Assignment 5

Reid Ginoza

3/8/2020

Simulation

This code is calculating the Itô integral and the Stratonovich integral for integrals with known solutions. This follows (Higham 2001), but adds in multiple trials in a vectorized manner.

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


def multiple_brownian_motion(end_time=1., num_tsteps=500, n_trials=1000):
    dt = (end_time - 0) / num_tsteps
    dw = np.random.normal(scale=np.sqrt(dt), size=(num_tsteps+1, n_trials))
    # Brownian motion must start at time 0 with value 0
    dw[0] = np.zeros_like(dw[0])
    w = dw.cumsum(axis=0)
    t = np.linspace(0, end_time, num=num_tsteps+1)  # not used in calculations
    assert w.shape[0] == t.shape[0], f'time and position arrays are not the same length. w.shape[0] - t
    assert w.shape == dw.shape, f'position and velocity arrays are not the same shape: w.shape: {w.shap
    return t, w, dw


def plot_results(approx, exact, title=''):
    plt.figure()
    plt.plot(exact, '.', label='Exact')
    plt.plot(approx, '+', alpha=0.5, label='Approx.')
    plt.legend()
    plt.title(title)
    plt.show()


def error_plot(approx, exact, title=''):
    plt.figure()
    plt.hist(approx-exact, bins='auto')
    plt.ylabel('Frequency')
    plt.xlabel('Error')
    plt.title(title)
    plt.show()
```

Set up for Trial

