Monte Carlo Integration

Lecture #4: Monday, 14 Sep. 2009
Lecturer: Ravi Ramamoorthi
Scribe: Fu-Chung Huang

1 Introduction and Quadrature Methods

In rendering we have a problem to determine the intensity of a certain 3D point. This is done by gathering lighting from all directions to that point, or mathematically speaking, integrating the incoming lights in the range of a unit hemisphere around that point, which could be hard. Here we introduce a very simple but effective method called Monte Carlo method, which utilizes the power of randomness to compute the expected value for the intensity of that point.

1.1 Integration in 1D with Quadrature Methods

Before diving into complex problem domain, we first illustrate integration over certain 1D function $f(x)$. The problem is defined to find the integrated value I of function $f(x)$ over some range $x \in [a, b]$:

$$I = \int_a^b f(x)dx$$

The easiest way is to find its integral function $F(x) = \int_{-\infty}^{x} f(u)du$, and plug in the range $I = F(b) - F(a)$. However, finding an analytic form for that integral function is the biggest challenge, or even worse when the function is not continuous and the there is no such analytical form.

One way to go around this is to equally divide the range [a, b] into many small intervals, says n intervals with width $h = (b - a)/n$, which we call step size. For each interval, we evaluate the function value, and make the value represents that interval. The other way to think about it is we are calculating the area covered by the function $f(x)$. Now we can add up the contribution from each interval, and say the summation should approximate the real value I. This is called the ***Quadrature Integration***, and various rules specify how to evaluate the area under $f(x)$ to represent that interval.

Rectangle rule uses one(or two) evaluated value(s) to calculate the area, Trapezoidal rule uses two, and Simpson's rule uses three, as illustrated in Fig. 1. Overall the method has a general form looks like:

$$\hat{I} = \sum_{i=1}^{n} w_i f(x_i) \tag{1}$$

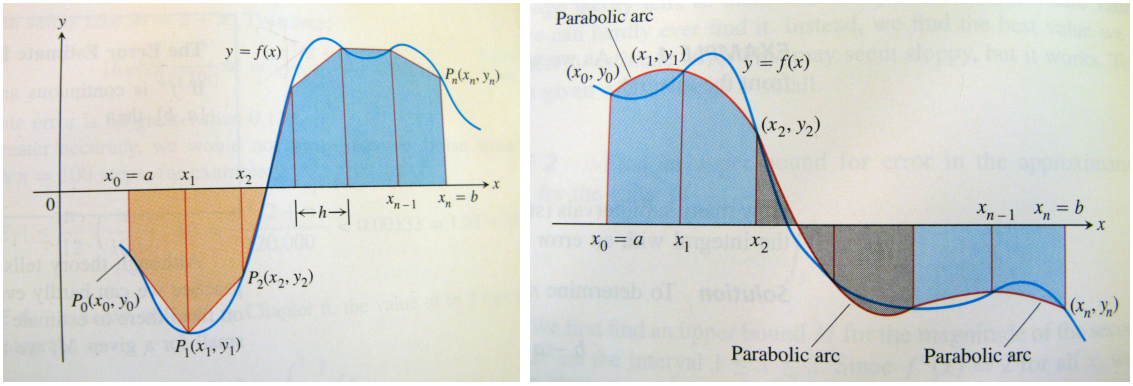

Figure 1: **Trapezoidal rule and Simpson's rule for quadrature method** The trapezoidal method uses linear approximation, and Simpson's method assumes a parabolic arc (polynomial of degree 2) to calculate the area.

Since various methods use different way to approximate area, there are still uncovered/over-covered errors. In general the error decreases as we decrease the step size(by increasing n). Providing that $f(x)$ has at least two continuous derivatives within $[a, b]$ and $|f''(x)| \leq M$, for the rectangle rule the error is given by:

$$\hat{I_R} - I \leq \frac{(b-a)}{24} h^2 M$$

and by substituting the step size(width) h with $(b-a)/n$, we have the error:

$$\hat{I_R} - I \leq \frac{(b-a)^3}{24n^2} M = O(n^{-2})$$

Trapezoidal rule has similar error bound since it also uses linear function to approximate the covered area, and its error is given by

$$\hat{I_T} - I \leq \frac{(b-a)}{12} h^2 M = \frac{(b-a)^3}{12n^2} M = O(n^{-2})$$

Simpson's rule uses a more sophisticated quadratic polynomial function(assuming $f(x)$ has at least 4-th derivatives) to calculate the covered area, and thus has better error bound:

$$\hat{I_S} - I \leq \frac{(b-a)}{180} h^4 M = \frac{(b-a)^5}{180n^4} M = O(n^{-4})$$

With 1D function the above methods can converge rapidly(or error reduces rapidly as n increases), but they do not have such a good behavior when it comes to higher dimension.

