CS 3200
Introduction to Scientific Computing

Instructor: Martin Berzins

Topic: Monte Carlo Methods for Quadrature

Example from Finance –
Expected Value of Collateralized Mortgage Options

$$E(PV_T) = \int_{[0,1]^{360}} \sum_{k=1}^{360} G_{k;T}(\zeta_1,\ldots,\zeta_k) u_k(\zeta_1,\ldots,\zeta_{k-1}) d\tilde{\zeta}_1 d\tilde{\zeta}_2,\ldots,d\tilde{\zeta}_{360}$$

Where

$$u_k(\zeta_1,\ldots,\zeta_{k-1}) = v_0 v_1(\zeta_1)\ldots.v_{k-1}(\zeta_1,\ldots,\zeta_{k-1})$$

$$v_j(\zeta_1,\ldots,\zeta_j) = \frac{1}{1+K_0^j i_0 e^{(\zeta_1+\ldots+\zeta_j)}}, j = 1,2,\ldots.,359$$

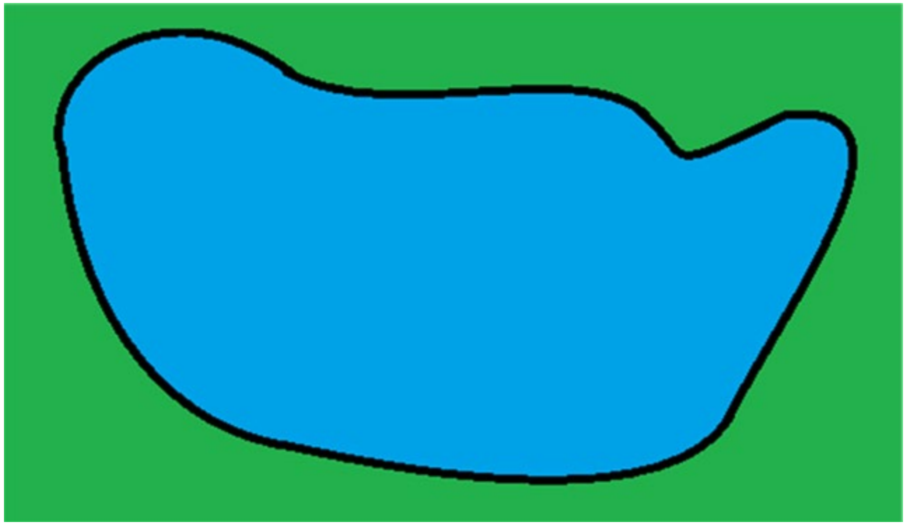
and where

$$\tilde{\zeta}_i = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\zeta_i} e^{-t^2/(2\sigma)} dt$$

This is a complex 360 dimensional integral –see
[Paskov – Computing High dimensional Integrals with Applications to Finance 1994

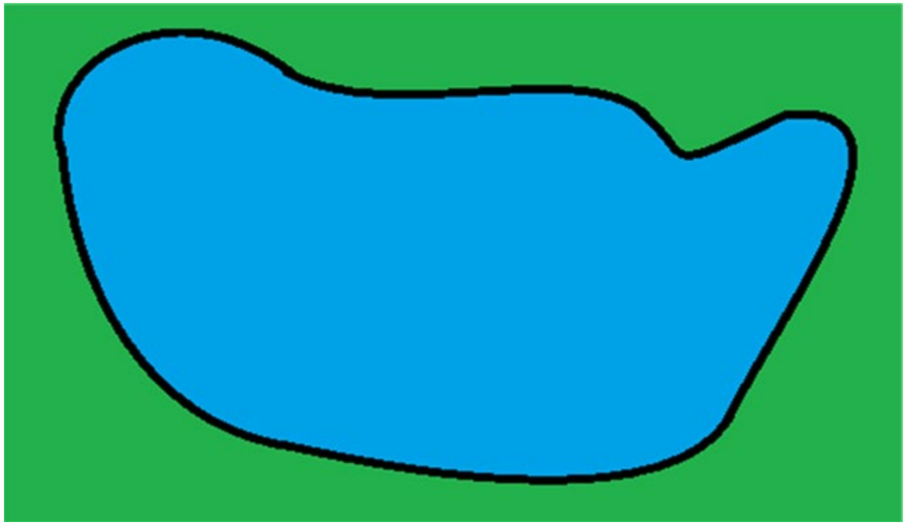
The Pond Example

- How do we find the area of the pond?



