Ps-3

Jaewoo Lee

September 24, 2023

Abstract

This document contains results of ps_2. The code written to solve the problems could be found in my github: patrick612. Codes for solutions to all problems could be found in ps_3.py.

1

In part a), using $\delta = 10^{-2}$ for the derivative of $x(x-1)$ evaluated at $x = 1$ yielded 1.010000000000001 while the analytic solution gave 1. This is because the derivative is defined such that $\lim_{\delta \to 0} \delta$. However the written code approximates this limit by inputting a small $\delta = 10^{-2}$, leading to some error.

For part b) the same calculation was repeated over different values of $\delta = [10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}, 10^{-14}]$. The values returned by the function were $[1.0000999999998899, 1.0000009999177333, 1.0000000039225287, 1.000000082840371, 1.0000889005833413, 0.9992007221626509]$. We see that the correspoding errors are $[-9.999999988985486e{-}05, -9.99917733279787e{-}07, -3.922528746258536e{-}09, -8.284037100736441e-08, -8.890058334132256e-05, 0.0007992778373491216]$. The increase in error occurs starting $\delta = [10^{-12}$ and this was due to truncation in significant figures after dividing by a very small number. The error due to significant figures start to overpower the accuracy gained from taking smaller δ.

2

The matrix multiplication of NxN matrices of varying sizes were computed and the computation time were found using both for loop and the dot product. Figure 1 and Figure 2 below shows how the computation time scales. The computation time using dot product scales linearly. The spike in the middle is an anomaly due to the fact that the computational time is low such that the ratio of computational time to its uncertainty is small. We have also shown that the computation time for for loop method scales like NE3 by fitting a curve with a multiplying coefficient that brings it to scale to the actual computation time.

3

For each decay steps individual functions were written. Each function uses a random number generator to determine if an isotope created at certain time will decay given the current time for all isotopes of each type at every time step. Plot containing number of each particles at each

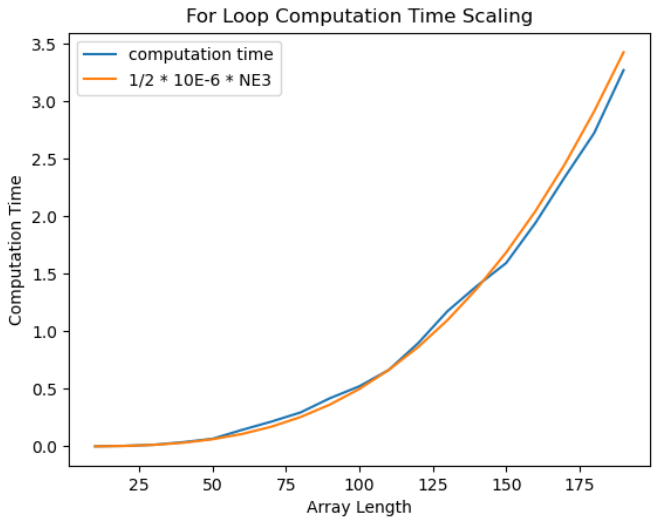

Figure 1: Computation time scaling of matrix multiplication using for loop with a fitted curve of equation Coomputation Time = 5E10 * NE3

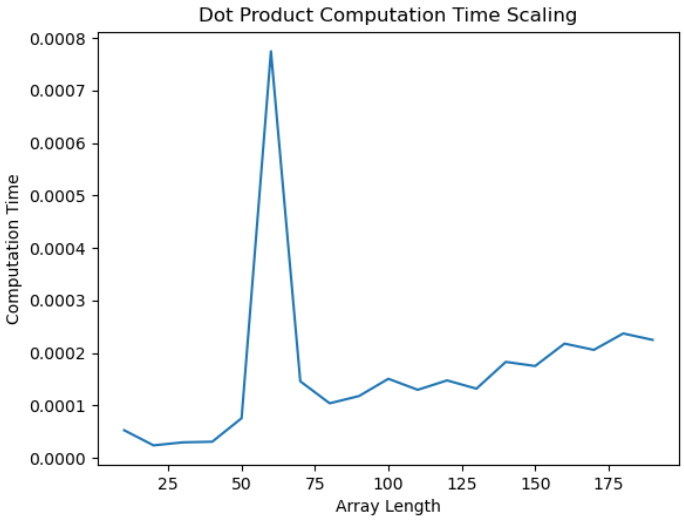

Figure 2: Computation time scaling of matrix multiplication using dot product.

