Lab 3 Report

Liron Mizrahi - 708810

August 2016

1 The Trapezoidal Rule

1.1 Derivation For The Trapezoidal Rule

For a function $f(x)$, $x \in [a,b]$ where $[a,b]$ is partitioned uniformly into N partitions with

$$a = x_0 < x_1 < x_2 < ... < x_N = b$$

The area of the i^{th} right trapezoid is given by,

$$area_i = (x_{i+1} - x_i)\left[\frac{f(x_i) + f(x_{i+1})}{2}\right]$$

But since $(x_{i+1} - x_i)$ is equal for each partition, denote

$$\Delta x = (x_{i+1} - x_i) = \frac{(b-a)}{N}$$

Now the area of the i^{th} right trapezoid is given by,

$$area_i = \Delta x\left[\frac{f(x_i) + f(x_{i+1})}{2}\right]$$

Now,

$$\begin{aligned}
\int_a^b f(x)dx &\approx \sum_{i=0}^{N-1} area_i \\
&= \sum_{i=0}^{N-1} \Delta x\left[\frac{f(x_i) + f(x_{i+1})}{2}\right] \\
&= \Delta x \sum_{i=0}^{N-1} \left[\frac{f(x_i) + f(x_{i+1})}{2}\right] \\
&= \frac{(b-a)}{N} \sum_{i=0}^{N-1} \left[\frac{f(x_i) + f(x_{i+1})}{2}\right] \\
&= \frac{(b-a)}{2N} \sum_{i=0}^{N-1} \left[f(x_i) + f(x_{i+1})\right]
\end{aligned}$$

1.2 Exact Value of $\int xe^{-x}dx$

This is evaluated using Integral By Parts.

$$\int udv = uv - \int vdu$$

$$\begin{aligned} dv &= e^{-x} & u &= x \\ \implies v &= -e^{-x} & du &= 1 \end{aligned}$$

$$\begin{aligned} \int xe^{-x}dx &= -xe^{-x} - \int -e^{-x} \cdot 1dx \\ &= -xe^{-x} - e^{-x} \\ &= -(x+1)e^{-x} + C \end{aligned}$$

But we are evaluating the integral from 0 to 20, so

$$\begin{aligned} \int_0^{20} xe^{-x}dx &= -(x+1)e^{-x} \mid_0^{20} \\ &= -(20+1)e^{-20} - (0+1)e^{0} \\ &= 1 - 21e^{-20} \\ &= 0.9999 \\ &\approx 1 \end{aligned}$$

2 Parallel Trapezoidal Integration

The parallel trapezoidal algorithm uses the end result of Section 1.1. Shown below are the graphs of the accuracy and time of the algorithm. The graphs will have a lot of noise as a result of these tests being run in a virtual machine.

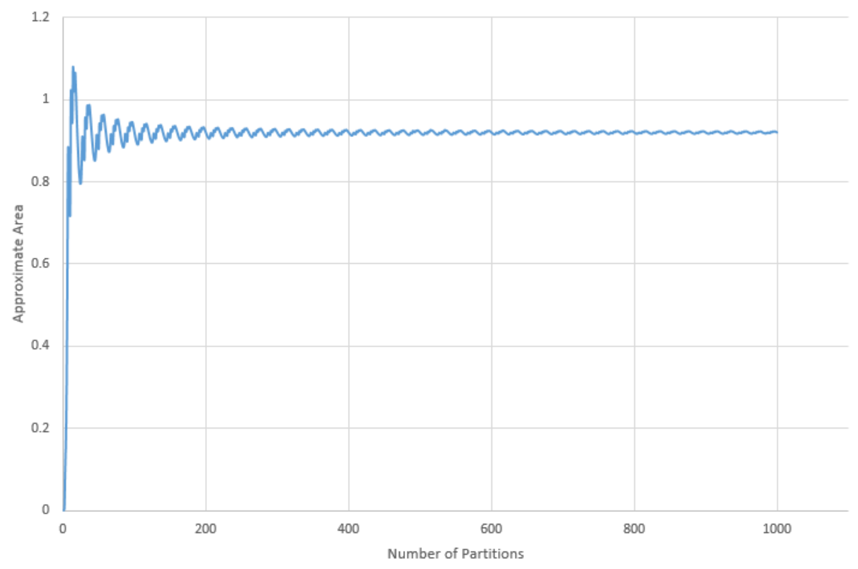

Figure 1: Accuracy of the approximated area as the number of partitions increased, with 4 threads

As the number of partitions increased the approximated area becomes closer and closer to the real value calculated above, which is approximately 1. The average area is 0.915 which is very close to the real value.

