Paul Galli

Fractional Brownian Motion

Introduction

The Fractional Brownian Motion (FBM) is the generalization of the Brownian Motion. It generalizes Brownian Motion by introducing long-range dependence. FBM is a centered gaussian process with stationary increments. However, these increments are not independent. FBM follows a self-similarity process. A FBM process can differentiate depending on the Hurst (H). The H parameter controls the smoothness and memory of the process. Hurst is shown as $0 < H < 1$. When $H = 0.5$ then it is considered a Classical Brownian Motion. If $H < 0.5$ then the process is anti-persistent. This means there is a negative autocorrelation of increments. An example would be if there is an increase then most likely the next step would be a decrease. If $H > 0.5$ then the process is persistent with a positive autocorrelation, meaning if there is an increase then most likely the next step would also be an increase. The higher H is the more smooth the process will be and the lower H is the rougher it will be. FBM shows the process of change/trend over time. In order to measure the height or size of each step in a FBM process we use the Fractional Gaussian Noise (FGN). The FGN follows a normal distribution with stationary increments. The Hurst (H) follows the same rules for the FBM. These processes are used to help simulate the way data could behave when there is an insufficient amount of data available. It can be used across multiple types of fields.

Methods

In order to demonstrate multiple Fractional Brownian Motion and Fractional Gaussian Noise processes the Python libraries fbm, matplotlib, and numpy were used. The library fbm is used to create the values needed. The function FBM takes four parameters: n that is the number of steps which will be 1000, Hurst (H) values 0.2, 0.4, 0.6, and 0.8 to control the smoothness, length to determine the size of the steps which will be 1, and method which affects the noise. There are three different methods used for this function: “hosking” is a recursive method that generates a few steps and then builds upon those steps one-by-one, “cholesky” which builds the full covariance matrix over all points then uses cholesky decomposition to generate a multivariate Gaussian vector with the correct correlations, “daviesharte” is a fast spectral method that uses the Fast Fourier Transform. Each method has its pros and cons. The hosking method is stable, accurate, and requires less memory since it is computing each value one-by-one. However, it can be slow when processing a large number of steps (n) since it builds sequentially. Cholesky is very accurate but uses a ton of memory especially for a larger number of steps since it builds and decomposes a matrix. The daviesharte method is the fastest and most memory efficient even when using a large number of steps. The only downside is that the numerical round-offs can have errors when H is very close to 0 or 1. The matplotlib and numpy libraries are used to create and display each graph for the various FBM and FGN results.

Results

Between three different methods, four different H values, creating both a fractional brownian motion and fractional gaussian noise, twenty-four graphs were created. Each graph visually represents each different type of process using the various parameters. Figure 1 and Figure 2 display the results using H = 0.2 and the method is Hosking.

Figure 1.

