before with a barrier at 0 and one at $b$, eventually taking the limit as $b \to \infty$. Let $f(x) = \mathbb{E}(T|Z_0 = x)$ with $f(b) = 0$ and $\lim_{x\to 0^+} f(x) = 0$, where we have to use the right limit because $Z$ is not defined for $x \leq 0$.

Using first step analysis on a small timestep of $\Delta t > 0$, we can get that

$$f(x) = \mathbb{E}[f(x + \sqrt{x}B_{\Delta t})] + \Delta t.$$

Then, by Taylor expanding the expectation and dropping $\mathcal{O}(x^3)$ terms, we see that

$$f(x) = f(x) + f'(x)\mathbb{E}(\sqrt{x}B_{\Delta t}) + \frac{1}{2}f''(x)\mathbb{E}[(\sqrt{x}B_{\Delta t})^2].$$

By computing the expectations and cancelling $f(x)$, we get

$$0 = \frac{1}{2}f''(x)x\Delta t + \Delta t \implies -2 = f''(x)x.$$

Solving this ODE, we find that

$$f(x) = -2x\log x + Cx + D.$$

Using the initial conditions, we can find that $D = 0$ and $C = 2\log k$. Thus, we have the final form of $f$:

$$f(x) = -2x\log x + 2\log kx \implies f(1) = 2\log k. \tag{3.9}$$

Taking the limit of $k \to \infty$, we see that this is actually unbounded. So, despite $T$ always being finite, the summation in $\mathbb{E}(T)$ diverges/is infinite!

We can observe this property in the simulations as well, which we will describe now. Like in §2, there exists an Euler-Maruyama discrete approximation that we can use to approximate the solution of (3.1).

$$Z_{k+1} = Z_k + \sqrt{Z_k}\Delta B_k^h. \tag{3.10}$$

We can run this simulation until extinction time. Some realizations for various step sizes are displayed in Figure 2. Additionally, we can observe the diverging expectation through some simple Monte Carlo simulation. The algorithm is essentially on the order of quadratic, so due to time and memory limitations it was only feasible to run the Monte Carlo simulation up to $n = 10^5$ iterations, but it is clearly observed in 3 that the expectation is indeed diverging as the number of iterations we run goes up.

# 4 Matrix Completion

We now turn to an interesting application of the diagonal matrix completion problem. We consider the following stochastic process in $\mathbb{R}^n$.

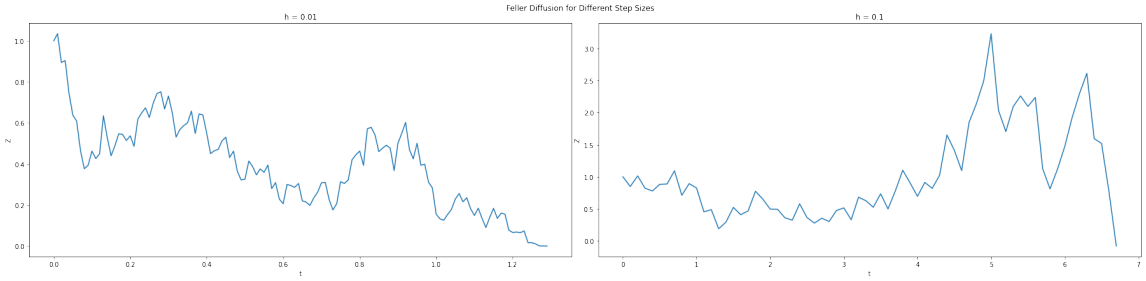



Figure 2: Realizations of Feller diffusion for step sizes $h = 0.01$ and $h = 0.1$.

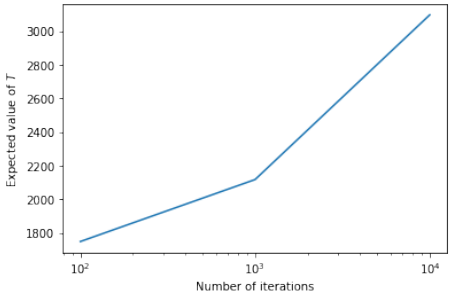



Figure 3: Number of steps until extinction time for Feller diffusion averaged over $n = \{10^3, 10^4, 10^5\}$ iterations.

**Definition 4.** (Matrix Completion Process) The **matrix completion process** is the stochastic process that is the solution to the following Itô SDE:

$$dv = \sqrt{P_\beta(r(t))}dB^{(n)}, \tag{4.1}$$

where $r(t)$ is a vector in which each $r_i^2(t) = 1 - |v_i(t)|^2$ for $1 \leq i \leq n$ called the residual vector and $P_\beta$ is a solution to the matrix completion problem with $r^2(t)$ as the diagonal.