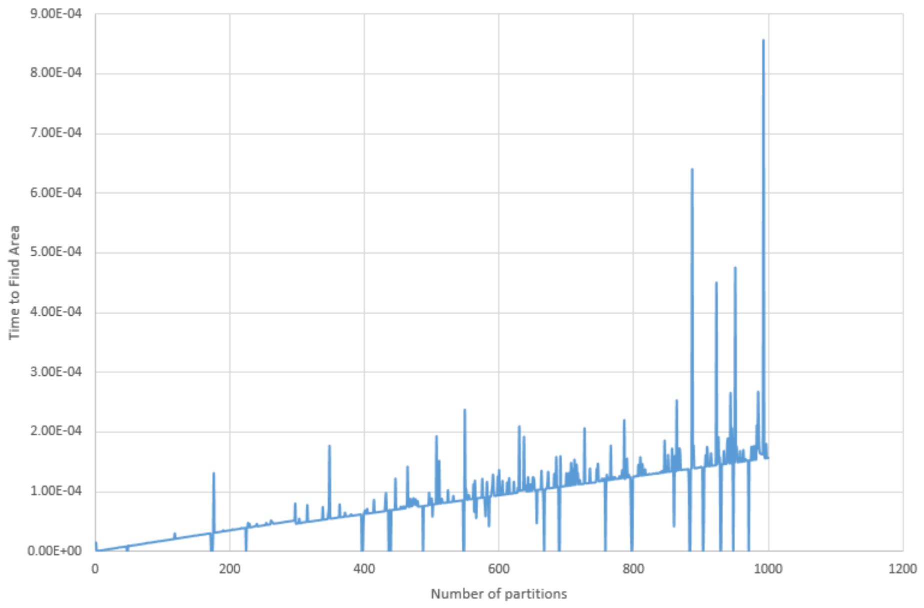

Figure 2: Time taken to find the approximated area as the number of partitions increased, with 4 threads

As the number of partitions increased the time taken to find the approximate area increases proportionally at a linear rate.

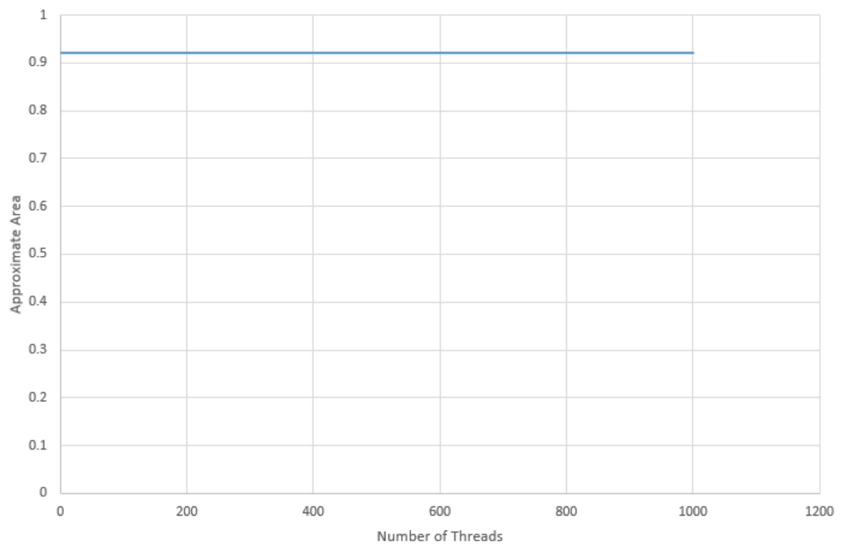

Figure 3: Accuracy of the approximated area as the number of threads increased, with 500 partitions

As expected the area obtained while increasing the threads remains constant.

