

12 Stochastic Differential Equations

Lab Objective: *Stochastic differential equations are used to model stochastic processes. In this lab we will explore Brownian motion and then derive the Euler-Maruyama numerical method for SDEs. We will build an Euler-Maruyama numerical solver and use this solver to predict future stock prices.*

Stochastic differential equations combine the concepts of Brownian motion and differential equations in order to model stochastic processes. A stochastic process is a mathematical object made from a family of random variables. These processes model events that occur with random changes over time, such as bacterial population growth, movement of gas molecules, and fluctuating stock prices. To understand stochastic processes, it is imperative to understand Brownian motion.

Brownian Motion

Brownian motion is random movement described by the Wiener process. The Wiener process is a stochastic process $W(t) \sim \mathcal{N}(0, t)$ which satisfies the following conditions:

1. $W(0) = 0$,
2. $W(t - s) = W(t) - W(s)$,
3. $W(t)$ is independent for all t .

For example, imagine a point at zero on a number line. The point can only move one number away and must move left or right. The probability of moving left and right is equal and can be modeled by a coin flip. Let's say that landing on heads moves the point right and tails moves the point left. On our first flip, we get heads and the point moves from 0 to 1. Now we flip the coin again. This coin flip does not depend on the previous coin flip and determines whether the point moves back to 0 or moves to 2. On the second flip, we get tails and the point moves back to 0. Continuing this process shows the random movement of the point on the number line.

As we flip the coin t times, the distribution of the point's position is approximately normally distributed with mean 0 and variance t . Note that $dW \sim \mathcal{N}(0, \Delta t)$ is also approximately normally

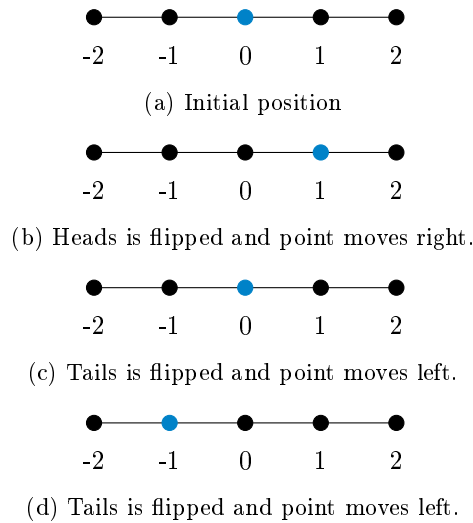


Figure 12.1: Example of Brownian motion. The blue dot travels left or right with equal probability. It is modeled by a coin flip. When heads, the dot moves right. When tails, the dot moves left.

distributed because $W(t-s) = W(t) - W(s)$ and $W(t-s) \sim \mathcal{N}(0, t-s)$. Using dW , we can show the movement of a point with the differential equation

$$\frac{dS_t}{S_t} = g(t, S_t)dW. \quad (12.1)$$

where $g(t, S_t)$ is a scalar function and S_t is the position of the point at time t . We can manipulate Equation 12.1 to model the position of the point numerically:

$$\begin{aligned} dS_t &\approx g(t, S_t)S_t dW \\ S_{t+1} - S_t &\approx g(t, S_t)S_t dW \\ S_{t+1} &\approx S_t + g(t, S_t)S_t dW \end{aligned}$$

where $dW \sim \mathcal{N}(0, 1)$. This model is itself a stochastic differential equation that is based completely on brownian motion.

Problem 1. Write a function `brownian_motion()` that accepts a scalar function g , initial condition y_0 , and an array of time points \mathbf{t} . The function should return an array of the solution to Equation 12.1 evaluated at t .

Animate this function for $g(t, S_t) = 1$, $y_0 = (1, 1)$ and $t \in [0, 100]$. The animation should show a particle moving with a tail indicating it's previous position.