1.2 Integration in higher dimensions

As in 1D the definition Eq. 1, we can extend the approximation using tensor product rule to s-dimensional function, which is redefined:

$$\hat{I} = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} ... \sum_{i_s=1}^{n} w_1 w_2 ... w_{i_s} f(x_1, x_2, ..., x_{i_s}) \tag{2}$$

We again simply use quadrature rules to evaluate everything with n samples in each dimension, and requiring n^s samples in total. In the case of 1D function, the error bound is $O(n^{-r})$(where $r = 2$ or 4). However with s-dimensional space using $N = n^s$ samples, the error bound is $O(N^{-r}) = O(n^{-r/s})$, which degrades rapidly because of the curse of dimensionality. This means that even if we increase the number of sampled region by slicing the region more finely, the error doesn't go away as fast as we increase samples. Furthermore, if the function $f(x)$ presents discontinuities, the error bound is at best $O(n^{-1})$ for 1D case, then with higher dimensional space is at best $O(n^{-1/s})$. There is an important result which limits the convergence rate(as we have already shown) for any deterministic quadrature rule, called *Bakhalov's theorem* (which we are not going to talk about it here).

Since the error bound(or convergence rate) is not so good as quadrature method going to higher dimension, we need something better, and it should also deals with discontinuities. Monte Carlo method has the properties that error bound is always $O(n^{-1/2})$ regardless of dimensionality, and it is also immune to discontinuities.

2 Monte Carlo Method

Monte Carlo method in plain words is simply randomness. Note there is another class of random algorithm called Las Vegas, which always leads to correct result, for example *quicksort* picks random pivot. Monte Carlo method does not provide 100% correctness, but in general the expected results will be correct. Before talking how to use Monte Carlo method to integrate function, we first review some probability concepts that are useful as building block.

2.1 Probability Reviews

2.1.1 CDF and PDF

Cumulative Distribution Function (CDF) $P(x)$ for random variable X describes the probability that random variable X is less than or equal to some value x. Let's take the dice as an example with possible outcome $X = \{1, 2, 3, 4, 5, 6\}$. The CDF P(2) for rolling the dice is 1/3, since the possibility of the outcomes smaller or equal to 2 out of 6 possible faces are 1/3; similarly $P(3) = 1/2$ and $P(4) = 2/3$.

Probability Density Function (PDF) $p(x)$ is the possibility of each outcome for random variable X, defined as $dP(x)/dx$. In the dice example, the PDF $p(x)$ for the occurrence of each face is 1/6.

The possibility of cumulated outcome over some range $x \in [a, b]$ is defined as

$$P(x \in [a, b]) = \int_a^b p(x)dx = P(b) - P(a)$$

2.1.2 Expected Value and Variance

Given the definitions for CDF and PDF, we want to ask: *what is the averaged outcome in general or in the long run?* The expected value is to answer such question and defined by:

$$E_p(f(x)) = \int_\Omega f(x)p(x)dx \tag{3}$$

The expected value $E_p()$ for some function $f(x)$ is computed via drawing random sample x with some probability distribution $p(x)$. For discrete cases, the expected value is given by

$$E_p(x) = \sum_{i=1}^{n} p_i x_i$$

For the dice example, the expected value is:

$$E_p(x) = \sum_{i=1}^{n} \frac{1}{6} x_i = \frac{1}{6}(1 + 2 + 3 + 4 + 5 + 6) = 3.5$$

In addition, we also want to know the expected deviation, called variance, of the function from its expected value. Variance is an important concept to quantify error, as we will see in the next section. The variance of a function is defined by:

$$V[f(x)] = E[(f(x) - E[f(x)])^2] \tag{4}$$

Expected value and variance have some nice properties to simplify Eq. 4:

$$\begin{aligned} E[af(x)] &= aE[f(x)] \\ E\left[\sum_i f(X_i)\right] &= \sum_i E[f(X_i)] \\ V[af(x)] &= a^2 V[f(x)] \end{aligned} \tag{5}$$