Similarly to the two previous sections, we can use an Euler-Maruyama scheme to discretely approximate the solution to this SDE.

$$v(k+1) = v(k) + \sqrt{P_\beta(r(k))}\Delta\mathbf{B}_k^h, \quad v(0) = \mathbf{0} \in \mathbb{R}^n. \tag{4.2}$$

To see how to implement this properly and avoid difficult computations (square rooting a matrix), we notice that the kicks in noise will be another multivariate Gaussian. To see what the sampling covariance matrix is, we can use that

$$\mathbf{var}(AX) = A\mathbf{var}(X)A^T. \tag{4.3}$$

In our case, $A = \sqrt{P_\beta}$, and $\mathbf{var}(X) = \mathbf{var}(dB^{(n)})$. Since we can approximate $dB^{(n)}$ with a vector of $(\Delta B_{k1}^h, \Delta B_{k2}^h, \ldots, \Delta B_{kn}^h)$, we know that the covariance matrix is a diagonal matrix with $h$s along

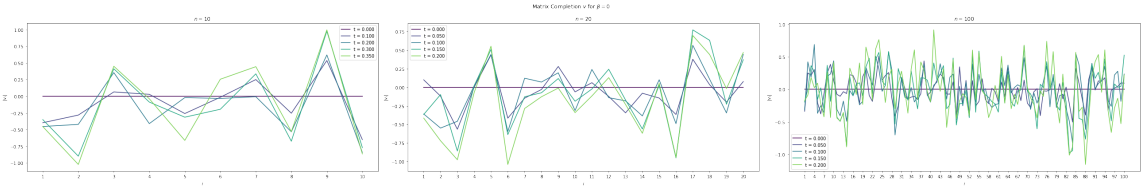


Figure 4: Realizations of the matrix completion stochastic process with $\beta = 0$.

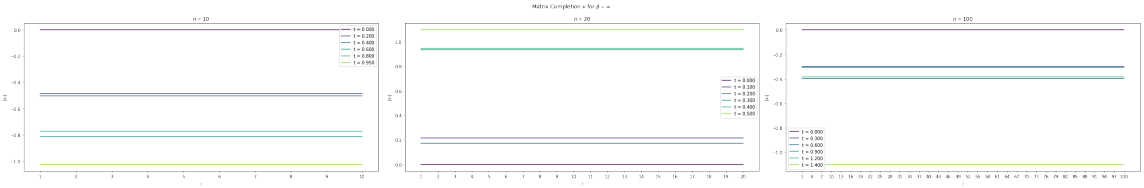


Figure 5: Realizations of the matrix completion stochastic process with $\beta = \infty$.

the diagonal and 0s everywhere else. So, (4.3) in this case evaluates to

$$\mathbf{var}\left(\sqrt{P_\beta}dB^{(n)}\right) = \sqrt{P_\beta}\mathrm{diag}(h)\sqrt{P_\beta}^T = hP_\beta. \tag{4.4}$$

So in practice, at every iteration, we sample a vector from a multivariate Gaussian with mean $\mathbf{0} \in \mathbb{R}^n$ and covariance kernel $hP_\beta$ and add it to the current value of $v$.

We will use the simplest $P_\beta$s, namely when $\beta = 0$ and $\beta = \infty$.

$$P_0 = \mathrm{diag}(r_1^2, r_2^2, \ldots, r_n^2) \text{ and } P_\infty = rr^T. \tag{4.5}$$

When simulating the process, we use the stopping time

$$T = \min\{k > 0 : \exists 1 \leq i \leq n \text{ where } |v_i(k)| \geq 1\}. \tag{4.6}$$

For $\beta = 0$, we see the diffusion occuring in Figure 4). The peaks slowly spread away from 0 in each index until there is one index with magnitude greater than 1.

For $\beta = \infty$, we see that because the residual $r_i = 1$ for all $i = 1, \ldots, n$ in the beginning, the covariance matrix is a rank-one matrix with all identical elements (all 1s). A covariance matrix with all equivalent elements defines a degenerate process, as all the elements of the sampled vector will necessarily be equal. So, each index $i$ gets updated with the same kick in noise. However, this degenerate process self-propagates itself. Since each element in the initial condition (which are all equal) get updated with the same kick in noise, the new vector also has a constant residual vector, which results in another degenerate covariance kernel.

So, we expect a degenerate process where all indices are equal for all $t \geq 0$ (should be straight horizontal lines on the plot). This is observed in Figure 5.

Future work should investigate how to analytically analyze the stopping time presented in (4.6). It will probably just be a lot of annoying vector manipulations.

One gripe with this SDE is that we would like more control over when the diffusion stops, as one index in $v$ leaving $[-1, 1]$ is very strict - just because one index is outside of the domain does not imply anything about the other indices. This is what the SDE in the next section addresses.