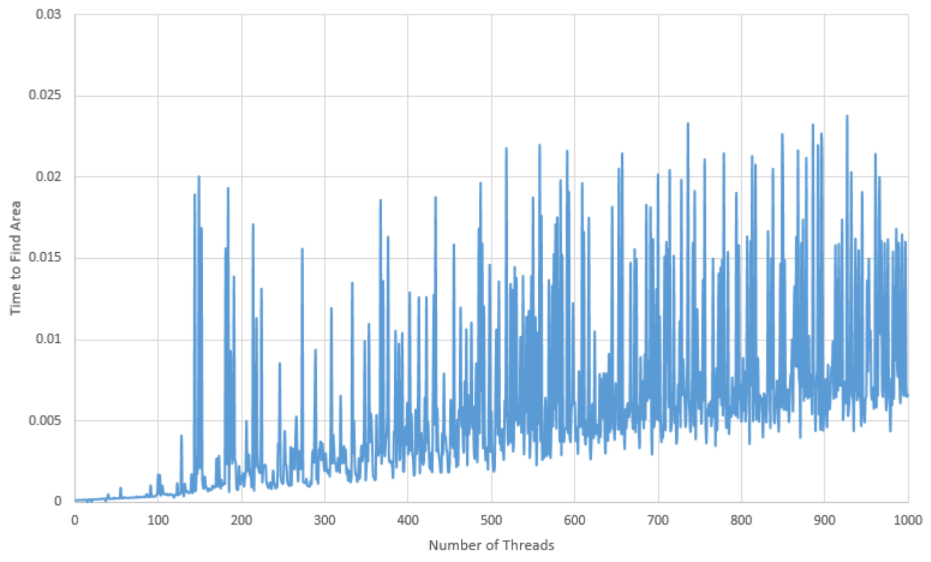

Figure 4: Time taken to find the approximated area as the number of threads increased, with 500 partitions

As the number of threads is increased beyond the amount of processing elements in the system, scheduling will occur which will increase the time taken to find the area.

3 Monte Carlo Integration

This algorithm will generate random points in a bounded box around the function and count how many points fall under the graph. The ratio of these points to the total number of points multiplied by the area of the box will give an approximation of the area under the graph. Shown below are the graphs of the accuracy and time of the algorithm.

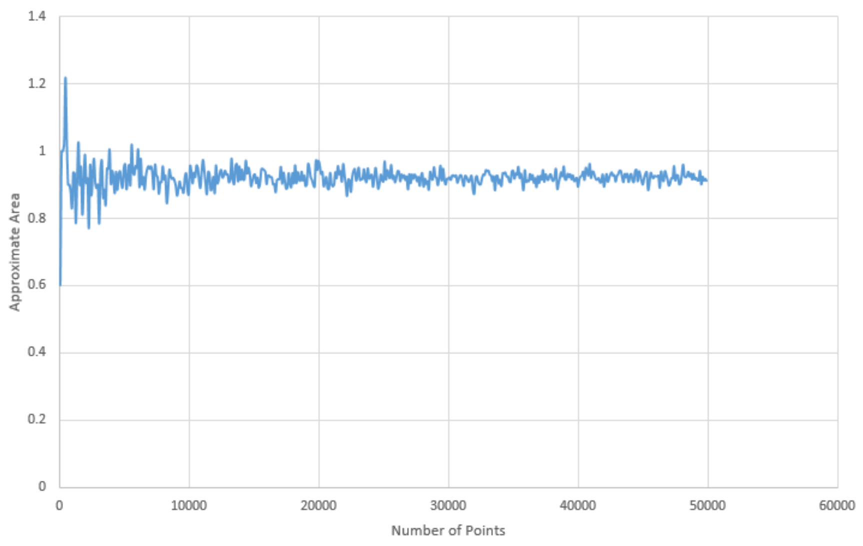

Figure 5: Accuracy of the approximated area as the number of points increased, with 4 threads

As the number of points is increased the approximated area becomes closer to the real value. The average value of the areas is 0.93 which is slightly more accurate than the trapezoidal integration.