(Hint: Because $y_0 \in \mathbb{R}^2$, $dW \in \mathbb{R}^2 \times \mathbb{R}^2$ so that each dimension of y_0 moves independently.

Thus, $dW = \begin{pmatrix} d_{w_1} & 0 \\ 0 & d_{w_2} \end{pmatrix}$ where $d_{w_i} \sim \mathcal{N}(0, 1)$).

Euler-Maruyama Method

A stochastic differential equation (SDE) is a differential equation that involves at least one stochastic component. Equation 12.1 is an example of a SDE, as its only component is stochastic. However, most SDEs are not made of just one stochastic component. A commonly used stochastic differential equation which contains a non-stochastic component is

$$\frac{dS_t}{S_t} = f(t, S_t)dt + g(t, S_t)dW \quad (12.2)$$

where $f(t, S_t)$ and $g(t, S_t)$ are scalar functions and $dW \sim \mathcal{N}(0, \Delta t)$. In this equation, the first term is a standard differential equation while the second term represents brownian motion. The combination allows for more accurate models of stochastic processes. To solve Equation 12.2, we apply the Euler-Maruyama method. This method combines the Euler method for ODEs with an additional stochastic component. Note that Problem 1 is the Euler-Maruyama method where $f(t, S_t) = 0$. We solve for S_{t+1} as follows:

$$\begin{aligned} dS_t &\approx f(t, S_t)S_t dt + g(t, S_t)S_t dW \\ S_{t+1} - S_t &\approx f(t, S_t)S_t dt + g(t, S_t)S_t dW \\ S_{t+1} &\approx S_t + f(t, S_t)S_t dt + g(t, S_t)S_t dW. \end{aligned}$$

Problem 2. Write a function `euler_maruyama()` which accepts a scalar function `f`, a scalar function `g`, initial condition `y0s`, and an array of time points `t`. The function should return an array of the solution to 12.2.

To test your function, set $f(t, S_t) = 1 - (S_t)^2$, $g(t, S_t) = 0.1$, $y_0 = 1$ and $t \in [0, 10)$. Your function should result in a plot which randomly oscillates with y generally in the interval $[0, 2]$. An example is shown in Figure 12.2.

Drift and Volatility

SDEs are often used in mathematical finance models. Particularly, the Geometric Brownian Motion (GBM) model is useful in predicting future stock prices. The GBM is defined as follows:

$$dS_t = \mu S_t dt + \sigma S_t dW, \quad (12.3)$$

where μ is the drift of the stock and σ is the volatility. The drift μ of a stock is the average change in return of historic stock data. The volatility of a stock is the standard deviation of the change in return of historic stock data. It is expected that μ and σ will remain the same for future stock data.

To find μ and σ , let $\theta = (\mu, \sigma)$. We want to find θ_{MAP} (maximum a posteriori) which maximizes the probability that μ and σ fit the historical data. We can calculate θ_{MAP} using Bayes Theorem:

$$\theta_{MAP} \left(\frac{dS}{S} \right) = \operatorname{argmax}_{\theta} P \left(\theta \mid \frac{dS}{S} \right) = \operatorname{argmax}_{\theta} \frac{P \left(\frac{dS}{S} \mid \theta \right) P(\theta)}{\int_{\Theta} P \left(\frac{dS}{S} \mid \vartheta \right) P(\vartheta) d\vartheta} = \operatorname{argmax}_{\theta} P \left(\frac{dS}{S} \mid \theta \right) P(\theta), \quad (12.4)$$