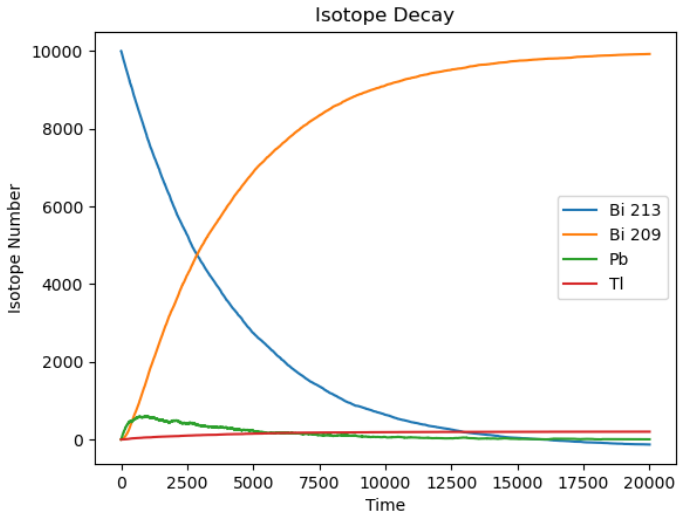

Figure 3: Number of isotopes at every time step

time is shown in Figure 3 From the plot we observe the expected decay behavior from Bi 213 to Bi 209. While Bi 213 has to decay into Pb in order to decay into Bi 209, we see relatively low number of Pb isotopes throughout because its half life is much shorter than that of Bi 213. Furthermore, we observe a low number of Tl throughout.

4

In order to transform a uniform distribution into the decay distribution of Tl 208, $P(t) = 2^{-\frac{t}{\tau}} * \frac{ln2}{\tau}$, we note that $2^{\frac{-t}{\tau}} = (e^{ln2})^{\frac{-t}{\tau}} = e^{\frac{-t}{\tau} * ln2}$. Thus $P(t) = \frac{ln2}{\tau} * e^{\frac{-ln2}{\tau} * t}$ and $\mu = \frac{ln2}{\tau}$. This gives us $t = \frac{-\tau}{ln2} * ln(1-z)$ where t is the time at which the Tl 208 has decayed in accord with its probability distribution given z a number sampled from a uniform distribution. Using such transformation, the number out of 1000 isotopes that have not decayed in time is shown in Figure 4.

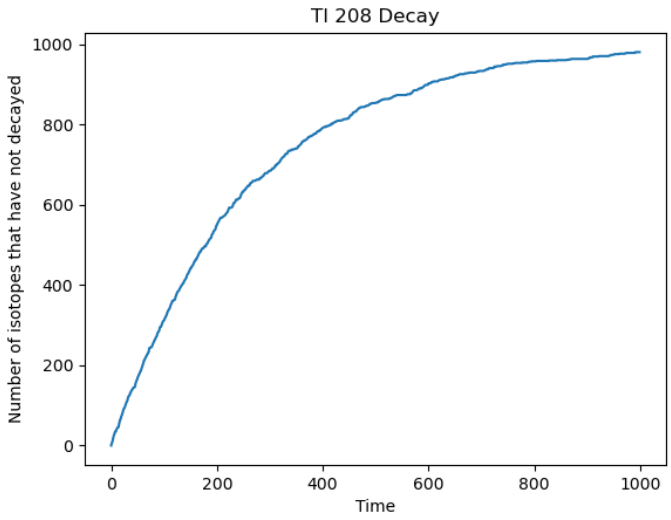

Figure 4: Number of Tl 208 isotopes that have not decayed in time.