Computational Physics Homework 1

Fang Yi Chu

September 25, 2013

Problem 1

a) The codes for Differentiation under single precision are in the file named "hw1.py".

b) Please run the codes and the derivative and its relative error will be generated in the file.

c) The log-log plots of relative error ε vs step h are as below.

1) Forward-Difference Algorithms

The total error of forward-difference algorithms:

$$\varepsilon_{total} \sim \; hf'' + \frac{\varepsilon_f |f|}{h}$$

The first term is truncation error and the second term is roundoff error.
Optimal choice of h:

$$h \sim \sqrt{\frac{\varepsilon_f |f|}{f''}} \sim \; 10^{-3} - 10^{-4}$$

Relative error ε

$$\varepsilon \sim \sqrt{\varepsilon_f} \sim \sqrt{\varepsilon_m} \sim \; 10^{-3} - 10^{-4}$$

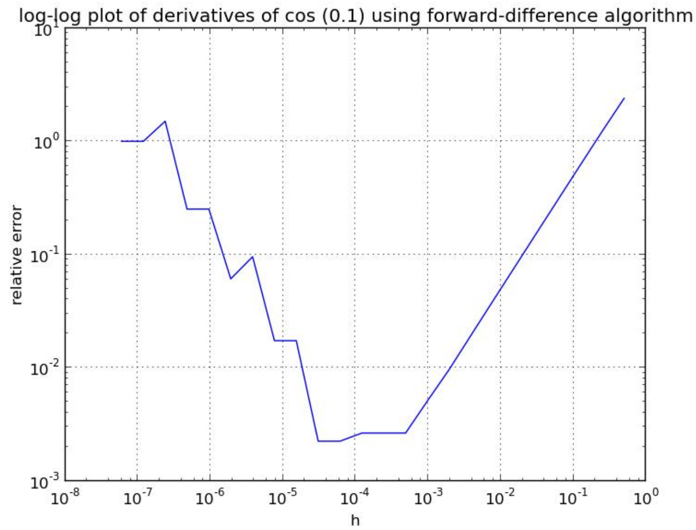

Figure 1: derivate of cos(x) at x=0.1 by forward-difference algorithm