```
END_TIME = 1
NUM_TSTEPS = 500
N_TRIALS = 1000
dt = END_TIME / NUM_TSTEPS
t, w, dw = multiple_brownian_motion(END_TIME, NUM_TSTEPS, N_TRIALS)
```

Itô Integral

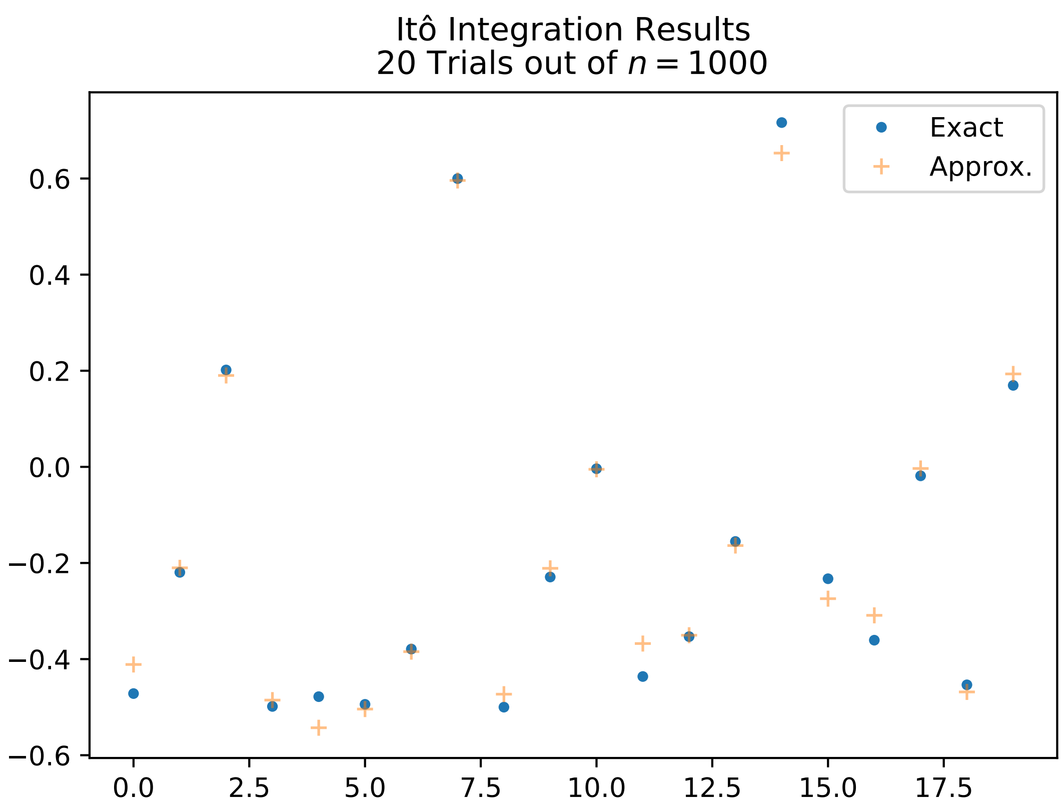
The Itô integral of $W(t)$ is known to be the following

$$\int_0^T W(t)dW = \frac{1}{2}W(T)^2 - \frac{1}{2}T.$$

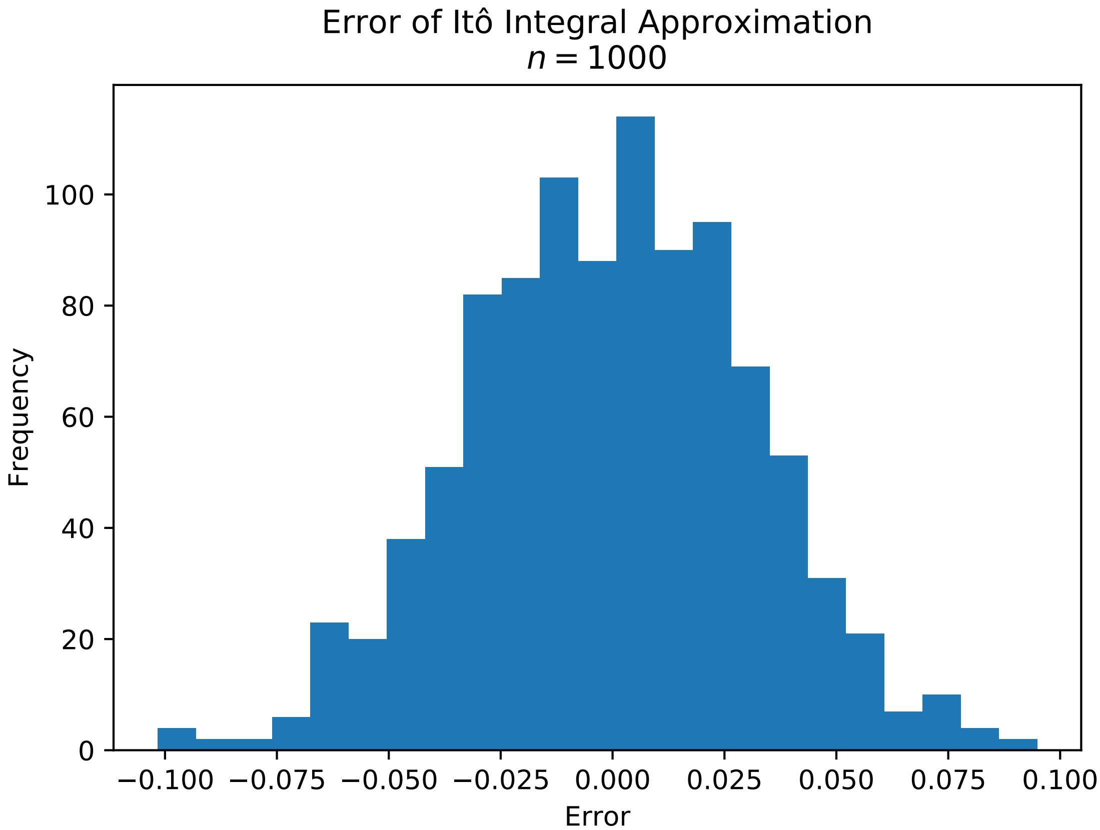
The integral is approximated by the following:

$$\int_0^T W(t)dW \approx \sum_{j=0}^{N-1} W(t_j)\left(W(t_{j+1}) - W(t_j)\right).$$

```
def ito_integral(w, dw):
    return (w[:-1]*dw[1:]).sum(axis=0)


def exact_ito(w, end_time):
    return 0.5 * (w[-1]**2 - end_time)


ito_approx = ito_integral(w, dw)
ito_ex = exact_ito(w, END_TIME)
ito_error = np.abs(ito_approx-ito_ex)
plot_results(ito_approx[:20], ito_ex[:20], f'Itô Integration Results\n20 Trials out of $n={N_TRIALS}$')
```