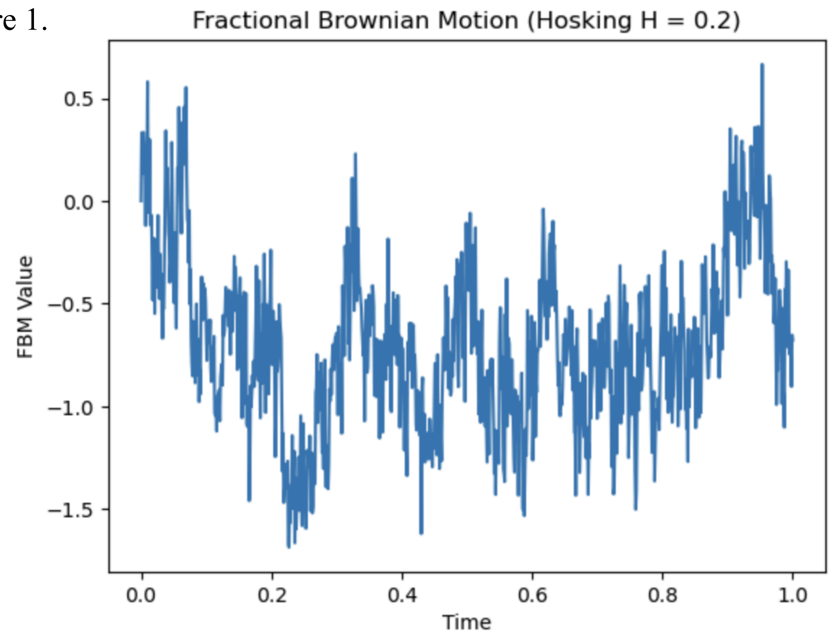
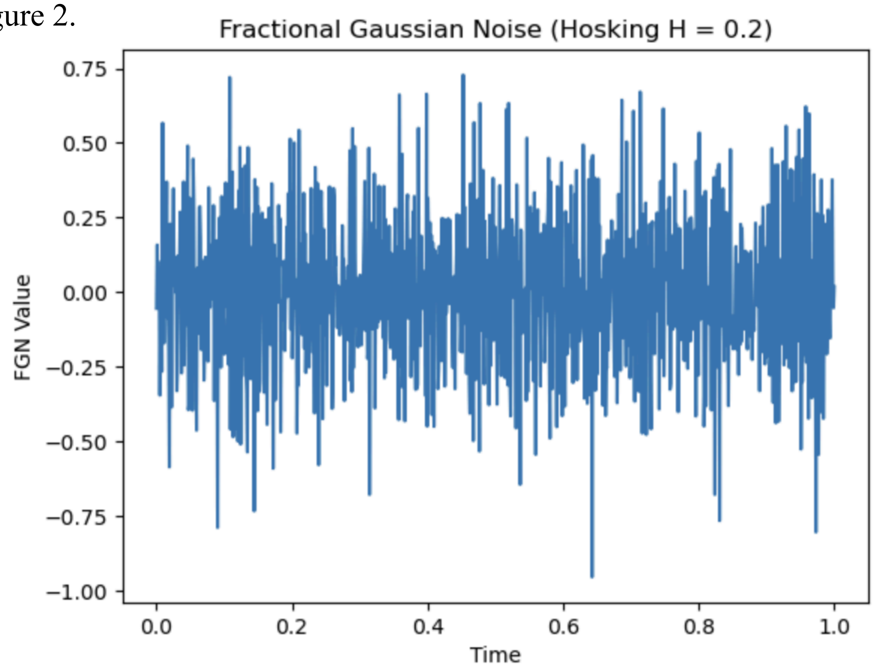


Figure 2.



Since H = 0.2 we can see how rough the process is. The process is anti-persistent because of the low H value. The hosking method builds the process value by value and since H = 0.2 each value appears to be constantly moving in different directions. There appears to be very few trends as the values keep flipping. Next, we have the same method but with a higher H value. Figure 3 and Figure 4 display the process using the hosking method with H = 0.6.

Figure 3.