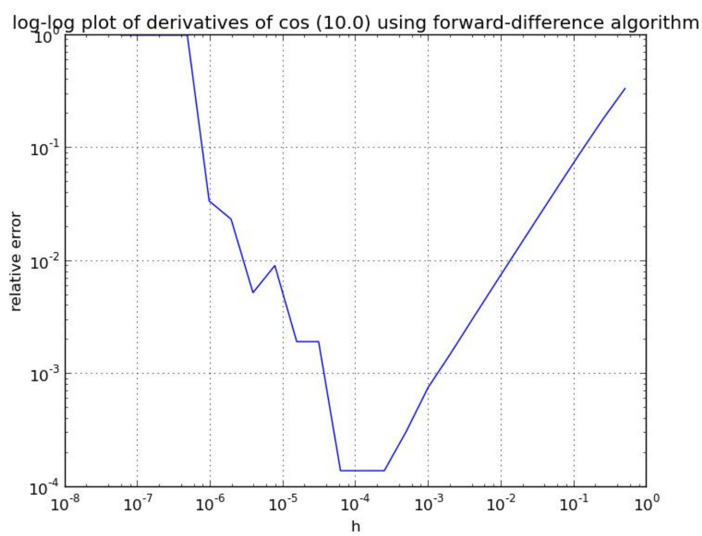

Figure 2: derivate of cos(x) at x=10.0 by forward-difference algorithm

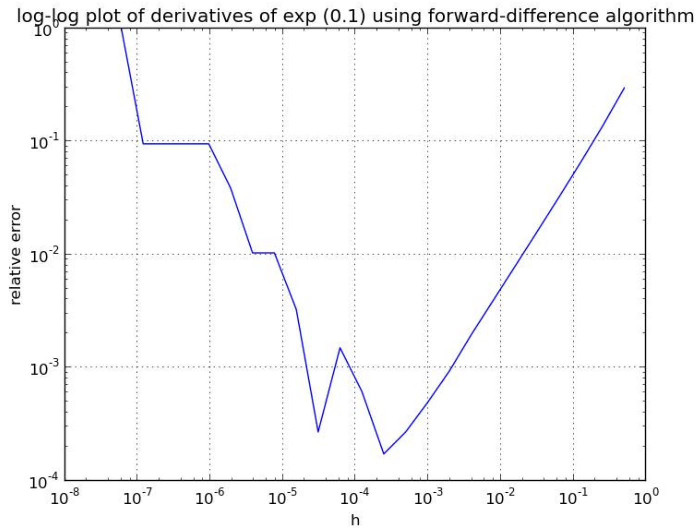

Figure 3: derivate of exp(x) at x=0.1 by forward-difference algorithm

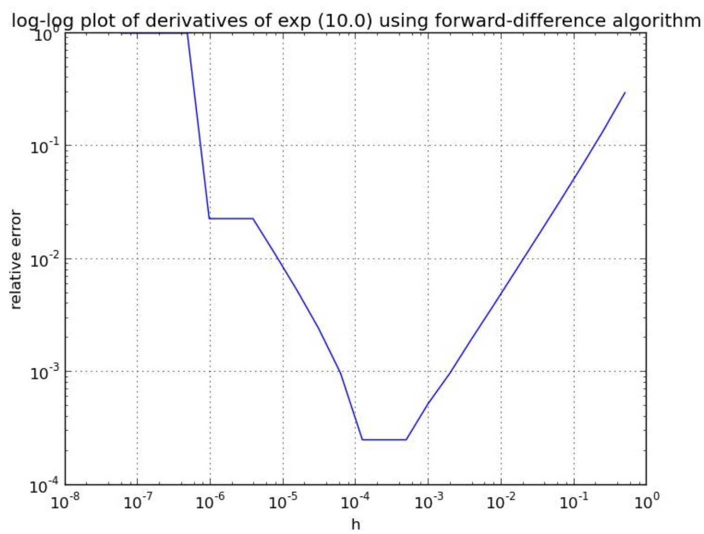

Figure 4: derivate of exp(x) at x=10.0 by forward-difference algorithm

When h decreases to $10^{-3} - 10^{-4}$, we can get the smallest relative error $10^{-3} - 10^{-4}$, which is what we expect. Truncation error dominates in large h, and the slope of truncation error is proportional to $hf"(x)$, which is also correspond to our expectation. Also, the sawtooth pattern reflects the roundoff error which dominates in small h.

2) Central-Difference Algorithms

The total error of central-difference algorithms

$$\varepsilon_{total} \sim h^2 f''' + \frac{\varepsilon_f |f|}{h}$$

Optimal choice of h:

$$h \sim (\frac{\varepsilon_f |f|}{f'''})^{1/3}$$

Relative error ε

$$\varepsilon \sim (\varepsilon_f)^{2/3} \sim \ 10^{-5}$$

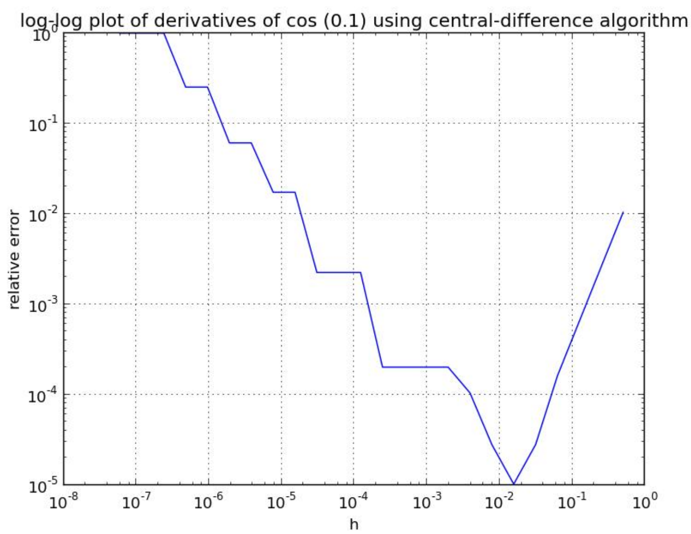

Figure 5: derivate of cos(x) at x=0.1 by central-difference algorithm

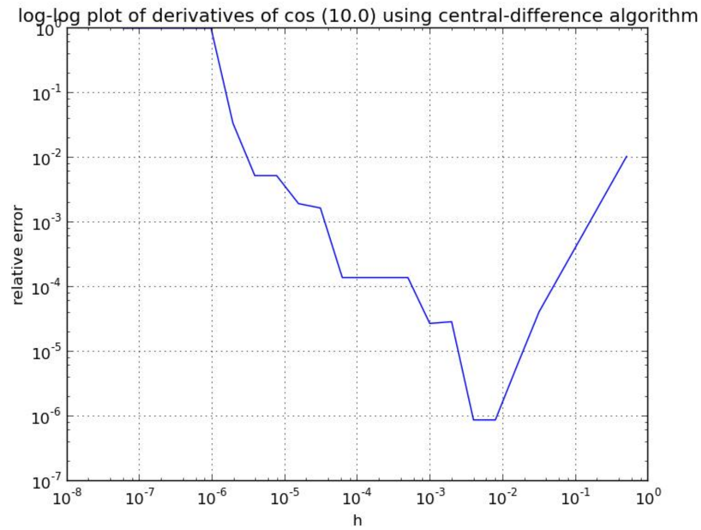

Figure 6: derivate of cos(x) at x=10.0 by central-difference algorithm

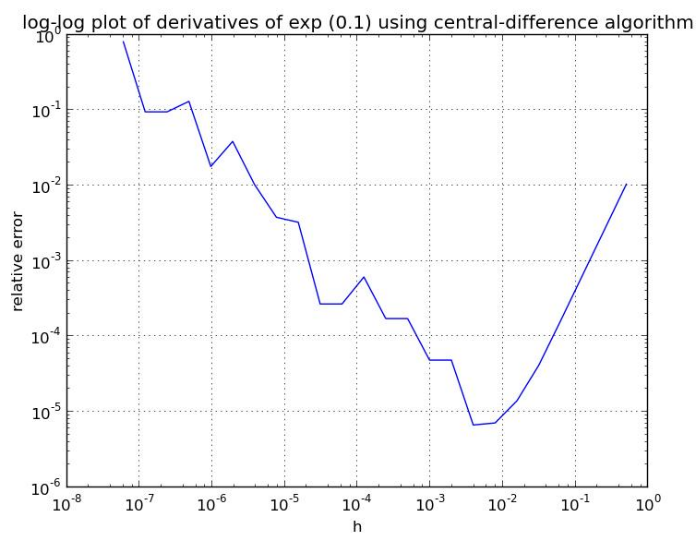

Figure 7: derivate of exp(x) at x=0.1 by central-difference algorithm

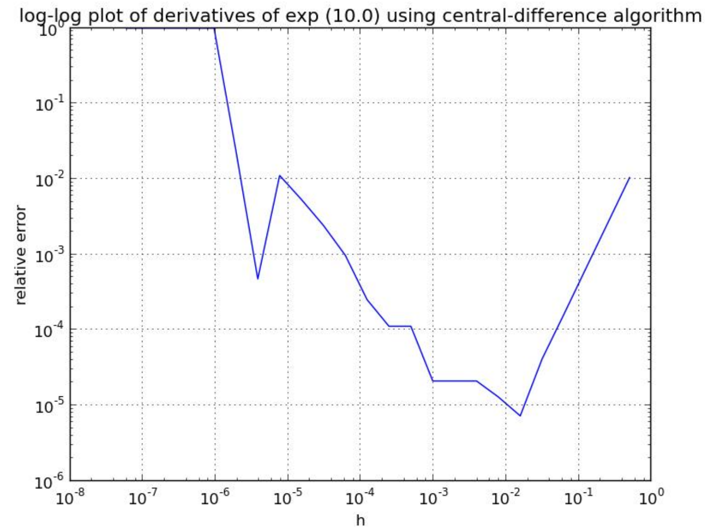

Figure 8: derivate of exp(x) at x=10.0 by central-difference algorithm

When h goes down to 10^{-2}, we can get the smallest relative error 10^{-5}, which is what we expect. Truncation error dominates in large h, and the slope of truncation error is proportional to $h^2 f^{(3)}(x)$, which is the same as what we expect. Also, the sawtooth pattern reflects the roundoff error which dominates in small h.

