Numerical Analysis 1 – Class 12

Thursday, April 8th, 2021

Subjects covered

- Monte Carlo methods for integration in high dimensions.
- Quasi-Monte Carlo methods for sampling and integration.
- The curse of dimensionality.

Reading

- “Introduction to Monte Carlo methods”, S. Weinzierl (linked on Canvas).
- “Computational Investigation of Low-Discrepancy Sequences in Simulation Algorithms for Bayesian Networks”, J. Cheng & M. J. Druzdzel (linked on Canvas).
- “The curse of dimensionality”, M. Koppen (on Canvas).

Problems

Problem 1

In class we met the gamma function, which may be regarded as a generalization of the factorial function to non-integer inputs. The gamma function is typically defined via the integral

$$\Gamma(z)=\int_0^\infty t^{z-1}e^{-t}\,dt$$

Matlab provides a built-in function gamma(z) which computes this function for real inputs.

In class 11 we discussed Gauss-Legendre quadrature which is used to integrate functions defined on the domain $[-1,1]$. In class mentioned other quadrature methods useful for integrating functions on other domains. Another such method is Gauss-Laguerre quadrature, which is used for integration on the domain $[0,\infty]$. Gauss-Laguerre quadrature works like this

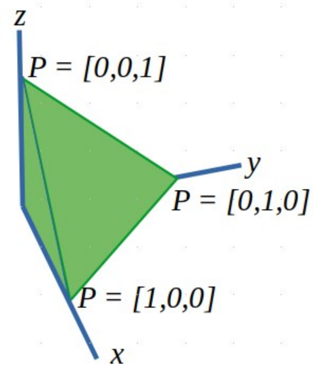
$$\int_0^\infty f(z)e^{-t}\,dt\approx\sum_{i=1}^N w_i f(z_i)$$

with sample points z_i and weights w_i given by properties of the Laguerre polynomials. Please do the following:

- Write a program implementing Gauss-Legendre quadrature for arbitrary function $f(z)$. I suggest you use the Gauss-Legendre coefficients available from the website https://keisan.casio.com/exec/system/1281279441 for your work. Please use N=32.
- Test your implementation by comparing against Matlab’s built-in function gamma(). Test your implementation over the input domain $z\in[1.0,10]$.
- How does your implementation do if you test for numbers in the domain $z\in[0.1,1.0]$?

Problem 2

In class we used the Monte Carlo method to compute the area of a circle. For this problem, you will integrate a 3-dimensional object using Monte Carlo. Please consider the “standard tetrahedron” shown at right (sometimes called the “trirectangular tetrahedron) which has one vertex at the origin, and the other three vertices laying on the x, y, and z axes. (As an aside, the nomenclature “standard tetrahedron” comes from finite-element analysis.)



Points inside the standard tetrahedron obey the following inequalities:

$$0 \le x \le 1 \quad 0 \le y \le 1 \quad 0 \le z \le 1 \quad \text{and} \quad x + y + z \le 1 \ .$$

Please do the following:

- This object is simple enough that it may be integrated analytically. The integral is

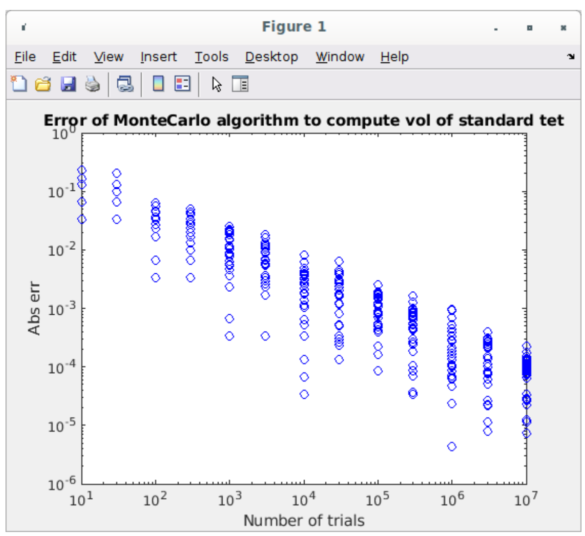
$$V = \int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} dz\, 1$$

  Please use this integral to compute the volume of the standard tetrahedron. This is a pencil-and-paper derivation. Please turn in your derivation.

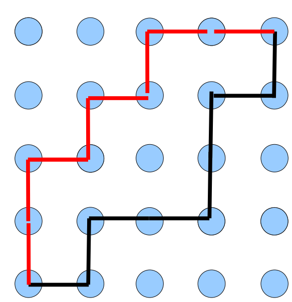
- Please write a program to compute the volume using standard Monte Carlo. Your program should accept as input the number of trials to run, and return the approximate volume.

- Please write a program which calls your Monte Carlo routine with different numbers of trials, then computes the difference between the Monte Carlo result and the exact, analytic result you computed above. Then make a plot similar to mine shown below using your program.

Feel free to use my code which computes pi using Monte Carlo as a starting point for your programs.



Problem 3



In class we talked about the “curse of dimensionality”. This problem provides another example of how complexity can explode as you scale a problem up to high dimensions.

Consider the so-called lattice path problem. The problem comes in many variants. A simple variant asks the following: You are given a 2-dimensional lattice of MxM points. The diagram at right shows a 5x5 lattice. You wish to walk from the lower left point to the upper right point. You are allowed to move up and rightward, but not left nor downward. You must also stay inside the lattice. How many legal paths exist between the lower left and upper right points? As a concrete example, two legal paths are illustrated at right: A black path and a red path.

Please do the following:

- In 2 dimensions the answer is well known – it is a binomial coefficient. Please derive the coefficient by considering the number of steps you must take in the upward and rightward directions and computing how many ways you can combine these steps. (This is a k-combination problem.)
- Now consider a 3-dimensional lattice of dimensions MxMxM. In this case the problem is, how many legal paths exist between two opposite vertices of a cube? The answer is a multinomial coefficient. Please derive the coefficient.
- Please generalize the answer to the case of the number of legal lattice paths in N dimensions for an MxMxMx … M hypercube. (To generalize the result it's enough to observe the pattern and write down the generalization … I am not asking you to derive the N-dimensional case.) The answer is still a multinomial.
- Please write a program which computes your answer found above as a function of input dimension N and lattice side M. Use your program to make a plot of the number of paths in a hyper-cube with 5 points on each side vs. the working dimension N. Please use N= 1 … 10. My plot is shown below. Note that I use a semilogy plot since the number of paths increases exponentially.

Regarding testing, I suggest you think about the simple cases of: how many paths exists in a 2D lattice of sides 2x2. How about a 3D lattices of sides 2x2?

Now consider the question, if you wanted to evaluate some function defined on these N-dimensional walks and then find its average (as is often done in computational finance), is it computationally feasible to do so by following all the walks for N=10?

By the way, another name for this phenomenon is "combinatorial explosion". The term "curse of dimensionality" is specifically used when the independent variable is the dimensionality of the problem (i.e. number of dimension you are working in).