Monte Carlo quadratures

Monte Carlo integration is a numerical quadrature where the abscissas are chosen randomly and no assumptions about smoothness of the integrand are made, not even that the integrand is continuous.

Plain Monte Carlo algorithm distributes points (in a process called "sampling") uniformly from the integration region using either uncorrelated pseudo-random or correlated quasi-random sequences of points.

Adaptive algorithms, such as VEGAS and MISER, distribute points non-uniformly, attempting to reduce integration error, using "importance" and "stratified" sampling, correspondingly.

Multi-dimensional integration

One of the problems in multi-dimensional integration is that the integration region Ω is often quite complicated, with the boundary not easily described by simple functions. However, it is usually much easier to find out whether a given point lies within the integration region or not. Therefore a popular strategy is to create an auxiliary rectangular volume V which contains the integration volume Ω and an auxiliary function F which coincides with the integrand inside the volume Ω and is equal zero outside. Then the integral of the auxiliary function over the (simple rectangular) auxiliary volume is equal the original integral.

Unfortunately, the auxiliary function is generally non-continuous at the boundary and thus the ordinary quadratures which assume continuous integrand will fail badly here while the Monte-Carlo quadratures will do just as good (or as bad) as with continuous integrand.

Plain Monte Carlo sampling

Plain Monte Carlo is a quadrature with random abscissas and equal weights ,

$$\int_V f(\mathbf{x})dV \approx w \sum_{i=1}^{N} f(\mathbf{x}_i) \ , \tag{1}$$

where $\mathbf{x}$ a point in the multi-dimensional integration space. One free parameter, w, allows one condition to be satisfied: the quadrature has to integrate exactly a constant function. This gives $w = V/N$,

$$\int_V f(\mathbf{x})dV \approx \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{x}_i) = V\langle f\rangle \ . \tag{2}$$

According to the *central limit theorem* the error estimate ϵ is close to

$$\epsilon = V\frac{\sigma}{\sqrt{N}} \ , \tag{3}$$

where σ is the variance of the sample,

$$\sigma^2 = \langle f^2\rangle - \langle f\rangle^2. \tag{4}$$

The $1/\sqrt{N}$ convergence of the error, typical for a random process, is quite slow.

Importance sampling

Suppose that the points are distributed not uniformly but with some density $\rho(x)$: the number of points Δn in the volume ΔV around point x is given as

$$\Delta n = \frac{N}{V}\rho\Delta V, \tag{5}$$

where ρ is normalised such that $\int_V \rho dV = V$.

The estimate of the integral is then given as

$$\int_V f(\mathbf{x})dV \approx \sum_{i=1}^{N} f(\mathbf{x}_i)\Delta V_i = \sum_{i=1}^{N} f(\mathbf{x}_i)\frac{V}{N\rho(\mathbf{x}_i)} = V\left\langle \frac{f}{\rho} \right\rangle \ , \tag{6}$$

Table 1: Plain Monte Carlo integrator

```
function plainmc(fun, a, b, N){
var randomx = function(a,b) //throws a random point inside inegration volume
  [a[i]+Math.random()*(b[i]-a[i]) for (i in a)];
var V=1; for(var i in a) V*=b[i]-a[i]; // V = integration volume
for(var sum=0,sum2=0,i=0;i<N;i++){ //main loop
  var f=fun(randomx(a,b)); // sampling the function
  sum+=f; sum2+=f*f} // accumulating statistics
var average =sum/N;
var variance=sum2/N-average*average;
var integral=V*average; // integral
var error=V*Math.sqrt(variance/N); // error
return [integral,error];
}//end plainmc
```

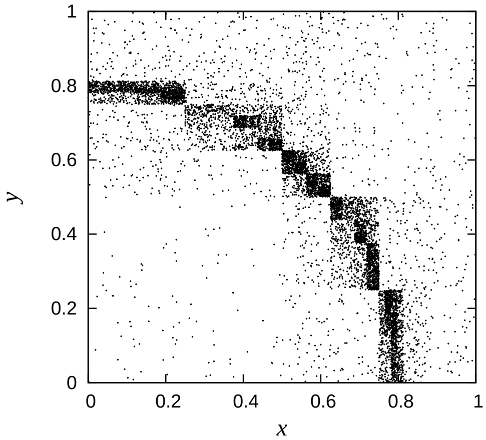

Figure 1: Stratified sample of a discontinuous function,
$f(x, y) = (x^2 + y^2 < 0.8^2)$? 1 : 0

where

$$\Delta V_i = \frac{V}{N\rho(x_i)} \tag{7}$$

is the "volume per point" at the point x_i.