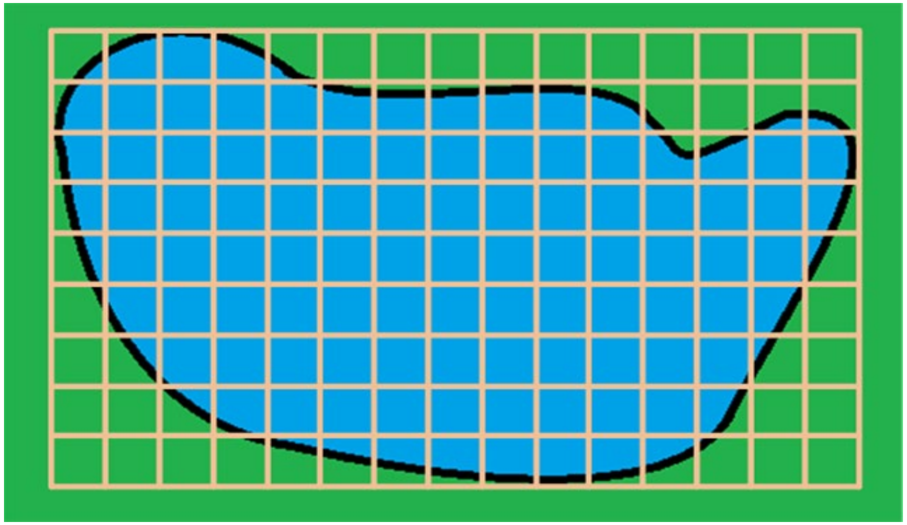
The Pond Example

- Measure perimeter?
  - Hard to calculate the area from the perimeter of an irregular shape



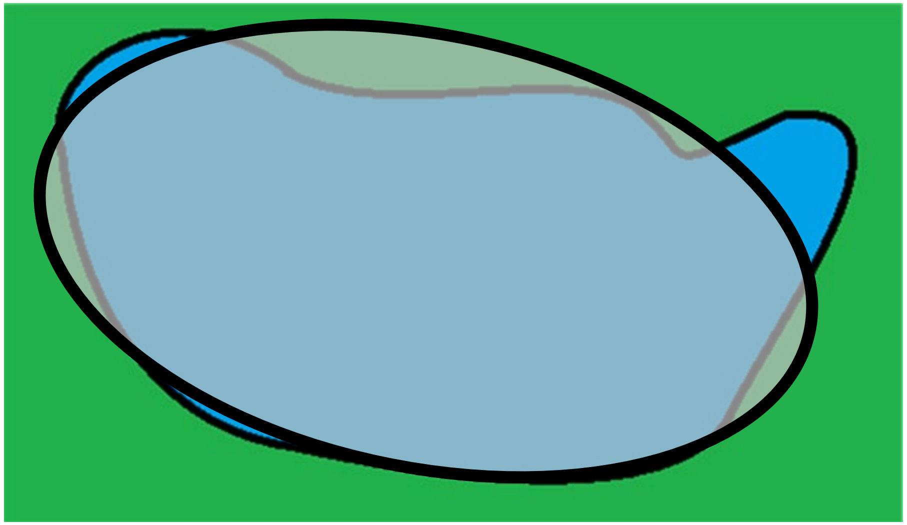
The Pond Example

- Grid-based approach?
  - Count the cells with water
  - What about partially full cells?