3) Extrapolated-Difference Algorithms
The total error of extrapolated-difference algorithms

$$\varepsilon_{total} \sim h^4 f^{(5)} + \frac{\varepsilon_f |f|}{h}$$

Optimal choice of h:

$$h \sim (\frac{\varepsilon_f |f|}{f^{(5)}})^{1/5}$$

Relative error ε

$$\varepsilon \sim (\varepsilon_f)^{4/5} \sim \ 10^{-6}$$

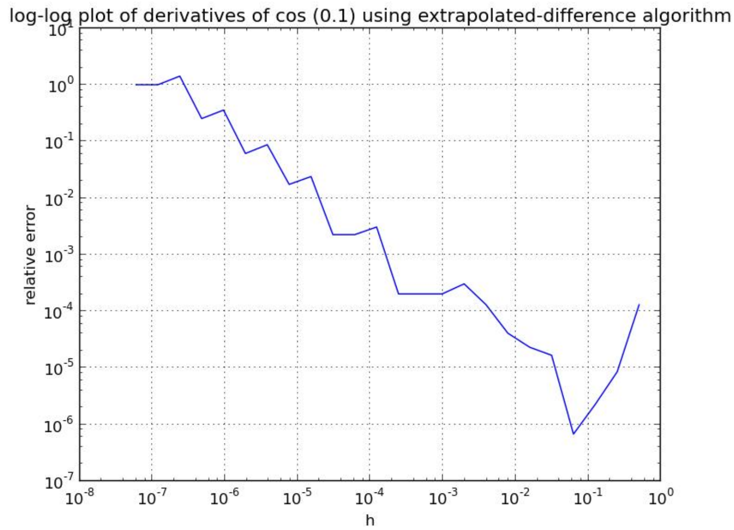

Figure 9: derivate of cos(x) at x=0.1 by extrapolated-difference algorithm

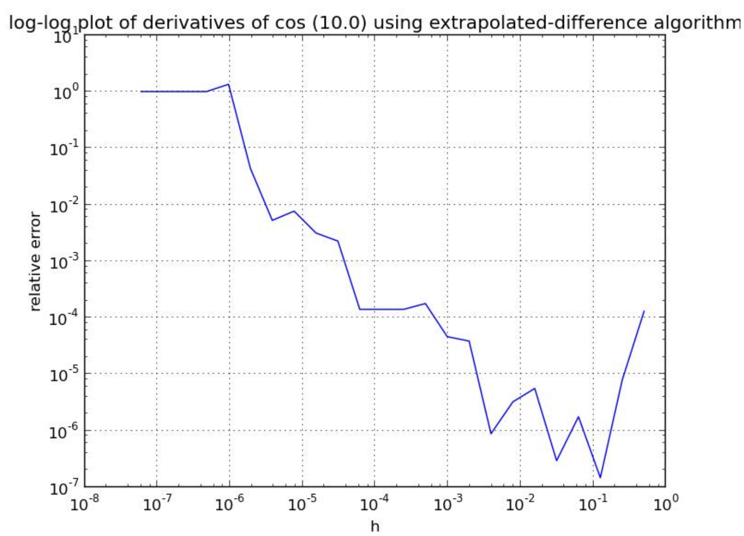

Figure 10: derivate of cos(x) at x=10.0 by extrapolated-difference algorithm

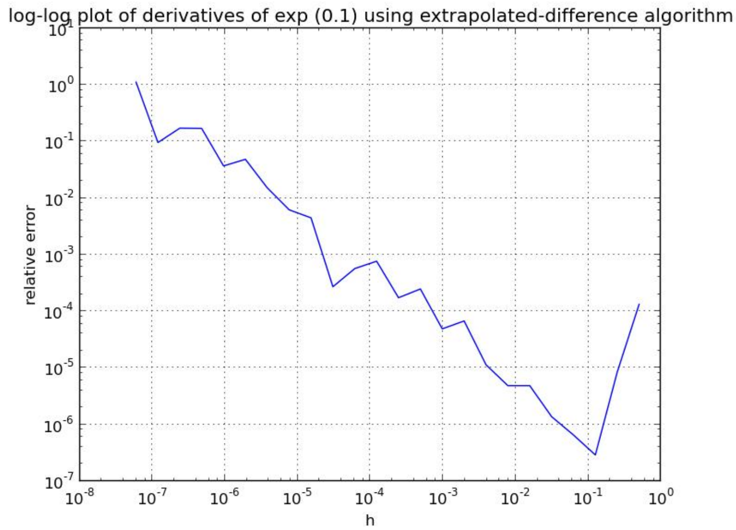

Figure 11: derivate of exp(x) at x=0.1 by extrapolated-difference algorithm

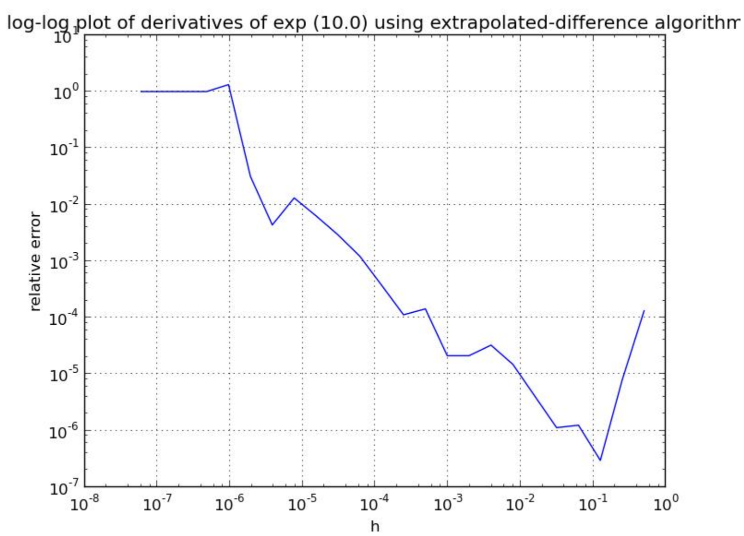

Figure 12: derivate of exp(x) at x=10.0 by extrapolated-difference algorithm