The corresponding variance is now given by

$$\sigma^2 = \left\langle \left(\frac{f}{\rho}\right)^2 \right\rangle - \left\langle \frac{f}{\rho} \right\rangle^2 . \tag{8}$$

Apparently if the ratio f/ρ is close to a constant, the variance is reduced.

It is tempting to take $\rho = |f|$ and sample directly from the function to be integrated. However in practice it is typically expensive to evaluate the integrand. Therefore a better strategy is to build an approximate density in the product form, $\rho(x, y, \ldots, z) = \rho_x(x)\rho_y(y)\ldots\rho_z(z)$, and then sample from this approximate density. A popular routine of this sort is called VEGAS. The sampling from a given function can be done using the *Metropolis algorithm* which we shall not discuss here.

Stratified sampling

Stratified sampling is a generalisation of the recursive adaptive integration algorithm to random quadratures in multi-dimensional spaces.

The ordinary "dividing by two" strategy does not work for multi-dimensions as the number of sub-volumes grows way too fast to keep track of. Instead one estimates along which dimension a subdivision

Table 2: Recursive stratified sampling

```
sample N random points with plain Monte Carlo;
estimate the average and the error;
if the error is acceptable :
  return the average and the error;
else :
  for each dimension :
    subdivide the volume in two along the dimension;
    estimate the sub-variances in the two sub-volumes;
  pick the dimension with the largest sub-variance;
  subdivide the volume in two along this dimension;
  dispatch two recursive calls to each of the sub-volumes;
  estimate the grand average and grand error;
  return the grand average and grand error;
```

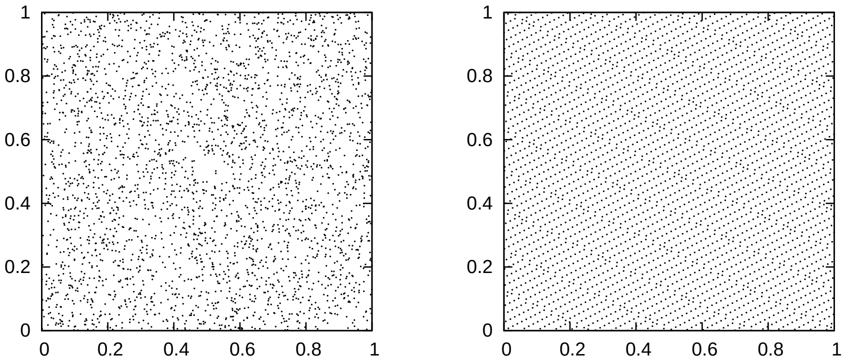

Figure 2: Typical distributions of pseudo-random (left) and quasi-random (right) points in two dimensions.

should bring the most dividends and only subdivides along this dimension. Such strategy is called *recursive stratified sampling*. A simple variant of this algorithm is given in table .

In a stratified sample the points are concentrated in the regions where the variance of the function is largest, as illustrated on figure .

Quasi-random (low-discrepancy) sampling

Pseudo-random sampling has high discrepancy[1] – it typically creates regions with hight density of points and other regions with low density of points, as illustrated on fig. 2. With pseudo-random sampling there is actually a finite probability that all the N points would fall into one half of the region and none into the other half.

Quasi-random sequences avoid this phenomenon by distributing points in a highly correlated manner with a specific requirement of low discrepancy, see fig. 2 for an example. Quasi-random sampling is like a computation on a grid where the grid constant must not be known in advance as the grid is ever gradually refined and the points are always distributed uniformly over the region. The computation can be stopped at any time.

The central limit theorem does not work in this case as the points are not statistically independent. Thus the variance can not be used as an estimate of the error.

Lattice sampling

Let α_i, $i = 1, \ldots, d$, (d is the dimension of the integration space) be a set of cleverly chosen irrational numbers, like square roots of prime numbers. Then the kth point (in the unit volume) of the sampling

[1] discrepancy is a measure of how unevenly the points are distributed over the region.

sequence will be given as

$$\mathbf{x}^{(k)} = \{\mathrm{frac}(k\alpha_1), \ldots, \mathrm{frac}(k\alpha_d)\} \ , \tag{9}$$

where $\mathrm{frac}(x)$ is the fractional part of x.

A problem with this method is that a high accuracy arithmetics (e.g. `long double`) might be needed in order to generate a reasonable amount of quasi-random numbers.