Computational Physics, Project 3

Volume of a hypersphere

Ena Salihovic

Jacobs University Bremen

Due March 18th, 2022

1 Rectangular Approximation

The function for rectangular approximation takes dimension N and the number of distinct integration points n_p (in file 1.py marked as n) as arguments. The main idea was to use theta function $\theta(x)$ in order to check which fraction of the hypercube was occupied by hypersphere. This checking if the hypersphere is contained within the hypercube is essentially done for every smaller hypercube into which the integration is divided.

The simulation was ran for values of $N \in \{1, ..., 6\}$ for $n_p = 20$. The following graph was obtained displaying the relationship of volume as a function of dimension.

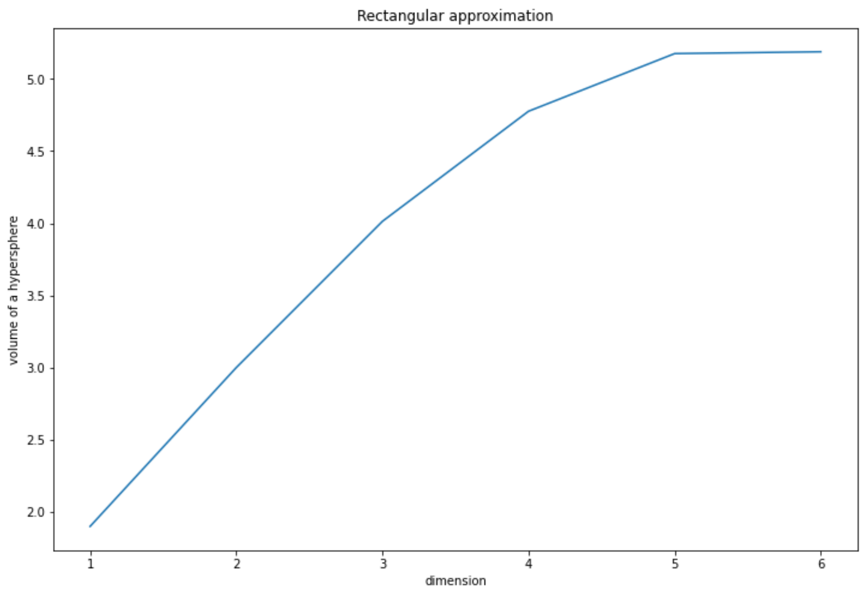

Figure 1: Plot of volume of hypersphere as a function of dimension for dimensions up to 6 using rectangular approximation

When the number of dimensions was increased further, the number of integration points n_p had to be reduced, otherwise the following error was obtained: "Unable to allocate x GiB for an array with shape (n, n, n, n, n, n, n) and data type float64", indicating that there was not computational power for these parameters.

Thus the number of dimensions was increased to $N \in \{1, ..., 10\}$ and integration points were decreased to $n_p = 5$. The following graph was obtained:

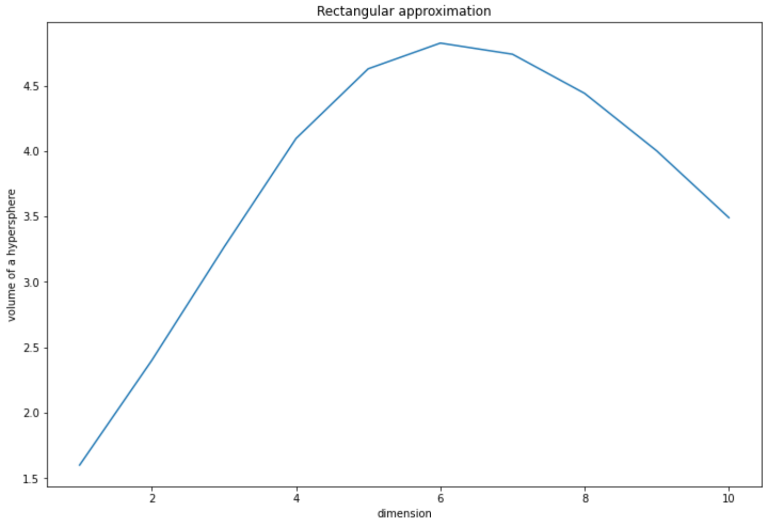

Figure 2: Plot of volume of hypersphere as a function of dimension for dimensions up to 10, using rectangular approximation

By comparing the two graphs it can be observed that the approximation of volume for the first 6 dimensions decreased, which indicates the decrease of the accuracy of this approximation, as the number of integration points is reduced. However, the second graph indicated the decrease of volume for dimensions after $N = 7$.