When h reaches 10^{-1}, we can obtain the smallest relative error $10^{-6} - 10^{-7}$, which corresponds to our expectation. Truncation error dominates in large h, and the slope of truncation error is proportional to $h^4 f^{(5)}(x)$. Also, the sawtooth pattern reflects the roundoff error which dominates in small h.

Problem 2

The codes for Integration under single precision are in the file named "hw1.py".

The log-log plot of relative error as a function of the number of interval N is as followed.

1) Trapezoid rule

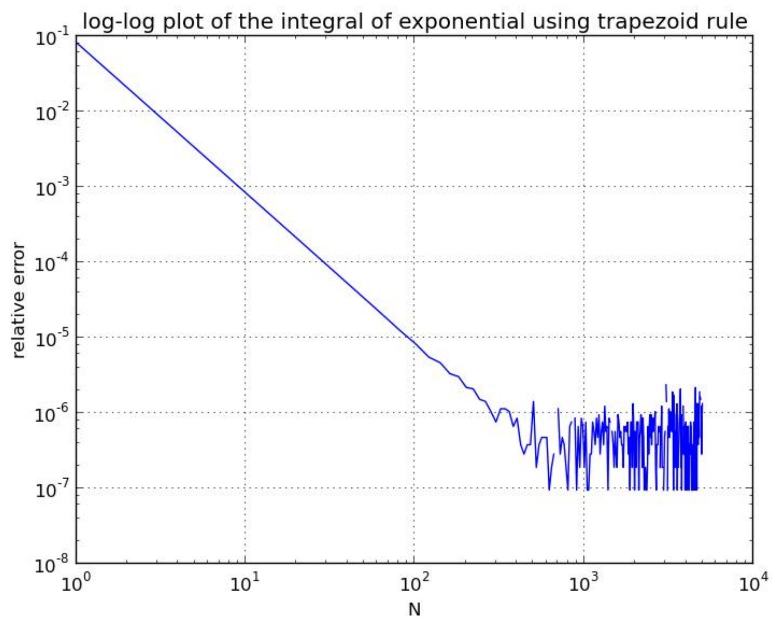

Figure 13: Trapezoid rule

In the plot of trapezoid rule, relative error decreases as the number of interval N increases because the order of truncation error is proportional to $\frac{f"(x)}{N^2}$. The optimal N is ~ 600 and the error is also close to 3×10^{-6}, which both correspond to the expectation.