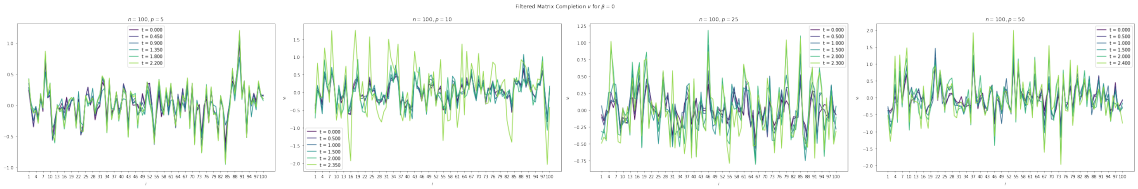


Figure 6: Realizations of the filtered matrix completion process for various ranks of $L(t)$ when $\beta = 0$ and $p = \{5, 10, 25, 50\}$.

# 5 Filtered Matrix Completion

The final SDE we will investigate is a "filtered" extension of the previous section's SDE. This SDE attempts to drive all residuals to 0 at the same time, compared to the SDE presented in §4.

We can consider a sequence of "filters" that satisfy

$$\dot{L} = LP_\beta L, \quad P_\beta = P_\beta(r) \text{ for } r_i^2(L) = 1 - L_{ii} \text{ and } 1 \leq i \leq n. \tag{5.1}$$

Again here, we consider only the two simplified values of $P_\beta$ presented in (4.5). We consider two hyperparameters $n$, the dimension of the system, and $p$, the rank of $L(t)$, our covariance kernel at the timestep $t$. With this, we are now ready to define the filtered process.

**Definition 5.** (Filtered Matrix Completion Process) The **Filtered Matrix Completion Process** is the solution $v$ to the following Itô SDE:

$$dv = L\sqrt{P_\beta(r(L))}dB^{(n)}. \tag{5.2}$$

We choose to stop this scheme whenever any of the diagonal elements on $L_{ii}$ exceed 1 or that $r_i^2(L) = 1 - L_{ii} \leq 0$. Like before, meaningful future work would be to study analytical computations of the expectation of this stopping time.

To get the initial value $L_0$, we can draw from the Wishart random matrix ensemble. Recall that this involves generating an $n$ by $p$ matrix $X$ whose elements are drawn from iid standard Gaussians, then computing the positive semi-definite square matrix $XX^T$. We don't want our initial point to be too far away from $\mathbf{0} \in \mathbb{R}^n$, so we normalize $XX^T$ such that the maximum singular value is 0.5 - we will let this be $L_0$. $v_0$, the initial value vector, is drawn from a multivariate Gaussian with covariance kernel $L_0$.

We can discretize (5.1) with the following approximation:

$$L_{k+1} = L_k + h(L_k P_\beta(r(L_k))L_k). \tag{5.3}$$

Of course, we still need to define the updates to $v$ at each timestep $k$, or discretizing (5.2). We do this through sampling a random multivariate Gaussian vector $\psi_k$ such that the covariance kernel is $P_\beta(r(L_k))$. Then, we update $v$ with $\psi_k$ through the following relationship.

$$v_{k+1} = v_k + L_k\psi_k\sqrt{h}. \tag{5.4}$$

In Figure 6 and 7, we can much more clearly observe the values diffusing toward 1 and beyond more uniformly compared to as observed in Figure 4 and 5. By maintaining this set of filters, we are able to better regulate the diffusion compared to before - instead of the process becoming undefined when one index exits the region $[-1, 1]$, we make all the residual indices go to 0 together.

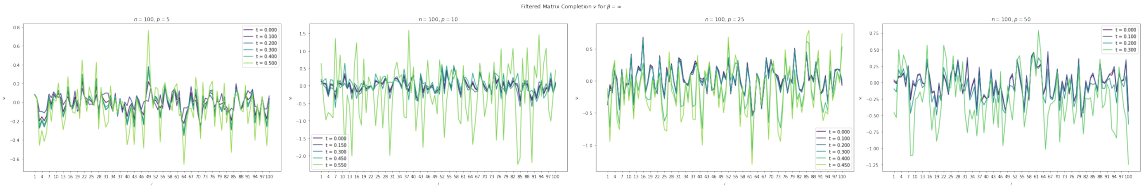


Figure 7: Realizations of the filtered matrix completion process for various ranks of $L(t)$ when $\beta = \infty$ and $p = \{5, 10, 25, 50\}$.

# 6 Dimensionality Reduction Visualization for §4 and §5

Because the last two processes live in $n$ dimensions, it is interesting to see how various dimensionality reduction techniques can change how we visualize these processes and if they can reveal any underlying geometrical structure in how the process is diffusing in $\mathbb{R}^n$. Thus, in this section, we will visualize the discrete approximations from §4 and §5 using dimensionality reduction techniques.