So directly from Eq. 4 and rewrite $E[f(x)]$ as E_f, we have:

$$\begin{aligned} V[f(x)] &= E\left[(f(x) - E_f)^2\right] \\ &= E\left[f(x)^2 - 2f(x)E_f + E_f^2\right] \\ &= E[f(x)^2] - E\left[2f(x)E_f\right] + E\left[E_f^2\right] \\ &= E[f(x)^2] - 2E[f(x)]E_f + E_f^2 \\ &= E[f(x)^2] - E_f^2 \end{aligned} \tag{6}$$

So back to the dice example, starting from the definition Eq. 4 the variance is given by:

$$\begin{aligned}
V[f(x)] &= E[(f(x) - E[f(x)])^2] && (\ from\ Eq.\ 4) \\
&= \frac{1}{6}\left[(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (4-3.5)^2 + (5-3.5)^2 + (6-3.5)^2\right] \\
&= \frac{1}{6}\left[6.25 + 2.25 + 0.25 + 0.25 + 2.25 + 6.25\right] \\
&= 2.9167 \\
&= E[f(x)^2] - E_f^2 && (\ from\ Eq.\ 6) \\
&= \frac{1}{6}\left[1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2\right] - 3.5^2 \\
&= 15.1667 - 12.25 \\
&= 2.9167 && (\ same\ result\ )
\end{aligned}$$

Finally, given multiple independent random variable $X_1, X_2, ..., X_n$, the variance of their sum is equal to the sum of their variance:

$$V\left[\sum_{i=1}^{n} f(X_i)\right] = \sum_{i=1}^{n} V[f(X_i)] \tag{7}$$

which is useful when we later derive the error bound for Monte Carlo estimator.

2.2 Monte Carlo Estimator

So far we have seen how to use quadrature rules to deterministically evaluate an integral, and they suffer from bad error bound when dimensionality increase. In this section we introduce how to use the concept of drawing random samples with some distribution $p(x)$ to estimate the integral, and then derive its error bound.

2.2.1 Basic Monte Carlo Estimator

Remember that we want to find the integral $I = \int_a^b f(x)dx$ as defined initially. The simplest Monte Carlo estimator is very similar to the rectangular quadrature rule setting. We uniformly draw random samples from the domain $[a, b]$ of interest; rather than adding them up(since we don't actually sample the full domain), we do the averaging and scaling to properly represent the contribution of the range $[a, b]$. This in fact give us another random variable F_n, which is the averaged evaluated random variable X_i. The subscript n is used to denote that the random variable F_n also depends on how many samples we draw.

$$F_n = \frac{(b-a)}{n}\sum_{i=1}^{n} f(X_i) \tag{8}$$

Since F_n is a random variable(of the integral), we want to find its expected value, and hopefully it could approximate the real value I we want:

$$\begin{aligned}
E[F_n] &= E\left[\frac{(b-a)}{n}\sum_{i=1}^{n} f(X_i)\right] \\
&= \frac{(b-a)}{n}\sum_{i=1}^{n} E[f(X_i)], && (from\ Eq.\ 5) \\
&= \frac{(b-a)}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty} f(x)p(x)dx && (p(x) = \frac{1}{(b-a)}\ for\ U[a,b]) \\
&= \frac{(b-a)}{n}\frac{1}{(b-a)}\sum_{i=1}^{n}\int_{a}^{b} f(x)dx \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{a}^{b} f(x)dx && \left(\sum_{i=1}^{n}\int_{a}^{b} f(x)dx = nI\right) \\
&= I
\end{aligned} \tag{9}$$

The result is exactly what we want. In fact we are not restricted to use uniform sampling, but arbitrary distribution $p(x)$ on interval $[a, b]$. From Eq. 8 we have:

$$F'_n = \frac{(b-a)}{n}\sum_{i=1}^{n} f(X_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{\frac{1}{(b-a)}} = \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)} \tag{10}$$

The new random variable F'_n basically says if we draw more samples somewhere in the domain, then their weights should be scaled down. Conversely if we draw very few samples in other larger places, then their weights should be scaled up(though counter-intuitively, this is to properly account for the area they actually represent.) This mechanism also give us the ability to do importance sampling later. So for now, what is the expected value for the new random variable F'_n?

$$\begin{aligned}
E[F'_n] &= E\left[\frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{f(X_i)}{p(X_i)}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\frac{f(x)}{p(x)}p(x)dx \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{a}^{b} f(x)dx \\
&= I
\end{aligned} \tag{11}$$