2) Simpson's rule

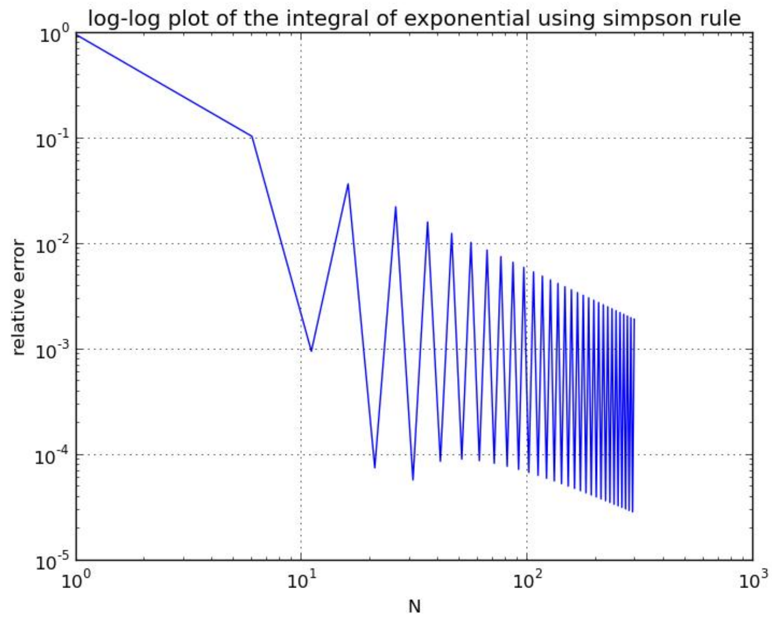

Figure 14: Simpson's rule

In the plot of Simpson's, truncation error is proportional to $\dfrac{f^{(5)}(x)}{N^4}$. Roundoff error is proportional to $\sqrt{N}\varepsilon_m$. The expected relative error at optimal N should be $\sim 6 \times 10^{-7}$, but in my plot the error is jumping dramatically. Therefore, although the minimum of relative error is around the expected optimal N ($\sim \varepsilon_m^{-2/5} \sim 30$), it is really hard to define.

3) Gauss-Legendre quadrature

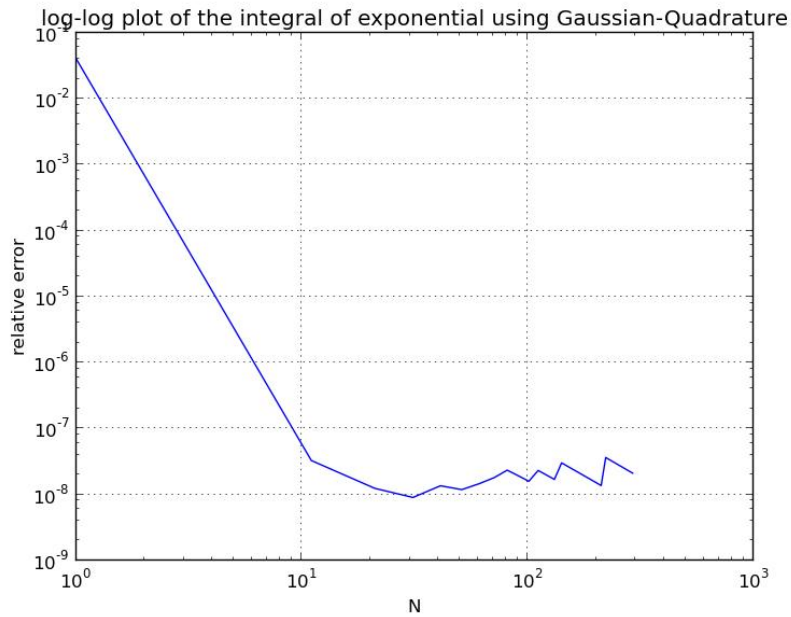

Figure 15: Gauss-Legendre quadrature

In the plot of Gauss-Legendre quadrature, the optimal N is around 20, and the relative error decreases to $\sim 10^{-8}$. Comparing to Simpson's rule, Gauss-Legendre have more freedom to choose weighting coefficients and abscissas,so it's more precise than trapezoid and Simpson's rules.

