7.2 Monte Carlo integration of a Gaussian

Use a uniform random-number generator such as the function routine *ran1* of *Numerical Recipes*; make N calls to it, scaling the $(0 \rightarrow 1)$ random numbers to the range of σs required, say $k\sigma$. For each resulting value x_i, compute $f(x_i) = \frac{1}{\sigma\sqrt{2\pi}}\exp[-\frac{x_i^2}{2\sigma^2}]$. The integral from 0 to $k\sigma$ is simply

$$\frac{k}{N}\sum_{i=1}^{N} f_i(x).$$

The figure below shows the error function (erf) the integral of a Gaussian from 0 to $k\sigma$, determined in this way. It mirrors the values of Table A2.1, Practical Statistics. Here $N = 10^6$ for each integration. The figure shows the answer to part (a) - some 34% of the area occurs out to $+1\sigma$, 68% within $\pm 1\sigma$.

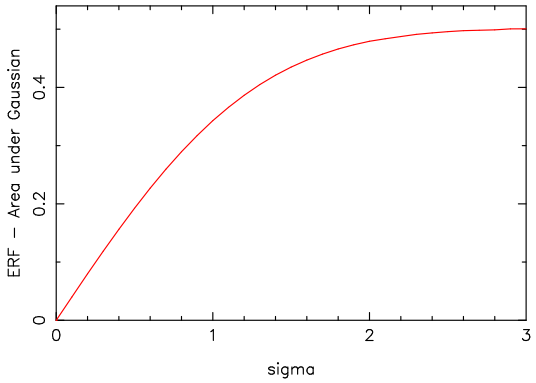

Figure 1: ERF or the error function: the Normal distribution integrated from 0 to $k\sigma$, using Monte-Carlo integration.

What price N? For this simple one-dimensional integral, it matters little, because values of N of up to 10^7 still mean perhaps only seconds of work with a modern laptop/desktop processor.

As for part (b), if we know the function has long tails, then of course the integration must encompass these. But calculating the function where it has little value is very inefficient, and we can do a *really* inefficient calculation if we take the tails grossly too large. However, to show how simple this kind of calculation is, and how little this matters in the present instance, consider a calculation of the total area under a Gaussian running from -10σ to $+10\sigma$, seriously on the safe side. The figure below shows how rapidly this mindless approach converges as N is increased.

This demonstrates that the process is stable. We have uniformly random-sampled a function running from -10 to +10, and this function is essentially a narrow spike between -1 to +1. It is surprising - if not astonishing - that a mere 10 samples provides

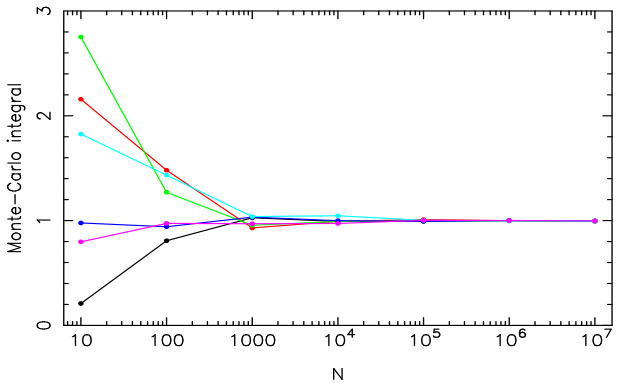

Figure 2: Monte-Carlo integration of the Gaussian, from $+10\sigma$ to -10σ using uniform random sampling across this interval. N is the number of random calls and the number of function computations. The different lines simply represent different starting seeds supplied to the random-number generator.

integration to the right order of magnitude, while just 1000 samples yields an estimate of the answer to within a few percent.

Note that this is a simple 1D case with a well-behaved function, and in this case we can afford the 10's of millions of function calculations at random - not necessarily so for each variable if the function is very multivariate, and not necessarily so if the function is badly-behaved and requiring intense sampling in some regions. *Importance sampling* has been mentioned in § 7.6. We know the function; we know where it will be meaningful to sample it, and by tailoring the sampling frequency to the function, efficiency can be drastically improved.

Starting with a simple-minded calculations like those above then, investigate a technique or techniques for *efficient* integration of a Gaussian, followed by a multivariate Gaussian, or other functions of choice. In the comparison process, rather than timing, use accuracy measurements with fixed number of samples.