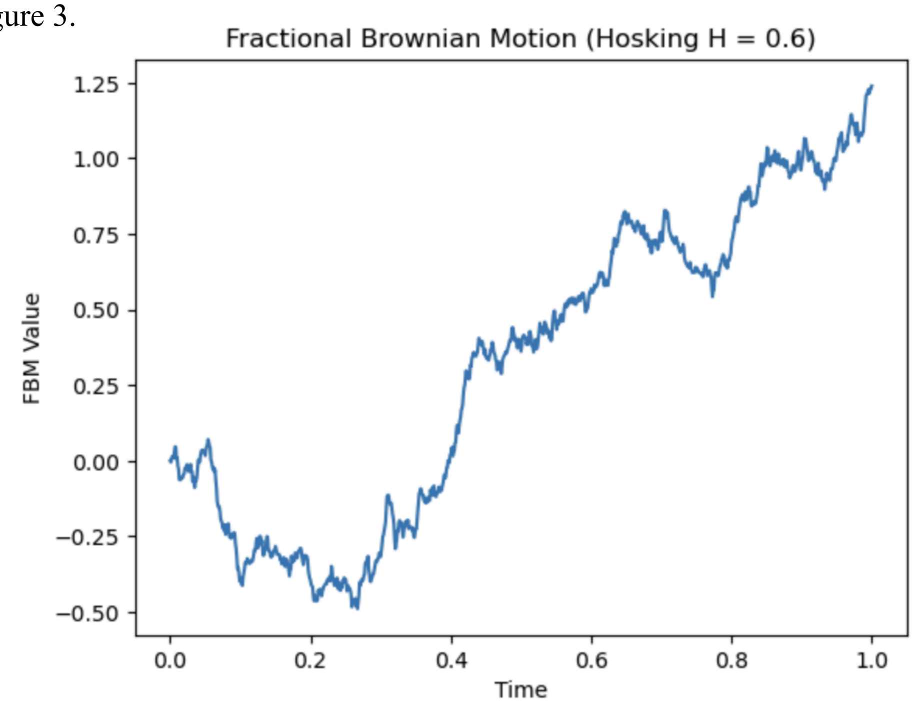
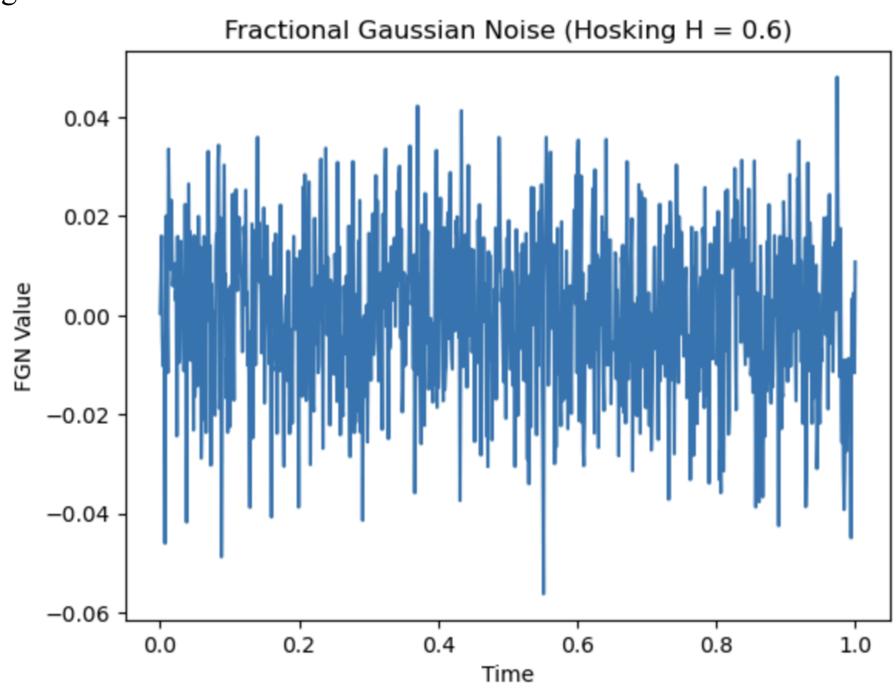


Figure 4.



When H > 0.5 there tends to be more persistence. This can be seen with H = 0.6. There appears to be a trend when the values decrease they keep decreasing and then when they start to increase they keep increasing. The process is much smoother than H = 0.2. This is because the values are more persistent.

While the hosking method is stable it sometimes isn't very accurate. To look at a more accurate process we have the cholesky method. Figure 5 and Figure 6 display a FBM and FGN process using the cholesky method with H = 0.4.

Figure 5.