Problem 3

a) The codes of 1D random walk generated by different values of iseed are in the file named "hw1.py"

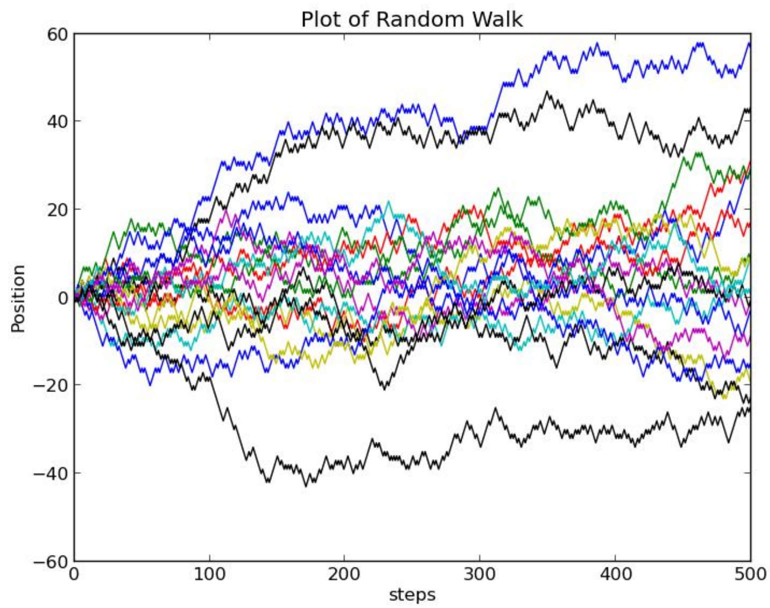

Figure 16: The path of random walk

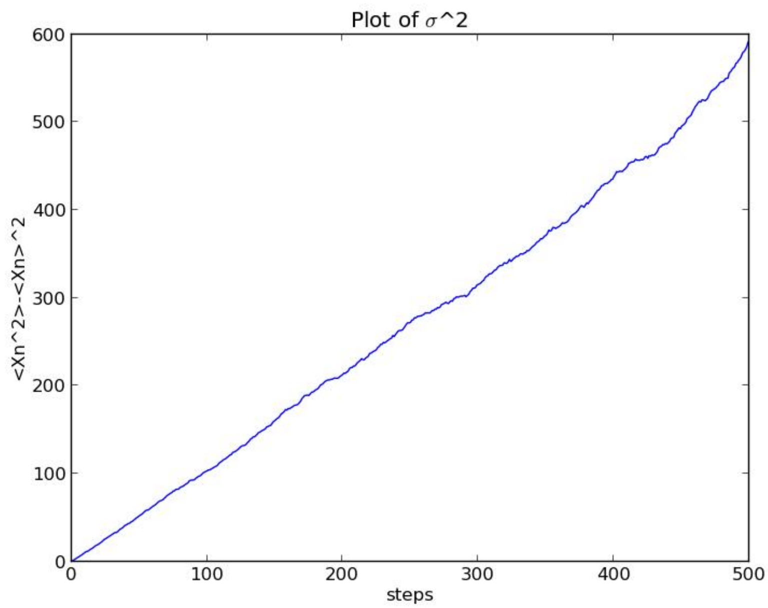

Figure 17: The σ_n^2 as a function of n

Set up a recursion relation:

$$x_n = x_{n-1} \pm l$$

Square, we will get:

$$x_n^2 = x_{n-1}^2 \pm 2x_{n-1}l + l^2$$

Calculate the expectation value:

$$\langle x_n^2 \rangle = \langle x_{n-1}^2 \rangle \pm 2\langle x_{n-1} \rangle \langle l \rangle + \langle l^2 \rangle$$

If N is large enough:

$$\langle x_{n-1} \rangle = 0$$

Then, we summate equations above from $n = 1$ to $n = n - 1$, and simplify:

$$\langle x_n^2 \rangle = nl^2$$

So, this explains σ_n^2 is in direct proportion to n in Figure 17.