## 6.1 Embedding Techniques

We will explore 2 forms of embedding into lower dimensions: t-stochastic neighbor embedding (tSNE) and principal component analysis (PCA). Future work can explore more, but we choose these two because they are very common and that one is stochastic while the other is deterministic.

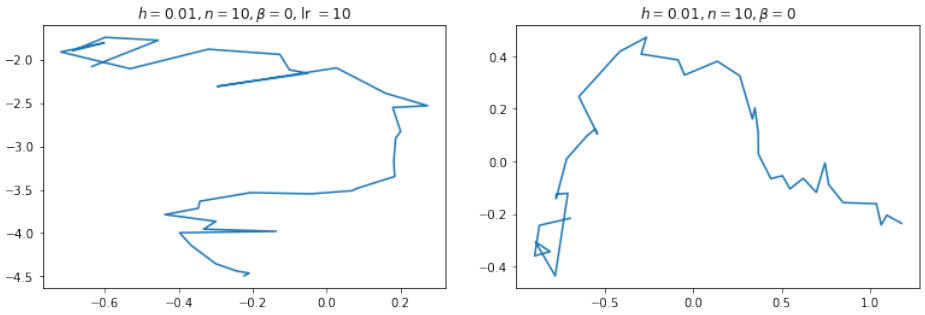
### 6.1.1 tSNE

Stochastic Nash Evolution is not the only SNE in town - we can use stochastic neighbor embedding (SNE) to project the process down to lower dimensions for visualization. Briefly put, the most basic form of SNE attempts to minimize the KL divergence through gradient descent between the distribution of lower dimensional points and the distribution of the original high dimensional data. These distributions are based on Gaussians centered at each point and use the Euclidean distance to neighboring points - hence the name stochastic neighbor embedding.

tSNE is a variant that optimizes this procedure by introducing a simpler gradient that exploits symmetry and basing the lower dimensional space on the student-t distribution rather than a Gaussian [2]. Altogether, this methodology often allows tSNE to preserve and reveal geometrical structure in higher dimensions that we cannot visualize normally, making it an interesting choice to visualize [1].

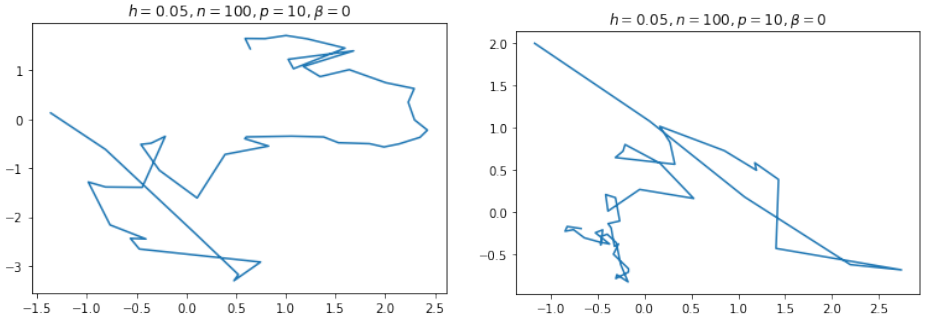
### 6.1.2 PCA

Principal Component Analysis (PCA), on the other hand, is the kingpin of dimensionality reduction algorithms and is a deterministic algorithm. It gathers $p$ component lines which minimize the squared residuals to the data points (best fit lines) which are all orthogonal to each other. It then performs a change of basis using these vectors as an orthonormal basis in $p$ dimensions. Because of PCA's simplicity, we can use it as a strong benchmark to compare tSNE to.




(a) tSNE (b) PCA

Figure 8: Visualizations of the same realization of the matrix completion process in 2 dimensions done by (a) tSNE and (b) PCA.




(a) tSNE (b) PCA

Figure 9: Visualizations of the same realization of the filtered matrix completion process in 2 dimensions done by (a) tSNE and (b) PCA.

## 6.2 Visualizations

Because tSNE is a stochastic algorithm, it can become unstable for higher dimensions, so we choose to be careful and slightly on the lower side with the values of $n$ that we use for both processes and the $p$ we use in the filtered case. Figures 8a and 8b display a realization of the matrix completion problem reduced into 2 dimensions from $n = 10$ by tSNE and PCA, respectively. The geometrical structure of some sort of loop is preserved, though seemingly rotated on an axis. Figures 9a and 9b display the same for $n = 100$ and $p = 10$ filtered process. Again, we see similar geometrical structures occurring, like the sudden offshoot into the upper left being displayed in both.

# 7 References


[1] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[2] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

# 8 Appendix

## 8.1 Code

All code used in this project is in this Github repository.