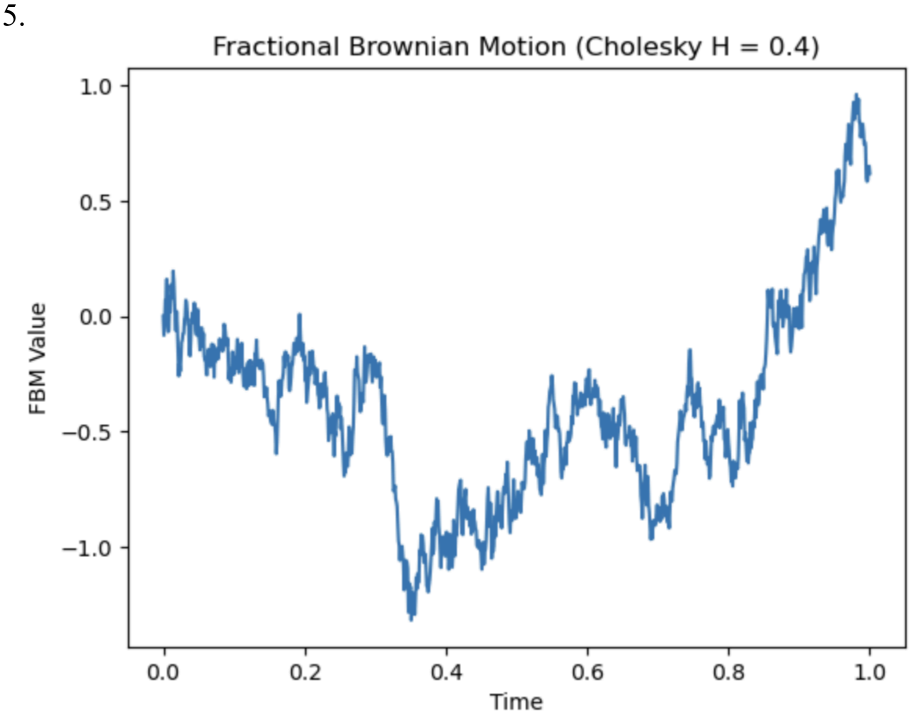
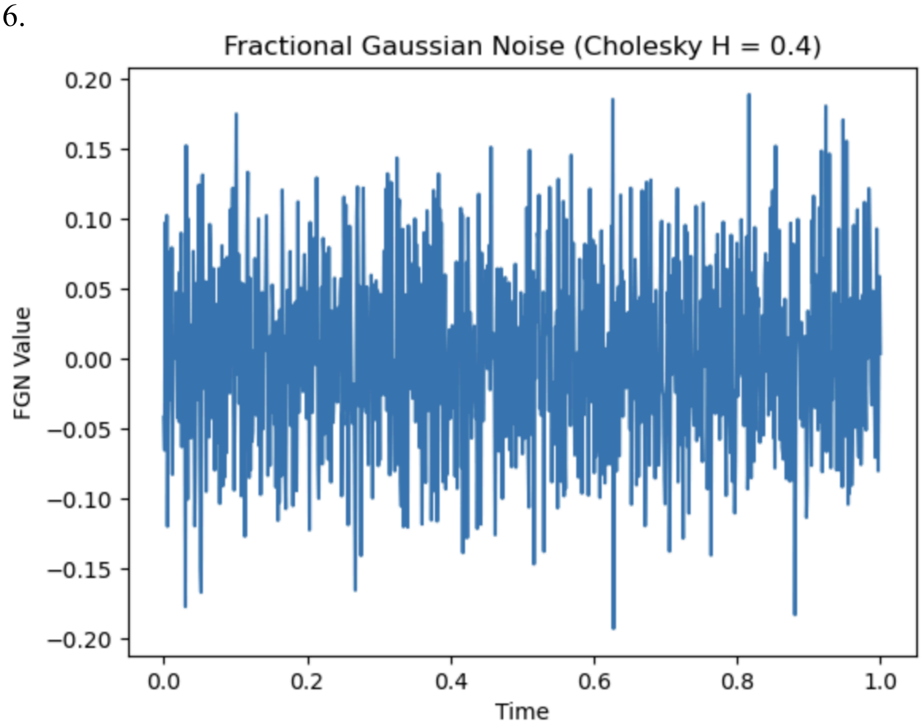


Figure 6.



Since H < 0.5 we can still see some roughness. However, there is more trend because of the slightly higher H value. As some values decrease the next value follows and as it increases so does the next value. There is a short downward trend and then an upward trend. In Figure 7 and Figure 8 we see how persistent the process is because of a high H value.

Figure 7.