```python
error_plot(ito_approx, ito_ex, f'Error of Itô Integral Approximation\n$n={N_TRIALS}$')
```



Stratonovich Integral

The Stratonovich integral of $W(t)$ is known to be the following

$$\int_0^T W(t)dW = \frac{1}{2}W(T)^2.$$

The integral is approximated by the following:

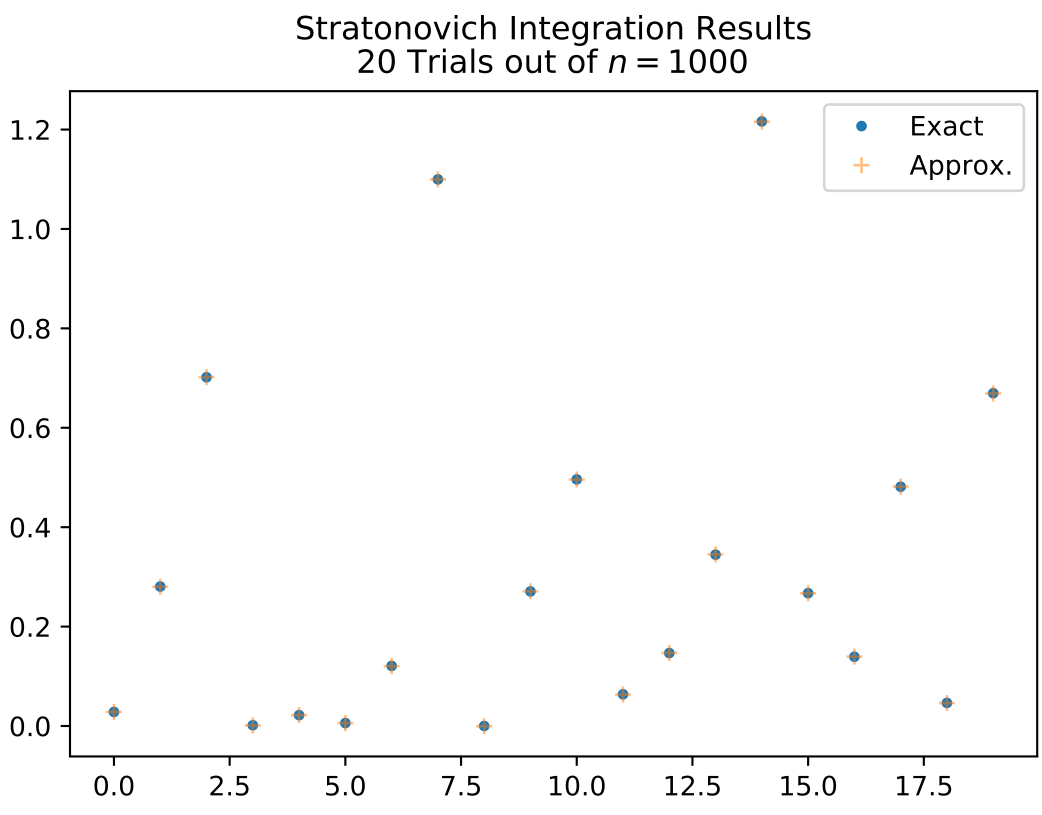
$$\int_0^T W(t)dW \approx \sum_{j=0}^{N-1} W\left(\frac{t_j + t_{j+1}}{2}\right)\left(W(t_{j+1}) - W(t_j)\right),$$

but since $W\left(\frac{t_j + t_{j+1}}{2}\right)$ is not computed directly, it is approximated as follows:

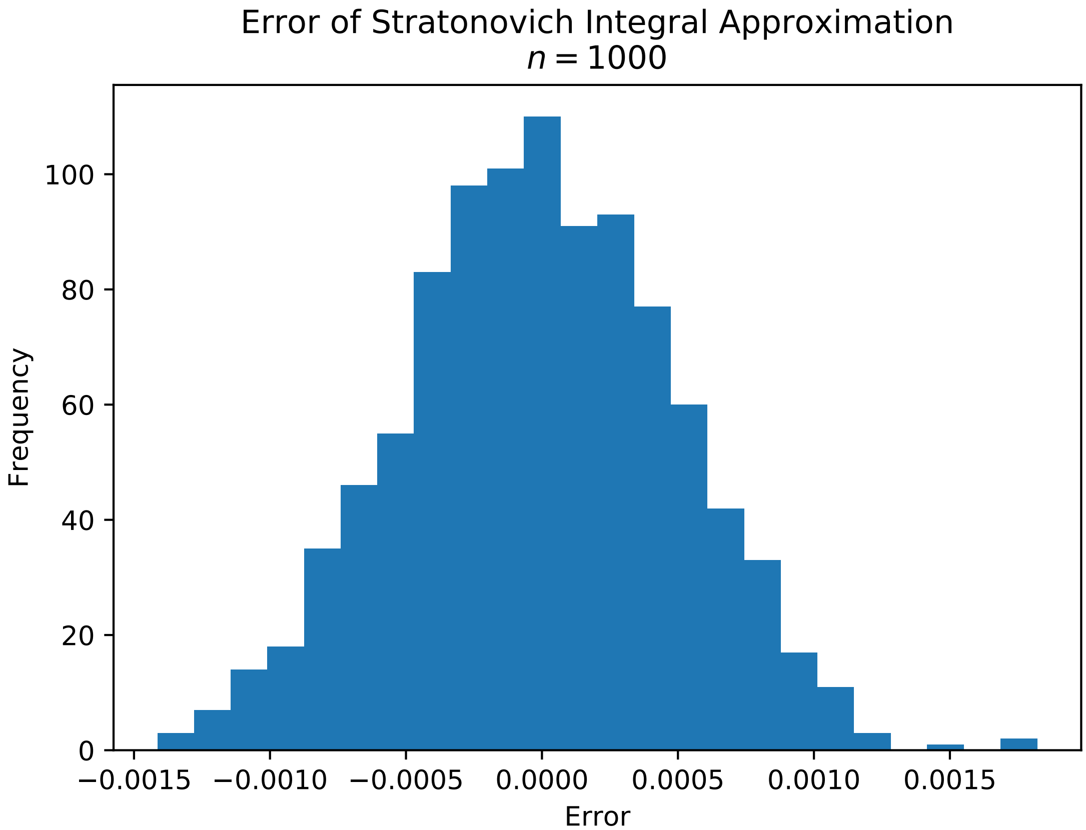
$$W\left(\frac{t_j + t_{j+1}}{2}\right) \approx \frac{W(t_j) + W(t_{j+1})}{2} + \Delta Z_j,$$

where for each j, ΔZ_j is independent and $\Delta Z_j \sim N\left(0, \frac{\Delta t}{4}\right)$. Below, this is calculated in the function `approx_w_midpt`.