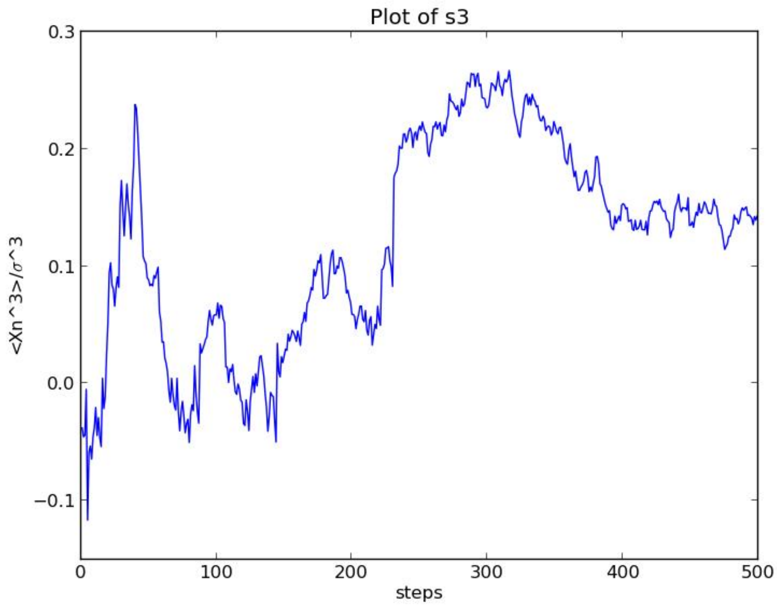

Figure 18: The $\langle x_n^3 \rangle / \sigma_n^3$ as a function of n

In the same way, we can also get:

$$\langle x_n^3 \rangle = \langle x_{n-1}^3 \rangle \pm 3\langle x_{n-1}^2 \rangle \langle l \rangle + 3\langle x_{n-1} \rangle \langle l^2 \rangle \pm \langle l^3 \rangle$$

If N is large enough:

$$\pm \langle x_{n-1}^2 \rangle = 0 = \langle x_{n-1} \rangle \ = \pm \langle l^3 \rangle$$

Therefore, we get

$$s_3 = 0$$

This explains why s_3 in Figure 18 is around zero (~ 0.15).

Additional, according to the central limit theorem, s_3 should be inversely proportional to $\sqrt{N}$. When $N \to \infty$, $s_3 \to 0$

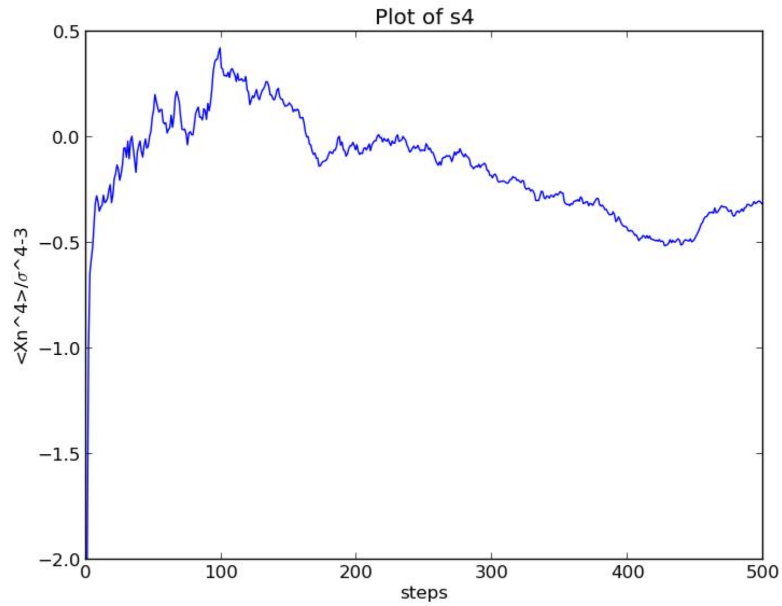

Figure 19: The $\langle x_n^4 \rangle / \sigma_n^4 - 3$ as a function of n

For s_4 , we can get:

$$\langle x_n^4 \rangle = \langle x_{n-1}^4 \rangle \pm 4\langle x_{n-1}^3 \rangle \langle l \rangle + 6\langle x_{n-1}^2 \rangle \langle l^2 \rangle \pm 4\langle x_{n-1} \rangle \langle l^3 \rangle + \langle l^4 \rangle$$

If N is large enough:

$$\pm \langle x_{n-1}^3 \rangle \langle l \rangle = 0 = \pm \langle x_{n-1} \rangle \langle l^3 \rangle$$

Then, we summate equations above from $n = 1$ to $n = n - 1$, and simplify:

$$\langle x_n^4 \rangle = 3n^2 + 4n$$

In the end, we get:

$$s_4 = (\langle x_n^4 \rangle / \sigma_n^4) - 3 \simeq 0$$

In Figure 19, we can see that s_4 approaches zero (~ -0.3).
In addition, according to the central limit theorem, s_4 should be inversely proportional to N. When $N \to \infty$, $s_3 \to 0$