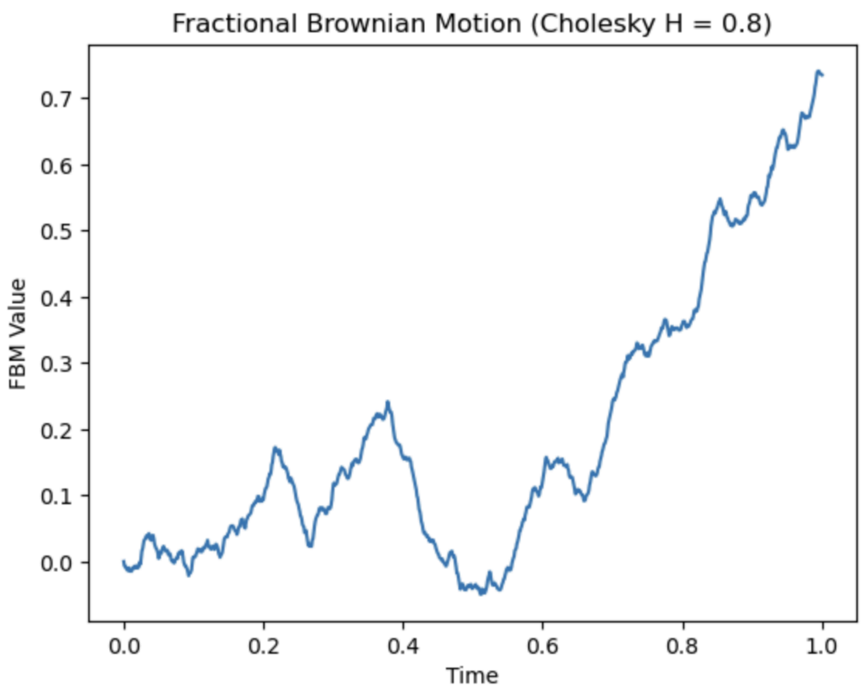
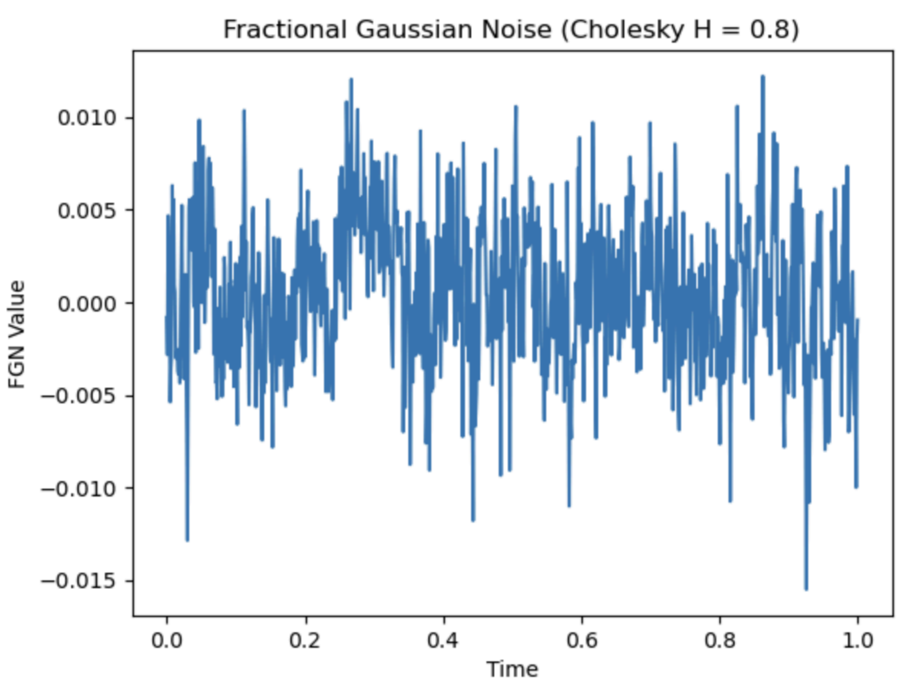


Figure 8.



Since H is high, the process has a positive memory. The noise tends to move in the same direction for a while. There are longer trends as there is a long increasing trend. The final method is considered to be the best overall method. The Davies-Harte method is the fastest and most memory efficient. However, when H is closer to 0 or 1 the numerical values may not be as accurate. The processes in Figure 9 and Figure 10 are created using the Davies-Harte method.

Figure 9.