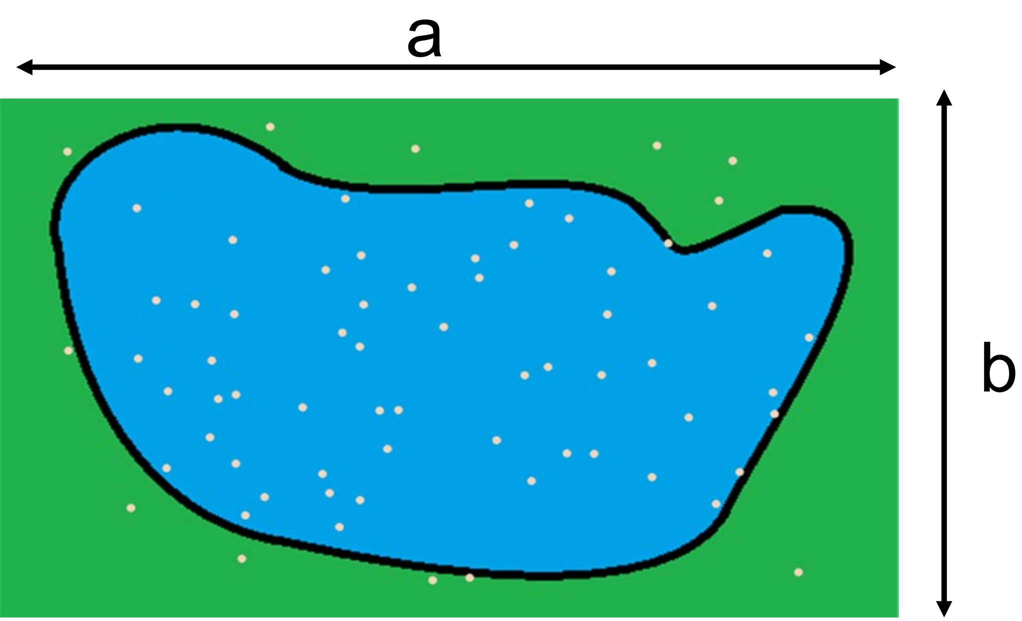
The Pond Example

- Fit a shape?
  - Difficult to fit complex shapes
  - May use composite shapes



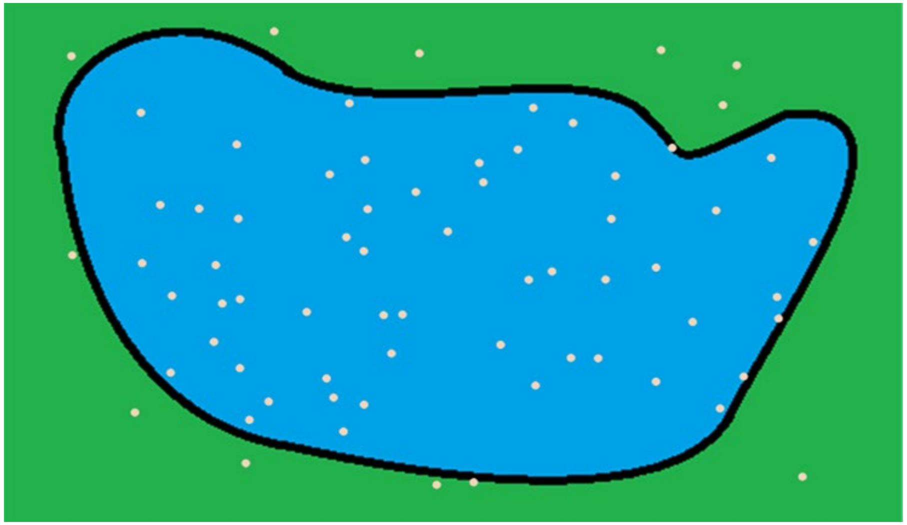
The Pond Example

- What about random points?
  - How do we approximate the area?