2 Monte Carlo simulation

Similar to the first part, the volume was calculated again using the same number of integration points n_p. What is different about the Monte Carlo approach is that the function randomly samples n_p positions r from the integration volume. Thus the function *random.uniform()* was used to sample random numbers on a uniform interval between [0, 1].
Furthermore, the approximation was ran 100 times in order to get average estimate and a standard error. Th results found are summarized in Table ().

N	V	ΔV
2	2	0
3	3.1390	0.00547785
4	4.17528	0.01369407
5	4.9560	0.02208772
6	5.26432	0.03888580
7	5.16928	0.06068066
8	4.82304	0.07120651
9	3.93216	0.09841254
10	3.46624	0.14127148
11	2.52928	0.15706391
12	2.06848	0.19634018
13	1.26976	0.20743101
14	0.90112	0.28270223
15	0.65536	0.32267695

Table 1: Table showing approximation of volume, including the standard error, for different dimensions, using Monte Carlo simulation

The values of volume were then plotted as a function of dimension and the following graph was obtained:

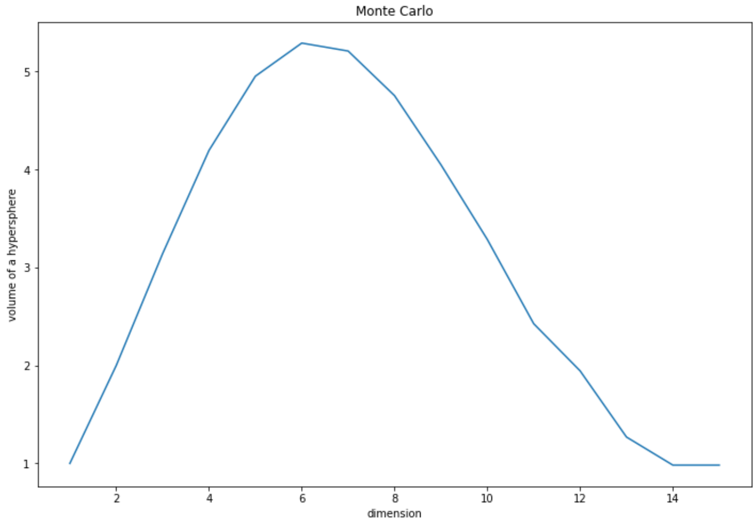

Figure 3: Plot of volume of hypersphere as a function of dimension for dimensions up to 15, using Monte Carlo approximation

It can be seen that the shape of the graph for dimensions up to 10 is similar as the graph using rectangular approximation, and that after $N = 10$ it continues to decrease, until it converges at $N = 15$.

3 Recursive function

Here, the exact values of volume was found for dimension up to $N = 15$, using the analytical solution:

$$V_N = \frac{2 \cdot \pi}{N} \cdot V_{N-2} \tag{1}$$

where $V_0 = 1$ and $V_1 = 2$.

Thus the recursive function which computes this relation was implemented and the obtained volumes are summarized in Table ()

N	V
1	2
2	3.141592653589790
3	4.188790204786390
4	4.934802200544680
5	5.263789013914320
6	5.167712780049970
7	4.724765970331400
8	4.058712126416770
9	3.298508902738710
10	2.550164039877340
11	1.884103879389900
12	1.335262768854590
13	0.910628754783283
14	0.599264529320792
15	0.381443280823304

Table 2: Table showing exact volume obtained for different dimensions, from the recursive function

4 Error as a function of integration points - Rectangle Approximation

In this part of the project, the volumes were obtained using rectangular approximation. The value of error was calculated by finding the difference between the approximated volume and the exact value, found using the recursive formula. These errors were stored in array called err (See 4.py). The value of error was plotted as a function of number of integration points on a log-log plot.