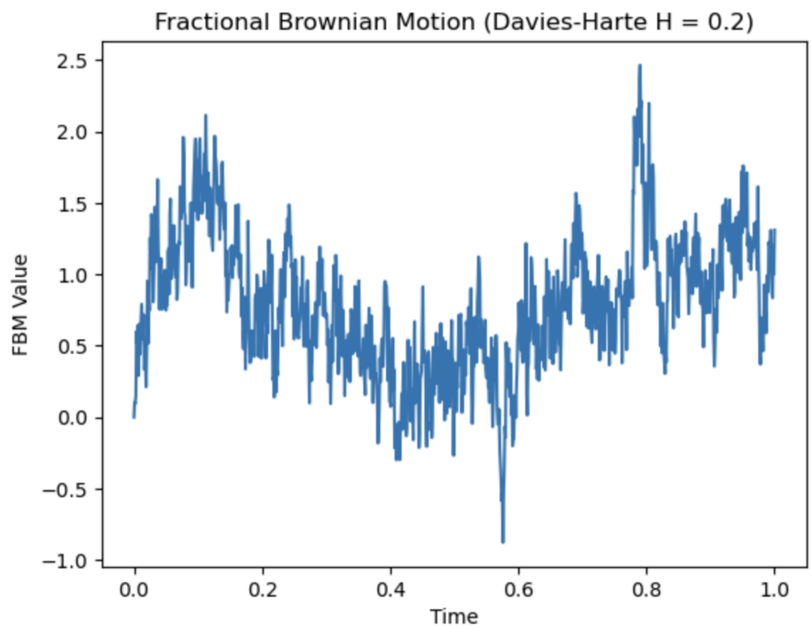
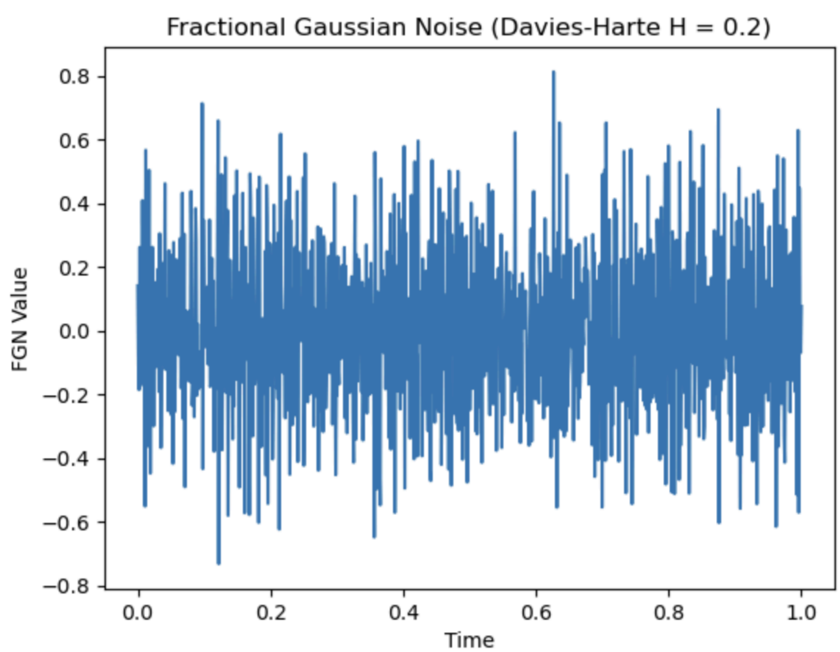


Figure 10.



The process is similar to that of the Hosking method with H = 0.2. This process is computed much quicker and uses less memory. The process is rough and shows how the values quickly change directions. The Davies-Harte method is prone to errors in numerical round offs when H is close to 0 or 1. Ideally, the best value to use is around 0.5. In Figure 11 and Figure 12 both processes are created using a Hurst = 0.6. Since the value is greater than 0.5 it will be more persistent with a smoother process.

Figure 11.