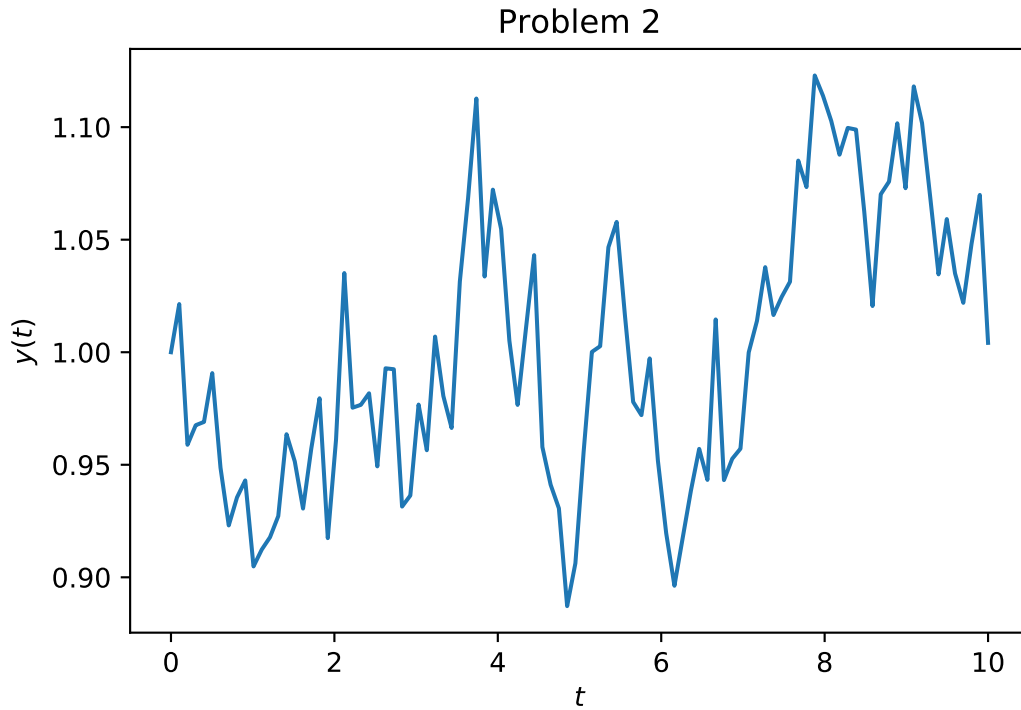


Figure 12.2: Possible solution for $f(t, S_t) = 1 - (S_t)^2$, $g(t, S_t) = 0.1$, $y_0 = 1$ and $t \in [0, 10]$.

where Θ is the collection of all possible θ . Since no information is known about $P(\theta)$, we choose $P(\theta)$ to be equal over all θ . This avoids giving bias to one value of θ over another. To ease calculations, we choose our improper prior to be uniformly 1, $P(\theta) = 1$. This implies $\theta_{MAP} = \operatorname{argmax}_{\theta} P(\frac{dS}{S} | \theta) = \theta_{MLE}$.

Now it is necessary to determine the probability distribution of $P(\frac{dS}{S} | \theta)$. Plotting the change in stock price results in the following histogram: The change in stock price is approximately normally distributed and thus we consider $P(\frac{dS}{S}) \sim \mathcal{N}(\mu, \sigma^2)$. Now we can calculate θ_{MAP} as follows:

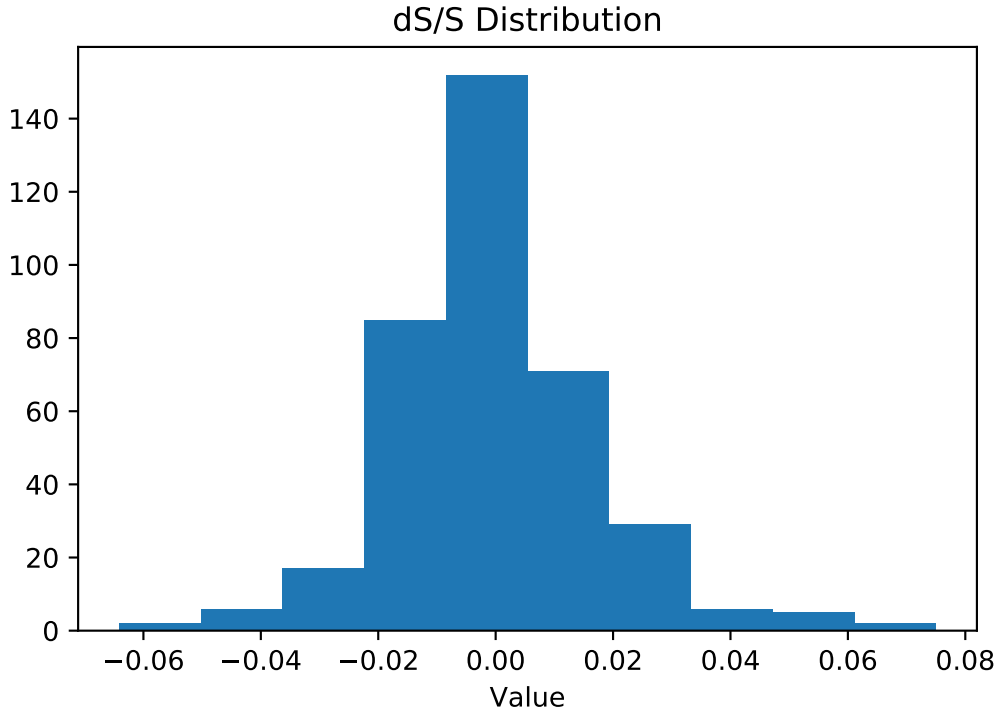


Figure 12.3: Distribution of change in stock price.

$$\begin{aligned}
\theta_{MAP} &= \operatorname{argmax}_{\theta} P\left(\frac{dS}{S} | \theta\right) \\
&= \operatorname{argmax}_{\theta} \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2}\right) \\
&= \operatorname{argmax}_{\theta} \log\left(\prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2}\right)\right) \\
&= \operatorname{argmax}_{\theta} \sum_{i=1}^N \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2}\right)\right) \\
&= \operatorname{argmax}_{\theta} \sum_{i=1}^N \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2} \\
&= \operatorname{argmax}_{\theta} N \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \sum_{i=1}^N \frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2} \\
&= \operatorname{argmin}_{\theta} N \log(\sqrt{2\pi\sigma^2}) + \sum_{i=1}^N \frac{\left(\left(\frac{dS}{S}\right)_i - \mu\right)^2}{2\sigma^2} \\
&= \operatorname{argmin}_{\theta} N \log(\sqrt{2\pi\sigma^2}) + \frac{1}{2\sigma^2} \sum_{i=1}^N \left(\left(\frac{dS}{S}\right)_i - \mu\right)^2
\end{aligned} \tag{12.5}$$

