Ahmet Emre Nas
21402357
CS342-2

Project – 1 Report

**How I calculated Trapezoidal rule in the project:*

One of the way to calculate trapezoidal rule is taking average of function for upper and lower x values and multiplying it by the difference of upper and lower.

$$\int_a^b f(x)\,dx \approx (b-a)\left[\frac{f(a)+f(b)}{2}\right].$$

However more approximate result can be get by portioning the integration interval.

$$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1})+f(x_k)}{2}\Delta x_k\,.$$

This formula can be written as following.

$$\begin{aligned}\int_a^b f(x)\,dx &\approx \frac{\Delta x}{2}\sum_{k=1}^{N}\left(f(x_{k-1})+f(x_k)\right)\\ &= \frac{\Delta x}{2}\left(f(x_0)+2\sum_{k=1}^{N-1} f(x_k)+f(x_N)\right)\\ &= \frac{\Delta x}{2}\left(f(x_0)+2f(x_1)+2f(x_2)+2f(x_3)+\cdots+2f(x_{N-1})+f(x_N)\right)\end{aligned}$$

I have calculated the integrals in my program with this last formula.
(Images of formulas is taken from Wikipedia)

Part C)

All of the times calculated by time command and real time has been written into report. In the tables x axis shows the number of Sub Intervals and y axis shows number of child processes.

Process and Pipe:

Function: y=(x^2)+(2*x) from 0 to 250

	10	100	200	1000	1000000
1	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.023 seconds
10	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.096 seconds
25	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.215 seconds
50	0.0 minutes 0.005 seconds	0.0 minutes 0.005 seconds	0.0 minutes 0.005 seconds	0.0 minutes 0.005 seconds	0.0 minutes 0.413 seconds

Function: y = (x^2)+(x/2) from 0 to 250

	10	100	200	1000	1000000
1	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.026 seconds
10	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.002 seconds	0.0 minutes 0.107 seconds
25	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.246 seconds
50	0.0 minutes 0.005 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.005 seconds	0.0 minutes 0.006 seconds	0.0 minutes 0.479 seconds

Pthread:

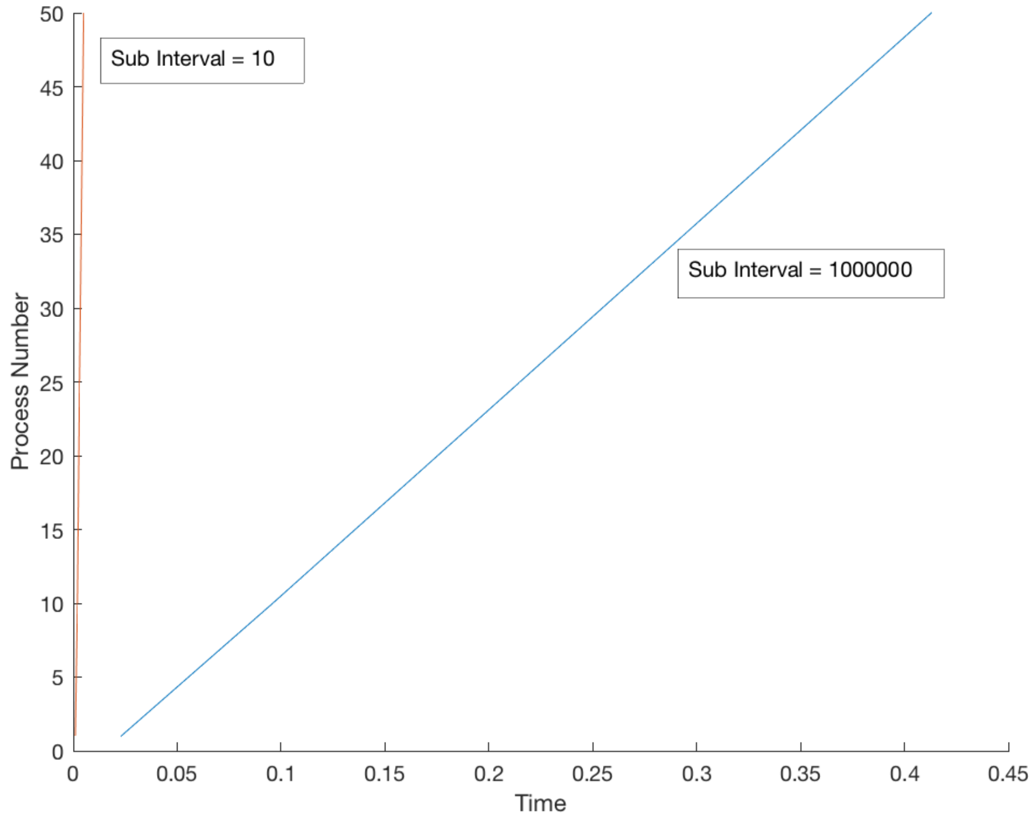
Function: y=(x^2)+(2*x) from 0 to 250

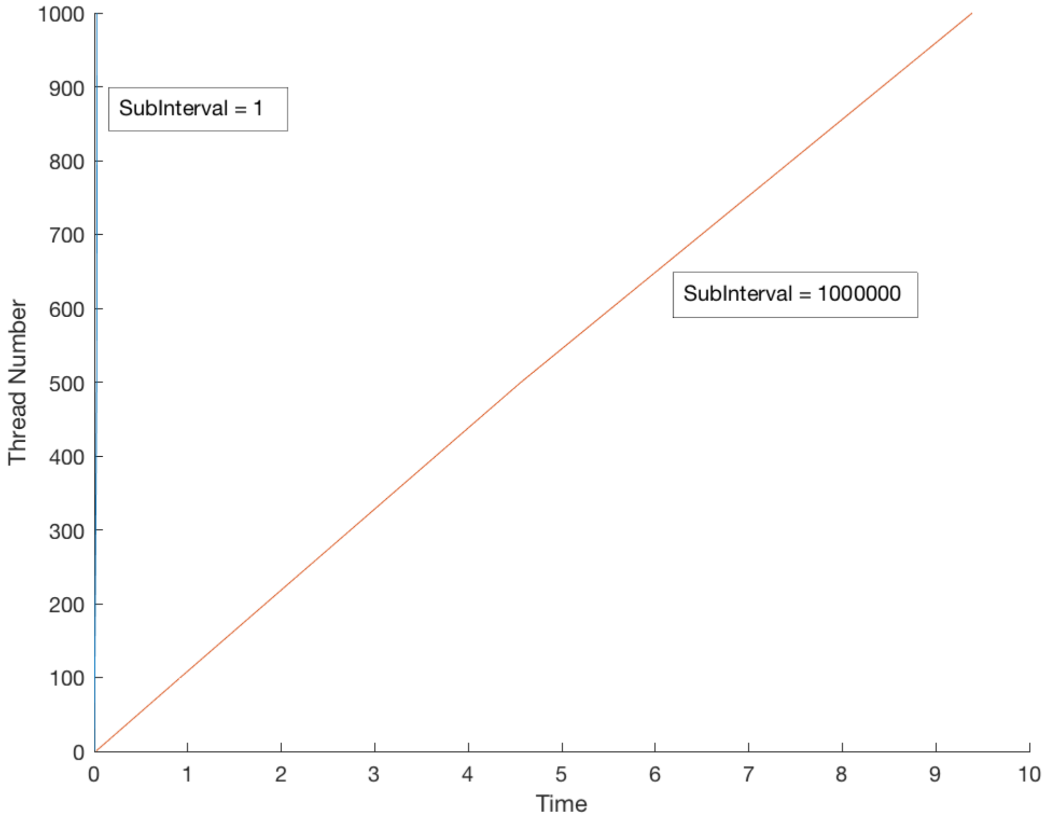
	10	100	200	1000	1000000
1	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.025 seconds
100	0.0 minutes 0.005 seconds	0.0 minutes 0.004 seconds	0.0 minutes 0.004 seconds	0.0 minutes 0.007 seconds	0.0 minutes 0.921 seconds
500	0.0 minutes 0.026 seconds	0.0 minutes 0.014 seconds	0.0 minutes 0.015 seconds	0.0 minutes 0.019 seconds	0.0 minutes 4.565 seconds
1000	0.0 minutes 0.032 seconds	0.0 minutes 0.036 seconds	0.0 minutes 0.031 seconds	0.0 minutes 0.038 seconds	0.0 minutes 9.392 seconds

Function: y = (x^2)+(x/2) from 0 to 250

	10	100	200	1000	1000000
1	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.001 seconds	0.0 minutes 0.030 seconds
100	0.0 minutes 0.004 seconds	0.0 minutes 0.004 seconds	0.0 minutes 0.003 seconds	0.0 minutes 0.005 seconds	0.0 minutes 1.049 seconds
500	0.0 minutes 0.018 seconds	0.0 minutes 0.016 seconds	0.0 minutes 0.015 seconds	0.0 minutes 0.020 seconds	0.0 minutes 5.179 seconds
1000	0.0 minutes 0.039 seconds	0.0 minutes 0.045 seconds	0.0 minutes 0.038 seconds	0.0 minutes 0.037 seconds	0.0 minutes 10.384 seconds

Plots for both pipe and thread for same function (*(x^2)+(x/2) from 0 to 250) respectively*





Acording to results threads works faster than child processes and it's less expensive than creating new processes. Therefore using pthreads in this project is more logical than using child processes