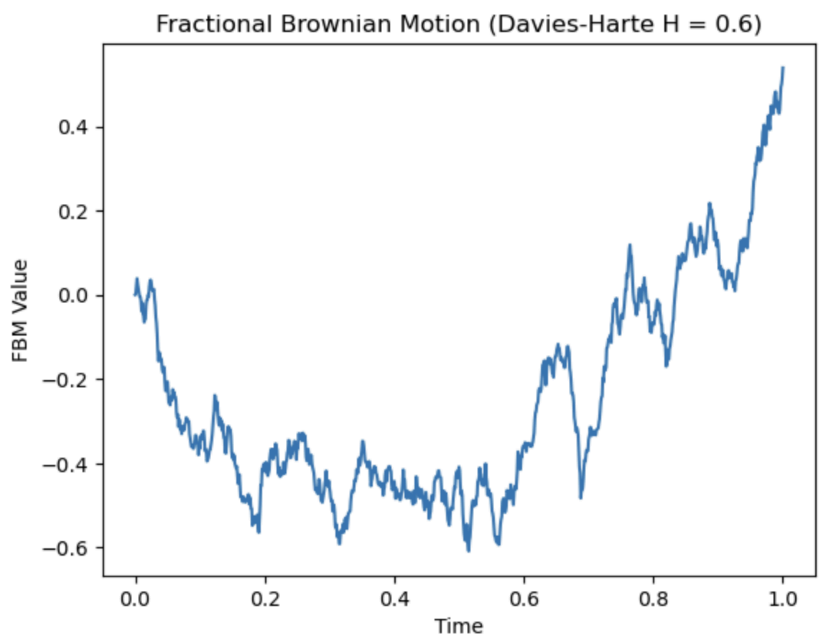
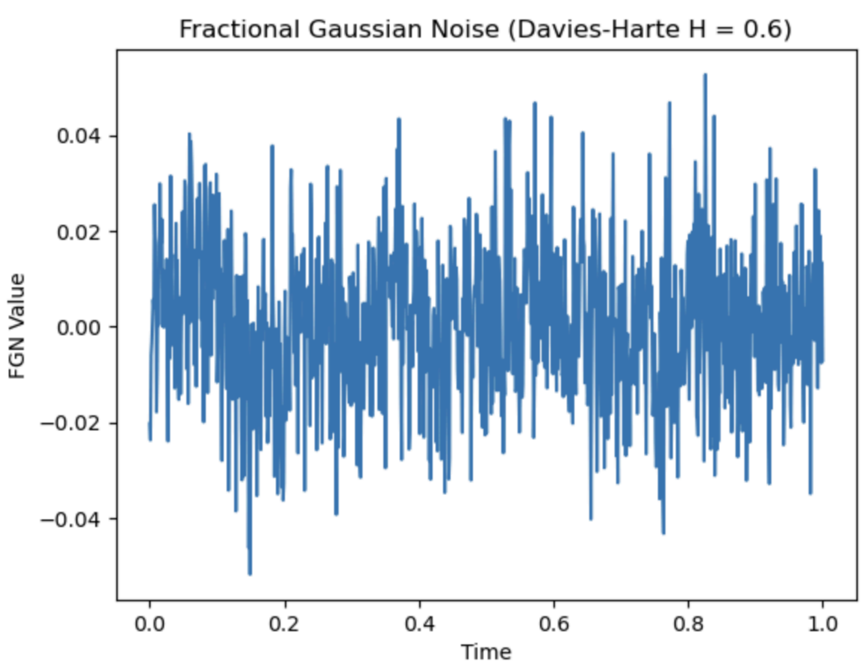


Figure 12.



The results of this process may not be as smooth as a process with a Hurst = 0.8. However, the values are more likely to be accurate. There is a trend and as the values decrease they continue to decrease before starting an upwards trend.

Discussion

All of these processes were only simulation but how could they be created and used in the real world? There are many real world uses for Fractional Brownian Motion and Fractional Gaussian Noise. It can be used for financial stock analysis, environmental analysis such as checking rainfall data or river flow data, climate change, internet traffic, or biomedical data. A real world example created using Apple stock data. Figure 13 and Figure 14 show how FBM and FGN can be used to analyze stock data.

Figure 13.