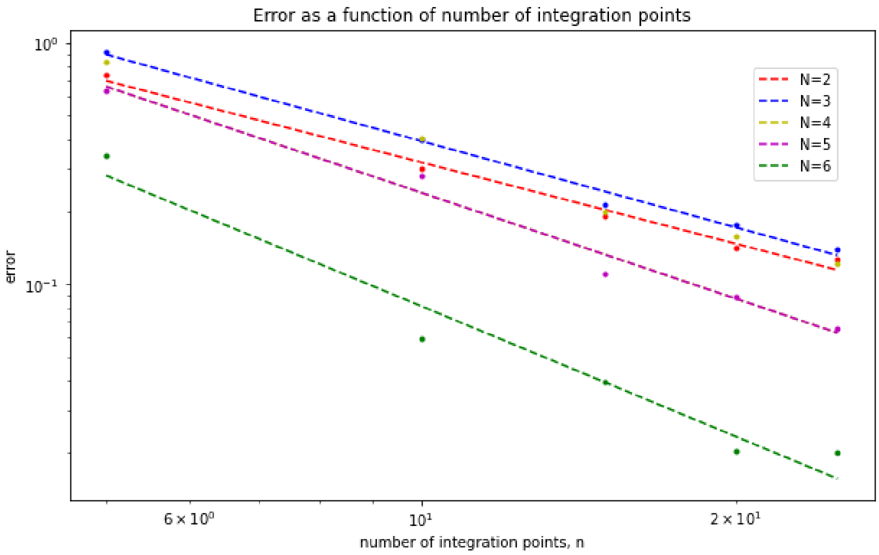

Figure 4: Plot of error as a function of number of integration points for different dimensions, using rectangular approximation

A straight line was fit to the curves, and the following equations of lines were obtained:

$$
\begin{aligned}
N &= 2 : y = -1.122520x + (1.446900)\\
N &= 3 : y = -1.190912x + (1.807496)\\
N &= 4 : y = -1.225865x + (1.820421)\\
N &= 5 : y = -1.461414x + (1.937807)\\
N &= 6 : y = -1.798506x + (1.630767)
\end{aligned}
$$

where the value of the slope indicates the power. Furthermore, a straight line on a log-log plot indicates that there is a power relationship between error and number of integration points.

The main idea was to span several orders of magnitude for number of integration points n, but this was not done due to the error that is obtained if the number of integration points exceeds a certain value (order of 10^1). Nonetheless, from the graph it can be observed that the error tends to decrease exponentially as the number of integration points increases.

5 Variance as a function of integration points - Monte Carlo Approximation

Lastly, for each combination of dimension N and number of integration points n, Monte Carlo simulation was performed to approximate the values of volume. For each combination of N and n, the Monte-Carlo approximation

was done 20 times to get a sample.
The sample variance was then calculated for the data obtained, using function np.var(). A graph of sample variance as a function of number of integration point n, was made on a log-log plot for each value of dimension N, and it can be seen in Figure ().

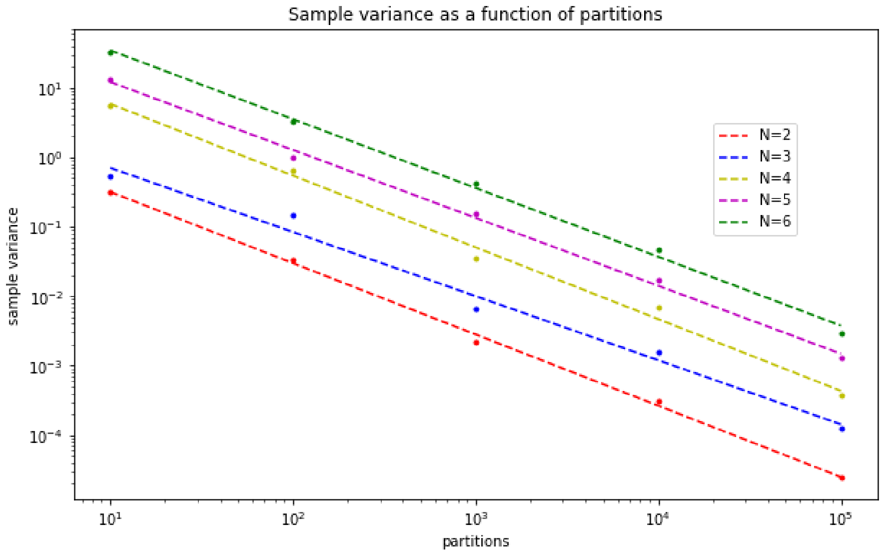

Figure 5: Plot of sample variance as a function of number of integration points for different dimensions, using Monte Carlo approximation

A straight line was fit to the curves, and the following equations were obtained:

$$N = 2 : y = -1.027063x + (1.220307)$$
$$N = 3 : y = -0.924082x + (1.780577)$$
$$N = 4 : y = -1.034232x + (4.159711)$$
$$N = 5 : y = -0.978977x + (4.756903)$$
$$N = 6 : y = -0.991721x + (5.836590)$$

where the value of the slope indicates the power. Furthermore, a straight line on a log-log plot indicates that there is a power relationship between sample variance and number of integration points. From the graph it can be concluded that the sample variance tends to decrease as the number of integration points increases.