1 Results

1.1 Gauss-Legendre

Solving our integral with Legandre polynomials gives unstable results for $N \in [-5, 5]$ as seen in the table below. Though with a carefull choise of N = 27 and integration limits a = -2.9 and b = 2.9 our results are precice with 4 leading digits after the decimal point.

Legandre		
N	Approximate integral	Error
11	0.297447	0.104681
15	0.315863	0.123098
21	0.268075	0.075310
25	0.240135	0.047370
27	0.229623	0.036858
27*	0.192725	0.000039

Table 1: Values for the integral for different N. *: Special case with a = -2.9 and b = 2.90.

1.2 Gauss-Laguerre

Improving our algorithm using Legandre polynomials for angles and Laguerre polynomials for radial parts improved accuracy and stability of our results. An increase in $N \in [-5, 5]$ from N = 11 to N = 15 also gives an increase in precistion, tough for and higer increase the accuracy decrese slightly, which is shown in Table 2.

Laguerre		
N	Approximate integral	Error
11	0.183021	0.009743
15	0.193285	0.000520
21	0.194807	0.002050
25	0.194804	0.002030
27	0.194795	0.002029

Table 2: Fill me in!

1.3 Monte Carlo

1.3.1 Naïve approach

The results from our Monte Carlo integration program (main.exe), are listed in this table:

Naïve Monte Carlo		
N	Approximate integral	Error
10^5	0.16799913	?
10^6	0.14673294	?
10^7	0.21039322	?
10^8	0.1898926	?
10^9	0.19482898	?

Table 3: Results from running Monte Carlo with cartesian coordinates and integration limits $x \in [-5, 5]$ - our approximation of infinity.

1.3.2 Importance sampling

The results from our Monte Carlo integration program (main.exe), are listed in this table:

Improved Monte Carlo		
N	Approximate integral	Error
10^5	0.21375956	0.0684499
10^6	0.1898093	0.0414734
10^7	0.19337239	0.00909323
10^8	0.20778867	0.00601385
10^9	0.21422913	0.00256967

Table 4: Results from running Monte Carlo with importance sampling along the exponential distribution and using spherical coordinates.

1.4 Paralellization

Our paralellization results was achieved using a quad core Intel Core i5-8250U processor with 6MB cache at 1.6GHz base clock, which boosted to 3.4GHz during testing. Thermal throttling was avoided. The memory was 4GB 1866MHz LPDDR3 soldered on board. See table 5

We also ran this test on an octa-core processor with memory of 8GB 1866MHz, and achieved no noticable speedup compared to the abovementioned computer. See table 6

For runtime imputs the number of samples was set to 10^8, with an approximation of infity of $\lambda = 5$.

Compile flags	-O3 -fopenMP	-O3	-fopenmp	no optimzation
Naive MC	12s	31s	71s	173s
Improved MC	15s	38s	79s	200s

Table 5: Shows the time spent on the same calculations with different compile parameters on a quad core processor.($N = 10^8, \lambda = 5$)

Compile flags	-O3 -fopenMP
Naive MC	12s
Improved MC	15s

Table 6: Shows the time spent on the Monte-Carlo calculations on an octa-core system.($N = 10^8, \lambda = 5$)