Numerical Solutions to Multivariable Integrals using the Monte-Carlo Method

Zain Ul Abideen[1*]

[1]Department of Space Science, Institute of Space Technology.
[*]Corresponding author. Email: dominuszain@gmail.com

Abstract

This study details the procedure for solving integrals using the monte-carlo method. The method was generalized for higher dimensional integrals by using the multivariable form of the 'Mean Value Theorem'. All the relevant mathematical formalism has been included in this document. The solutions were implemented with the Scilab language.

1 Introduction

Solutions to many physical problems require solving complex and often times multivariable integrals. The analytical solutions are extremely time consuming assuming they exist in the first place, thus numerical solutions are preferred. The standard numerical techniques rely on approximating the integral with a reimann sum and using a step to approximate the solution. This method is convenient if the integrand is a function of a single variable but quickly become unviable as the dependency of the integrand increases to more variables. In those scenerios, the monte-carlo method shines. The choice of the Scilab language for the solutions was made because of its friendliness with arrays, and its rich collection of built-in functions and commitment to being open-source.

2 Results and Discussions

The effect of any function, be it single or multivariable, in a limited domain can be approximated with an array. The mean of that array will have the following trivial form:

$$\langle f \rangle = \frac{\Sigma F_i}{N} \tag{1}$$

Where N is the total number of points in the function array. The so called 'Mean Value Theorem' is another source of knowledge for the mean of a function. It conveys the following information:

$$\langle f \rangle = \frac{\int_a^b f(x)dx}{b-a} \tag{2}$$

Note that for multivariable functions, the form of the 'Mean Value Theorem' will change to incorporate the multiple integrals as follows:

$$\langle f \rangle = \frac{\int_a^b \int_c^d f(x,y)dxdy}{(b-a)(d-c)} \tag{3}$$

And so on. The theorem generalizes in a very straight forward way. For now, let us only concern ourselves with the single variable form of the theorem. We can simply equate both sides and make the integral the subject to get the final form:

$$\int_a^b f(x)dx = \frac{b-a}{N} \Sigma F_i \tag{4}$$

the variables a and b will hold constants for any particular integral. The decision for N is up to us, the higher it is the more accurate our results will be. This comes from the law of large numbers from statistics which says that the larger the sample size will be, the closer it would approximate the actual population. ΣF_i will be an array that will hold the value of the function, be it single or multivariable, at the randomly chosen N points. Simply evaluating the right hand side would give the aproximation to the integral.

The form for a tripple integral can be found using the respective generalization of the 'Mean Value Theorem'. It would have the following form:

$$\int_a^b \int_c^d \int_e^f g(x,y,z)dxdydz = \frac{(b-a)(d-c)(f-e)}{N} \Sigma G_i \tag{5}$$

All the results were verified with the results from Wolfram Alpha. The integrals did approximately converge with increasing the value for N. However, the solutions were always less accurate than the ones obtained by other deterministic methods. Another thing to notice was that the solutions would always change on running the code again. The optimal solution was found to be the running of the code an n number of times, and then making a histogram to make an educated guess about the actual value of the integral solution.