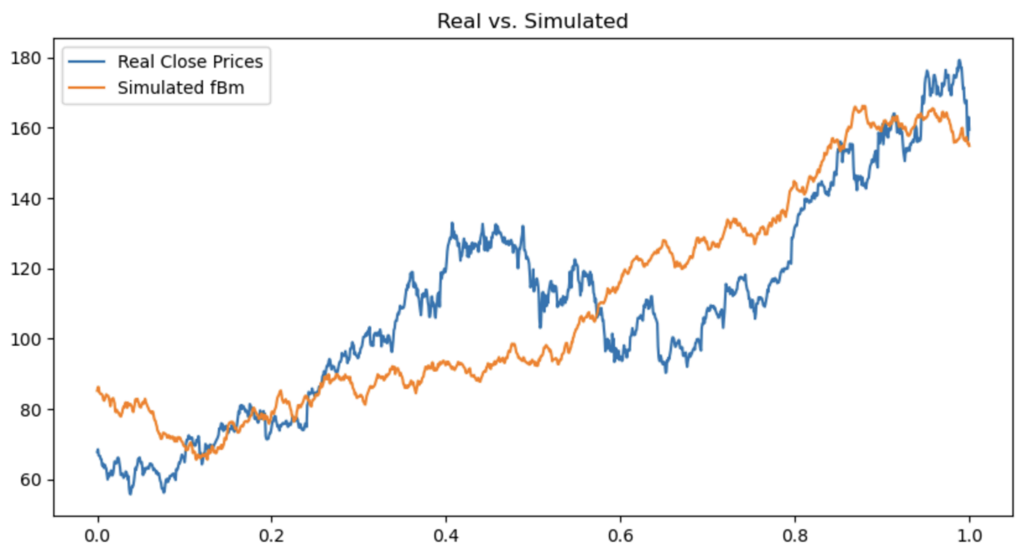
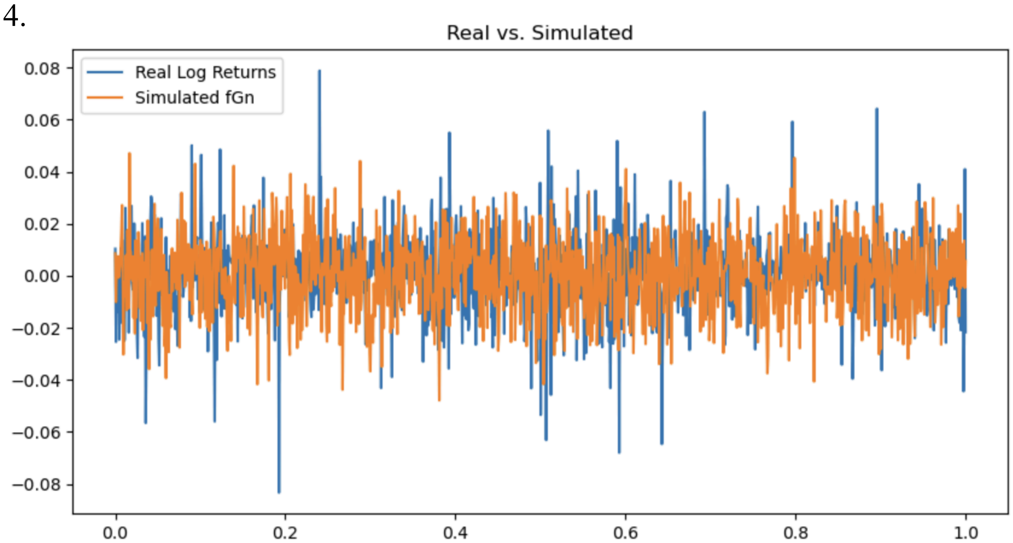


Figure 14.



Using the Davies-Harte method for better accuracy and memory efficiency and a Hurst value 0.6 for less numerical errors, the FBM process of real vs simulated data appears to have some similarities. The simulated data is persistent as the trend is similar to that of the real data. While the values are not accurate the graph still shows an insightful process that can be interpreted for stock price trends. The FGN appears to be more accurate as the values of the simulated data are closely related to the actual values. The log returns are the percentage of change between prices. Since the simulated data shows persistent trends and trends similar to the actual data, the simulated FGN aligns closely with the actual data.

Conclusion

Fractional Brownian Motion and Fractional Gaussian Noise are important for simulating data by implementing memory. Since it uses memory it can help understand why certain data may have many ups and downs. They are used across many fields such as finance, environmental, geography, and medicine. These processes give scientists a more realistic way to simulate data. When there isn't enough data scientists can use these methods to simulate and analyze how the data may behave. In order to understand both Fractional Brownian Motion and Fractional Gaussian Noise different processes were created using different methods to help visualize how they work. FBM allows scientists to simulate trends using memory in order to capture persistence and anti-persistence in ways that Brownian Motion may not.

References

1. https://thebookofshaders.com/13/
2. https://www.sciencedirect.com/topics/mathematics/fractional-brownian-motion
3. https://www.columbia.edu/~ad3217/fbm/thesisold.pdf
4. https://pictureperfectportfolios.com/how-to-use-the-hurst-exponent-strategy-in-trading/