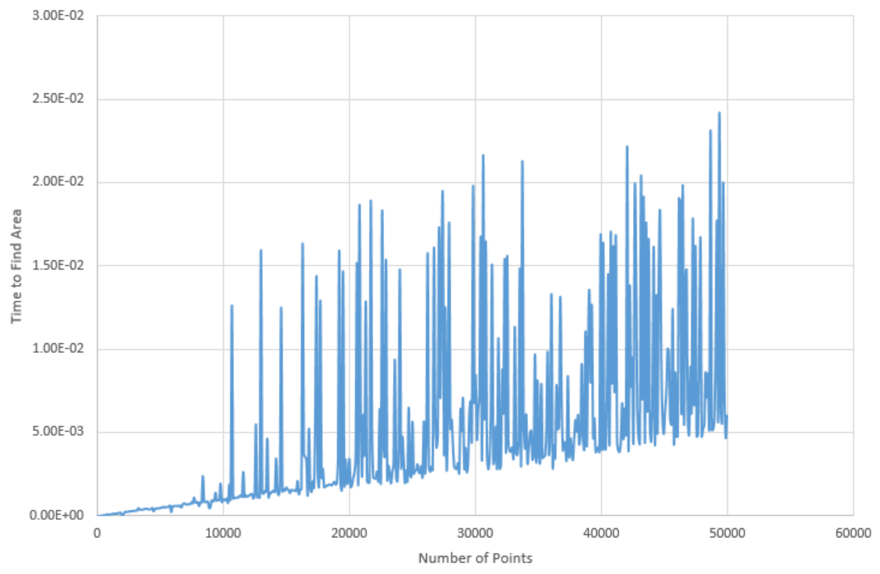

Figure 6: Time taken to find the approximated area as the number of points increased, with 4 threads

As the amount of points is increased the time taken to find the area increases proportionally at a linear rate.

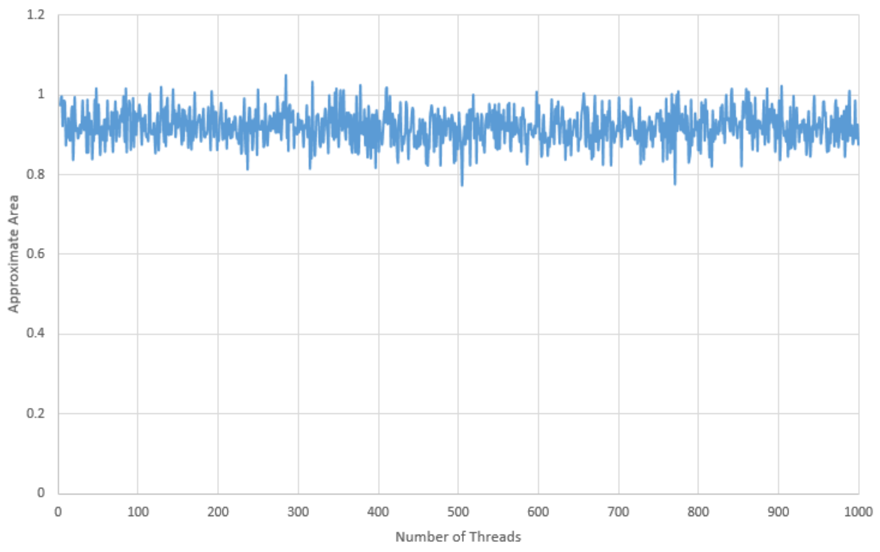

Figure 7: Accuracy of the approximated area as the number of threads increased, with 5000 points

The accuracy of the area as the number of threads increased remained fairly consistent with some slight deviations.

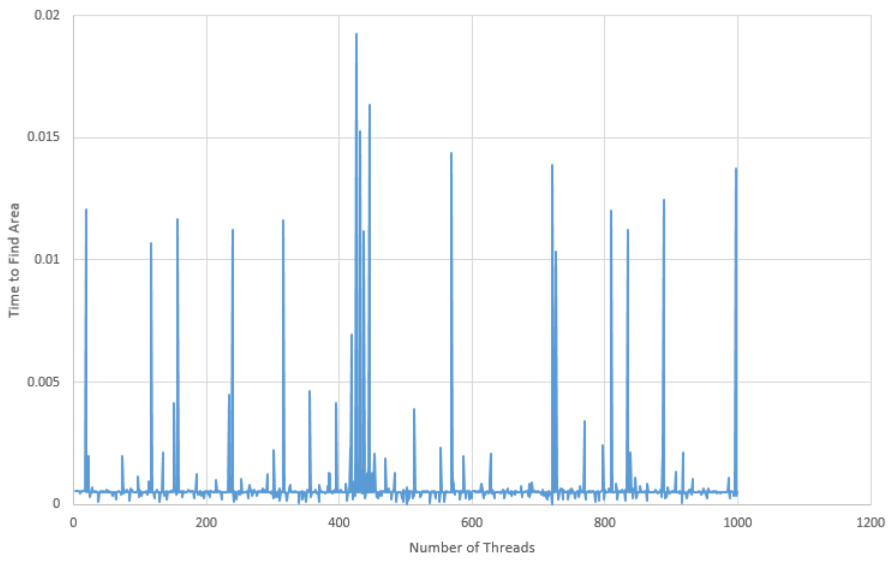

Figure 8: Time taken to find the approximated area as the number of threads increased, with 5000 points

The time taken to find the area as the number of threads increased remains very low. This is because there is not much computation for each thread to do.