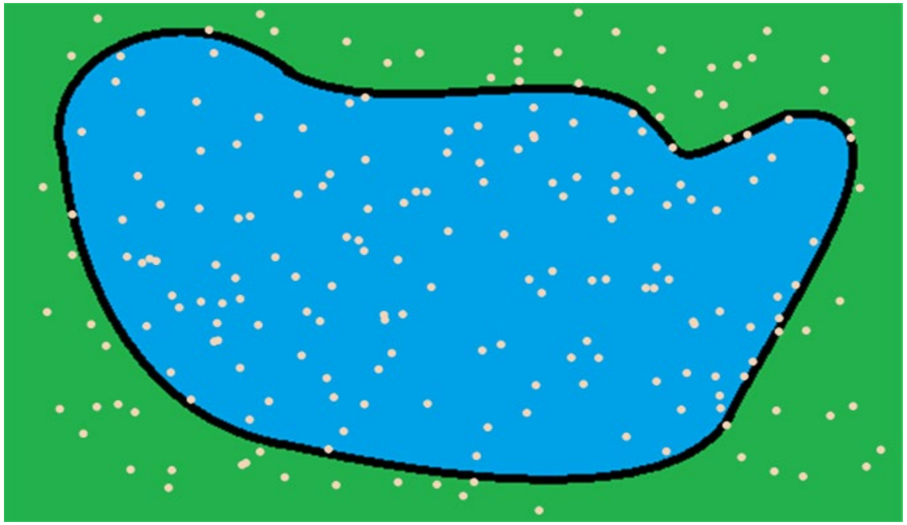
The Pond Example

- What about random points?
  - Count the points inside and outside the pond
  - Ratio of points inside to total points approximates the area
- This is an example of a *Monte Carlo approximation*



The Pond Example

- As the number of points increases, the solution becomes more accurate
  - Requires well-distributed random function
  - Trade accuracy for computation



Origins of Monte Carlo Methods

In 1946, physicists at Los Alamos Scientific laboratory were investigating radiation shielding and the distance that neutrons would likely travel through various materials and were unable to solve the problem using conventional, deterministic mathematical methods. Stanislaw Ulam had the idea of using random experiments after thinking about randomization in analyzing card games.

"I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations." Stanislaw Ulam

Monte Carlo Methods

- Approximate solutions to a variety of mathematical problems by performing statistical sampling experiments on a computer
- Stochastic method, i.e., nondeterministic
  - Usually uses random or pseudo-random numbers
- Applies both to problems with inherent probabilistic structure and to those without any probabilistic content