```python
def approx_w_midpt(w, dt):
    """ Takes w, a vector of length N returns a vector of length N-1
    given W(t_j) and W(t_{j+1}), this approximates W evaluated at the
    midpoint, ie. W( (t_j+t_{j+1}) / 2 )"""
    return 0.5*(w[:-1] + w[1:]) + np.random.normal(scale=dt/4, size=w[:-1].shape)


def stratonovich_integral(w, dw, dt):
    w_mid = approx_w_midpt(w, dt)
    return (w_mid * dw[1:]).sum(axis=0)


def exact_stratonovich(w):
    return 0.5 * w[-1]**2


strat_approx = stratonovich_integral(w, dw, dt)
strat_ex = exact_stratonovich(w)
strat_error = np.abs(strat_approx-strat_ex)
plot_results(strat_approx[:20], strat_ex[:20], f'Stratonovich Integration Results\n20 Trials out of $n=
```



```python
error_plot(strat_approx, strat_ex, f'Error of Stratonovich Integral Approximation\n$n={N_TRIALS}$')
```



Analytical

This is exercise 4.1(a) from (Särkkä and Solin 2019), but using the notation W for Brownian motion as (Evans 2012). Compute the Ito differential of $\phi\left(W, t\right) = t + \exp\left(W\right)$.

Since ϕ is a function of a random variable, we will first look at $u : \mathbb{R} \times [0, T] \to \mathbb{R}$:

$$\begin{aligned} u(x, t) &= t + \mathrm{e}^x && (1) \\ u_x &= \mathrm{e}^x && (2) \\ u_{xx} &= \mathrm{e}^x && (3) \\ u_t &= 1 && (4) \end{aligned}$$

where the subscripts denote the variable that the derivative was taken respect to.

Now, let's consider the random variable X, which is a function of Brownian motion W:

$$\begin{aligned} X(t) &= W(t) && (5) \\ dX &= dW && (6) \\ dX &= \underbrace{0dt}_{F=0} + \underbrace{1}_{G=1} dW && (7) \end{aligned}$$

Since X is a solution of the differential equation of the form $dX = Fdt + GdW$, we may apply Ito's chain rule.

$$\phi := u\left(X, t\right) \tag{8}$$

$$d\phi = du = \left(u_t + u_x F + \frac{1}{2} u_{xx} G^2\right) dt + u_x G dW \qquad \text{Itô's chain rule} \tag{9}$$

$$d\phi = \left(1 + \frac{1}{2} \mathrm{e}^W\right) dt + \mathrm{e}^W dW \tag{10}$$

References

Evans, Lawrence C. 2012. *An Introduction to Stochastic Differential Equations*. Vol. 82. American Mathematical Soc.

Higham, Desmond J. 2001. "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations." *SIAM Review* 43 (3). SIAM: 525–46.

Särkkä, Simo, and Arno Solin. 2019. *Applied Stochastic Differential Equations*. Vol. 10. Cambridge University Press.