The Wiener process and the GBM, Derivations and simulations

Giacomo Babudri

January 2024

1 Wiener process

The Wiener process, often referred to as Brownian motion, is a continuous-time stochastic process that models the random and continuous movement of a particle. The Wiener process $W(t)$ is a stochastic process defined for $t \geq 0$, where t

1. **Independent Increments:** For any sequence of time points $0 \leq t_1 < t_2 < \ldots < t_n$, the increments $W(t_2) - W(t_1), W(t_3) - W(t_2), \ldots, W(t_n) - W(t_{n-1})$ are independent random variables.
2. **Stationary Increments:** The distribution of $W(t_2) - W(t_1)$ depends only on the length $t_2 - t_1$ and not on the specific values of t_1 and t_2.
3. **Gaussian Distribution:** The increments $W(t_2) - W(t_1)$ are normally distributed with mean zero and variance $t_2 - t_1$.
4. **Continuous Paths:** The paths of the Wiener process are almost surely continuous. In other words, the process has no jumps, and it exhibits a form of pathwise continuity.
5. **Memoryless Property:** The future values of the Wiener process are not dependent on its past values. It is a Markov process, meaning that the current state contains all the information needed to predict future behavior.

1.1 Simulation

An example of Wiener process simulation, was realized through the use of python, as it's possible to see below

1.1.1 CODE:

```
import numpy as np
import matplotlib.pyplot as plt

def simulate_wiener_process(T, num_steps):
    # T: Total time
    # num_steps: Number of time steps

    # Time step size
    dt = T / num_steps

    # Generate random increments from a normal distribution
    increments = np.random.normal(0, np.sqrt(dt), num_steps)

    # Cumulative sum to get the Wiener process
    wiener_process = np.cumsum(increments)

    # Time array
    time = np.linspace(0, T, num_steps)  # Corrected to exclude the endpoint

    return time, wiener_process

# Parameters
total_time = 1.0  # Total time
num_time_steps = 500  # Number of time steps

# Simulate Wiener process
time, wiener_process = simulate_wiener_process(total_time, num_time_steps)

# Plot the Wiener process
plt.plot(time, wiener_process, label='Wiener Process')
plt.title('Simulation of Wiener Process')
plt.xlabel('Time')
plt.ylabel('Value')
plt.legend()
plt.show()
```

1.1.2 Output:

2 Geometric Brownian Motion

Geometric Brownian Motion (GBM) is a continuous stochastic process commonly employed to model the dynamics of financial instruments such as stock prices. Rooted in the foundational concept of Wiener process, GBM introduces additional components to capture key features observed in financial markets.

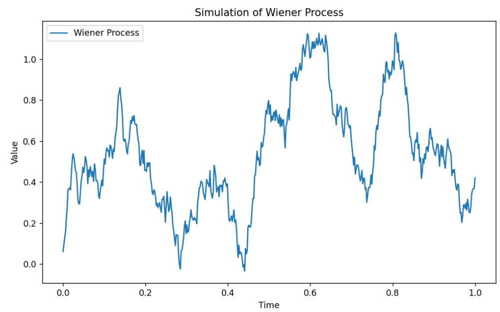

Figure 1: Simulation's output

1. **Stochastic Differential Equation (SDE):**

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t)$$

   where: μ is the average growth rate (drift), σ is the volatility, and $W(t)$ is a Wiener process (Brownian motion), representing the stochastic component.

2. **Key Properties:**
   - GBM integrates the Brownian motion as a core component, reflecting the intrinsic randomness present in the modeled phenomena.
   - The drift term introduces a deterministic trend or decay, capturing the average behavior of the process over time.
   - The volatility factor scales the Wiener process, influencing the amplitude of random variations.
   - GBM exhibits log-normal increments, resulting in log-normal distribution characteristics.
   - With its broad applicability, GBM serves as a foundational concept in stochastic modeling, finding applications in physics, biology, and finance.

3. **Applications:**
   - GBM is widely used in financial models to describe the evolution of prices of financial instruments such as stocks;
   - In physics GBM can be employed to model the diffusion of particles in a fluid.
   - In biology, GBM can be applied to model population dynamics or the growth of biological entities

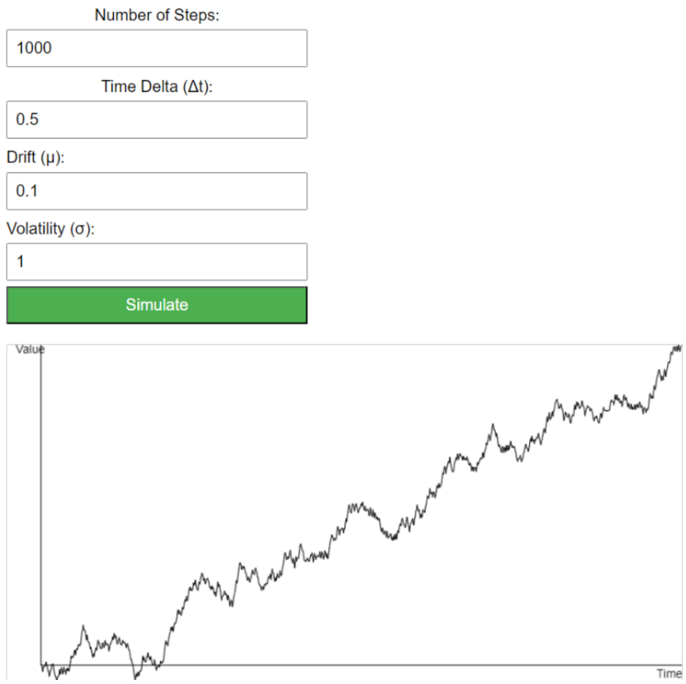

Figure 2: Simulation's output

2.1 Simulation

In the course was asked to execute a simulaiton for a geometric brownian process.

2.1.1 Output