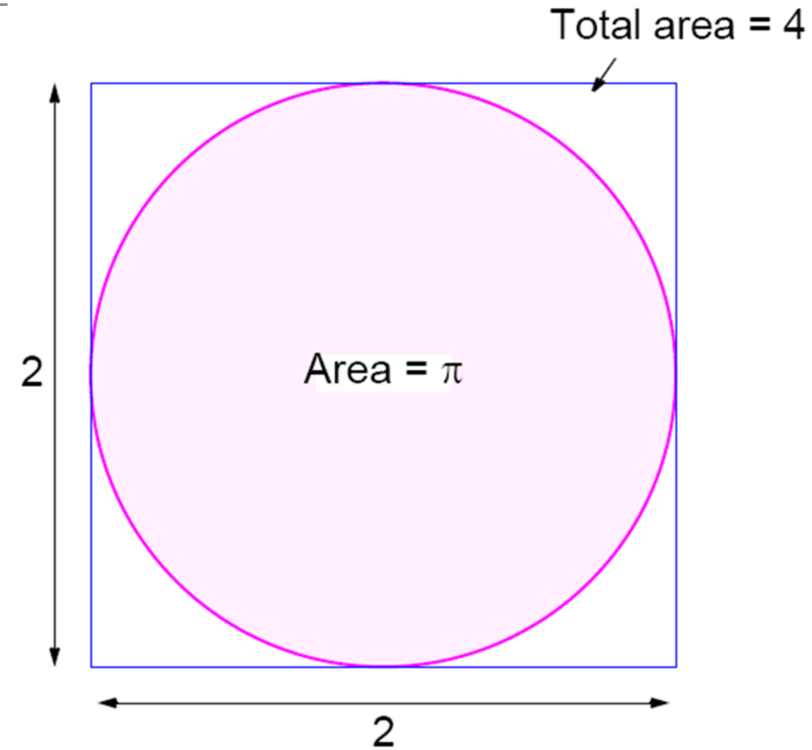
Monte Carlo Methods

- Problems commonly solved using Monte Carlo methods
  - Systems with a large number of coupled degrees of freedom
    - e.g., disordered materials, strongly coupled solids, and cellular structures
  - Phenomena with significant uncertainty in inputs
    - e.g., risk calculation in business
  - High-dimensional integrals
  - Optimization problems
    - Problems that compute equilibrium over time
  - Certain partial differential equations (PDEs)

Calculating π – A Monte Carlo Solution

Circle formed within a 2 x 2 square. Ratio of area of circle to square given by:

$$\frac{\text{Area of circle}}{\text{Area of square}} = \frac{\pi(1)^2}{2 \times 2} = \frac{\pi}{4}$$



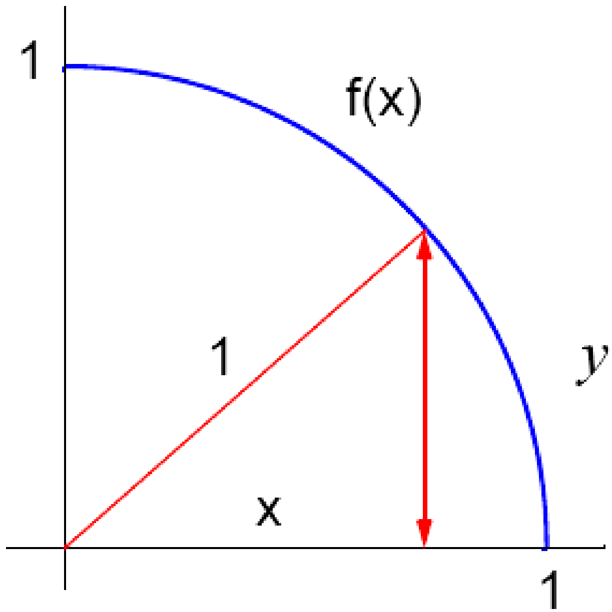
Points within square chosen randomly. Score kept of how many points happen to lie within circle.

Fraction of points within the circle will be $\pi / 4$, given a sufficient number of randomly selected samples.

Computing an Integral

One quadrant can be described by integral

$$\int_0^1 \sqrt{1 - x^2}\, dx = \frac{\pi}{4}$$



$$y = \sqrt{1 - x^2}$$

Random pairs of numbers, (xr,yr) generated, each between 0 and 1.

Counted as in circle if $y_r \leq \sqrt{1 - x_r^2}$; that is, $y_r^2 + x_r^2 \leq 1$.

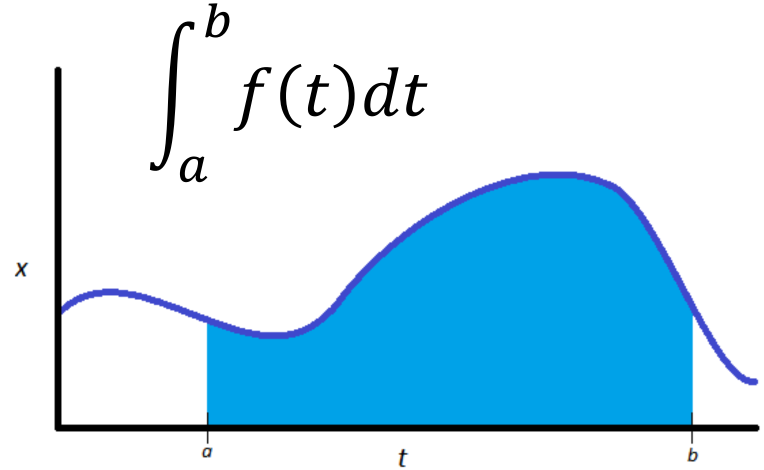
Calculating an Integral

Use random values of x to compute $f(x)$ and sum values of $f(x)$:

$$\text{Area} = \int_{x_1}^{x_2} f(x)\,dx = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} f(x_r)(\mathrm{x}_2 - \mathrm{x}_1)$$

where xr are randomly generated values of x between x_1 and x_2.