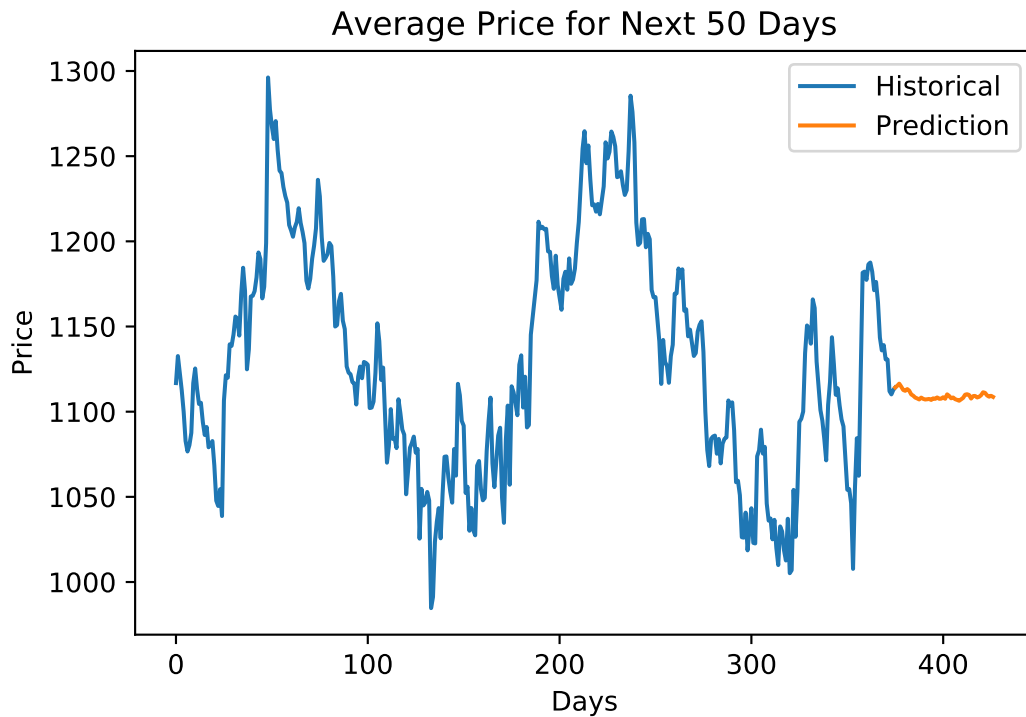


Figure 12.4: Possible plot of future Google stock prices.

The end result of Equation 12.5 gives an equation that can be optimized using `scipy.optimize.minimize` to find θ .

Problem 3. Write a function `theta()` which takes in an array of historical data. Use `scipy.optimize.minimize` and Equation 12.5 to calculate the optimal μ and σ . Return μ and σ . For the closing stock prices of `google_stock.csv`, $\mu \approx 1129.4321$ and $\sigma \approx 1.8548$.

(Hint: Use the sample mean and sample variance as an initial guess for `scipy.optimize.minimize`. Also use `method='Nelder-Mead'`.)

Problem 4. Use `euler_maruyama()` to predict the future closing stock prices of Google stock for $t \in [377, 427)$. Plot the original data and the average predicted stock prices. Return an array of the average future stock prices. Your plot should look similar to Figure 12.4.

(Hint: Let $f(t, S_t) = \mu$ and $g(t, S_t) = \sigma$. Each t represents one day, so there should be 50 predicted values.)

Convergence of Euler-Maruyama

The Euler-Maruyama method has strong convergence as $\Delta t \rightarrow \infty$. This can be seen heuristically by looking at the expected value of the error of the numerical method. Note equation 12.3 can be solved analytically to get the solution

$$S(t) = s_0 e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)} \quad (12.6)$$

where $s_0 = S(0)$. Let $A_{\Delta t}(t)$ be the approximation of $S(t)$ with step size Δt . Because we are working with SDEs, we take the expected value of the max error over all t and define the error as

$$E(A_{\Delta t}) = \sup_t \mathbb{E}(|S(t) - A_{\Delta t}(t)|) \approx C(\Delta t)^\gamma. \quad (12.7)$$

where γ is the order of convergence and C is some constant. Taking the log of both sides, we get

$$\log(E(A_{\Delta t})) = \log(C) + \gamma \log(\Delta t). \quad (12.8)$$

To find the order of convergence γ , we can plot $\Delta t \times E(A_{\Delta t}(t))$ as $\Delta t \rightarrow \infty$ with log axes. This should result in a straight line with slope γ , giving us the convergence rate. For Euler-Maruyama, $\gamma = \frac{1}{2}$.

Problem 5. Write a function `convergence()` that calculates the convergence of Euler-Maruyama. Calculate the drift and volatility of the closing prices in `google_stock.csv` and use Euler-Maruyama to predict stock values for $t \in [0, 50)$. Calculate the difference between the analytical solution 12.6 and predicted data from Euler-Maruyama. Plot the difference for $dt = 2^i$ for $i \in [0, 1, 2, 3, 4, 5]$ on a log-log plot. Your plot should be a straight line with slope $\frac{1}{2}$.

(Hint: Try using `np.arange`).