Monte Carlo method very useful if the function cannot be integrated numerically (maybe having a large number of variables)



Example

Computing the integral

$$I = \int_{x_1}^{x_2} (x^2 - 3x)\,dx$$

```
sum = 0;
for (i = 0; i < N; i++)
{     /* N random samples */
      xr = randv(x1, x2);               /* generate next random
value */
      sum = sum + xr * xr - 3 * xr;    /* compute f(xr) */
}
area = (sum / N) * (x2 - x1);
```


Routine randv(x1, x2) returns a pseudorandom number between x1 and x2.

Calculating π – A Monte Carlo Solution

- Demo
  - http://polymer.bu.edu/java/java/montepi/
- Accuracy is only $c/\sqrt{N}$ where *N* is the number of points

Example

$$\int_0^{\pi} \sin(x)\, dx = 2.0$$

n	Trapez.	Simpson	Monte Carlo
2	1.570796	2.094395	2.483686
4	1.896119	2.004560	2.570860
8	1.974232	2.000269	2.140117
16	1.993570	2.000017	1.994455
32	1.998393	2.000001	2.005999
64	1.999598	2.000000	2.089970
128	1.999900	2.000000	2.000751
256	1.999975	2.000000	2.065036
512	1.999994	2.000000	2.037365
1024	1.999998	2.000000	1.988752
2048	2.000000	2.000000	1.989458
4096	2.000000	2.000000	1.991806
8192	2.000000	2.000000	2.000583
16384	2.000000	2.000000	1.987582
32768	2.000000	2.000000	1.991398
65536	2.000000	2.000000	1.997360

Example

$$\int_0^{\pi} \frac{x}{x^2+1}\cos(10x^2)\,dx = 0.0003156$$

n	Trapez.	Simpson	Monte Carlo
64	0.004360	-0.013151	0.081207
128	0.001183	-0.001110	0.155946
256	0.000526	-0.000311	0.071404
512	0.000368	0.000006	0.002110
1024	0.000329	0.000161	-0.004525
2048	0.000319	0.000238	-0.010671
4096	0.000316	0.000277	0.000671
8192	0.000316	0.000296	-0.009300
16384	0.000316	0.000306	-0.009500
32768	0.000316	0.000311	-0.005308
65536	0.000316	0.000313	-0.000414
131072	0.000316	0.000314	0.001100
262144	0.000316	0.000315	0.001933
524288	0.000316	0.000315	0.000606
1048576	0.000316	0.000315	-0.000369
2097152	0.000316	0.000316	0.000866
4194304	0.000316	0.000316	0.000330

Multidimensional Integration

- Can be extended to n dimensions

- 2D Monte Carlo integration with N randomly chosen points

$$\int_a^b dx \int_c^d dy\, f(x, y)$$

$$\cong (b - a)(d - c)\frac{1}{N}\sum_{i=1}^{N} f(x_i, y_i)$$

7D Example

$$\int_0^1 dx_1 \int_0^1 dx_2 \int_0^1 dx_3 \int_0^1 dx_4 \int_0^1 dx_5 \int_0^1 dx_6 \int_0^1 (x_1 + x_2 + \cdots + x_7)^2 dx_7 = 12.8333333$$

	7D Integral
8	11.478669
16	12.632578
32	13.520213
64	13.542921
128	13.263171
256	13.178140
512	12.850561
1024	12.747383
2048	12.745207
4096	12.836080
8192	12.819113
16384	12.790508
32768	12.765735
65536	12.812653
131072	12.809303
262144	12.831216
524288	12.832844

Random Number Generators

- Early Matlab random number generator
- Multiplicative congruential generator [Lehmer]

$$x_{k+1} = (ax_k + c)\,\mathrm{mod}(m), x_0 = "seed"$$

$(ax_k + c)\,\mathrm{mod}(m)$ means take the remainder after division by m

Example

$a = 13, m = 31, c = 0$ gives 1,13,14,27,10,6,16,22,7,29,5,3,.....

repeats after 30 terms

Matlab used $a = 7^5 = 16807, c = 0, m = 2^{31} - 1 = 2147483677$

Mersenne Twister (MT) by Matsumoto and Nishimura

C source code freely available for a fast and efficient method.

Observing enough numbers generated by the Mersenne Twister allows all future numbers to be predicted. Hence not suitable in cryptography.

Astronomical period of $2^{19937} - 1$

623 dimensional k-equidistribution see*

Can be employed in practical important simulations because it is based on a good definition of randomness.

Many other generators are only random in a particular area.

Passed tests many stringent tests.

http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html

*http://en.wikipedia.org/wiki/Mersenne_prime

Sobol and Halton Points

Paskov shows for the 63 dimensional integral that :

Sobol and Hamilton low discrepancy points give better answers than straight Monte Carlo. These point sets are deterministic.

[Paskov – Computing High dimensional Integrals with Applications to Finance 1994]

Choice of Random Distribution of Points

A generalization of the MC approach is **importance sampling** expressed in terms of a function, *q*(*x*):

$$I = \int_x p(x)dx = \int_x \left(\frac{p(x)}{q(x)}\right) q(x)dx \quad \text{under the constraint:} \quad \int q(x)dx = 1$$

q(*x*) is the importance sampling distribution which samples *p*(*x*), non-uniformly giving more importance to some values of *p*(*x*).

Note: if q(x) is zero, then the bracketed term (p(x)/q(x)) is evaluated as zero. In that way q(x) acts as a filter.
There are lots of possible variations here.

How do we pick the distribution? We pick x and thus evaluate p(x) according to the distribution q(x) .

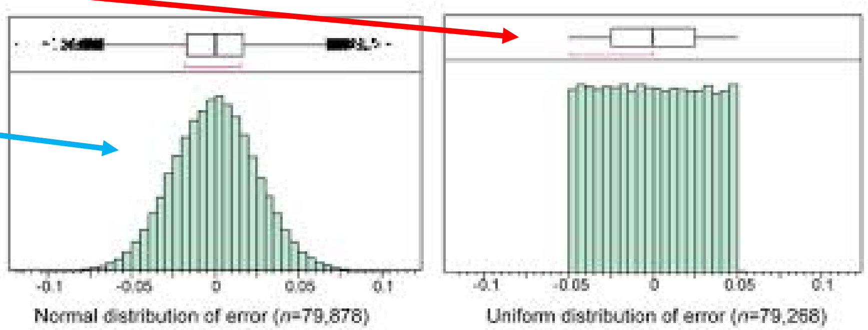
Choice of Random Distribution of Points

For example to go back to the integral

$$I = \int_x p(x)dx = \int_x \left(\frac{p(x)}{q(x)} \right) q(x)dx \quad \text{under the constraint:} \quad \int q(x)dx = 1$$

q(*x*) is the importance sampling distribution which samples *p*(*x*), non-uniformly giving more importance to some values of *p*(*x*).

If we know nothing about p(x) we may as well use a uniform distribution.
If p(x) has steep gradients in the center or in just one place then a normal distribution makes sense effectively clustering the points at the peak of p(x)



Recommended Reading

- Conceptual overview, with a finance example:
  - http://www.brighton-webs.co.uk/montecarlo/concept.asp
- Detailed treatment of the mathematical background, but from a scientific computing perspective:
  - http://www.cs.nyu.edu/courses/fall06/G22.2112-001/MonteCarlo.pdf
- Notes on theoretical and practical issues of Monte Carlo integration, by a student here at the UofU SoC:
  - http://www.cs.utah.edu/~edwards/research/mcIntegration.pdf