2.2.2 Convergence of Monte Carlo Estimator

The most important feature that we want to use Monte Carlo estimator is that: its error bound is independent of the dimensionality. Remember in Eq. 10 we scale the evaluated

random variable $f(X_i)$ by its probability of occurrence $p(X_i)$. Here we replace them with a new random variable notation $Y_i = \frac{f(X_i)}{p(X_i)}$, and re-write:

$$F'_n = \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{p(X_i)} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{12}$$

Also remember that the expected value for F'_n is $E[F'_n] = I$. The variance for F'_n is given by:

$$\begin{aligned} V[F'_n] &= V\left[\frac{1}{n}\sum_{i=1}^{n} Y_i\right] \\ &= \frac{1}{n^2} V\left[\sum_{i=1}^{n} Y_i\right] \qquad (from\ Eq.\ 5) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} V[Y_i] \qquad (from\ Eq.\ 7) \\ &= \frac{1}{n^2} nV[Y] \\ &= \frac{1}{n} V[Y] \end{aligned} \tag{13}$$

Now given the expected value and variance for random variable F'_n, to bound the error, we can use ***Chebychev's Inequality***, basically saying that for random variable X no more than $1/k^2$ of the values are more than k standard deviations away from the mean:

$$\begin{aligned} Pr\{|X - \mu| \geq k\sigma\} &\leq \frac{1}{k^2} \\ Pr\{|X - E[X]| \geq \left(\frac{V[X]}{\delta}\right)^{-1/2}\} &\leq \delta \end{aligned} \tag{14}$$

Now we have a nice inequality to calculate the bound. By plugging the F'_n for X into Chebychev's inequality, we have:

$$Pr\{|F'_n - I| \geq \left(\frac{1}{n}\right)^{1/2}\left(\frac{V[Y]}{\delta}\right)^{1/2}\} \leq \delta$$

We can see here for any fix δ, the error decrease as we increase the number of samples, in the rate $O(n^{-\frac{1}{2}})$.

2.2.3 Example of Monte Carlo Estimator: Solid Angle Sampling and Area Sampling

After introducing some basic idea of basic Monte Carlo estimator, we will show some real cases with rendering. From Fig. 2 Left, we want to know what is the exact estimator Y_i

for Eq. 10. Now let's consider what is going on when integrating all lighting directions ω coming to a point x. The expected value of x is given by:

$$\begin{aligned} E(x) &= \int_\Omega f(x)p(x)dx \\ &= \int_\Omega L(x,\omega)\cos\theta d\omega \end{aligned}$$

and thus $f(x)p(x) = L(x,\omega)\cos\theta$. In addition, we also sample the unit hemisphere uniformly, by setting $p(\omega) = c$, so we want to find what c is. Since the integral of probability distribution over the hemisphere should equal to 1, then we have:

$$\begin{aligned} \int_\Omega p(\omega)d\omega &= 1 \\ c\int_\Omega d\omega &= 1 \end{aligned}$$

Since we know $\int_\Omega d\omega = 2\pi$, $p(\omega) = c = \frac{1}{2\pi}$, then we know the estimator $Y_i = f(x) = L(x,\omega)\cos\theta 2\pi$.

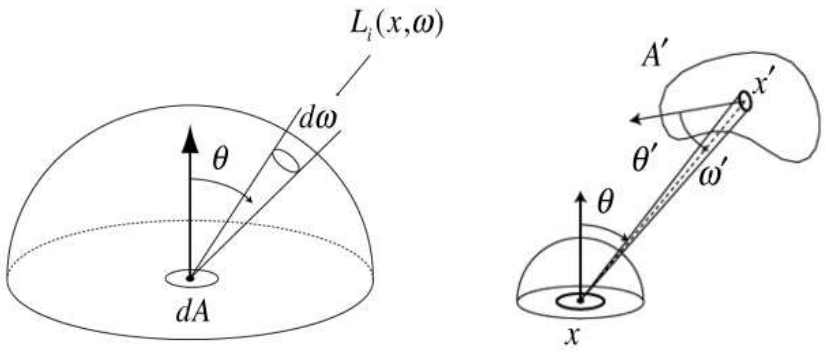

Figure 2: **Compute the estimator:** Create an appropriate Monte Carlo estimators for sampling over hemisphere or over some other object.

Now if the directions are coming from some other object with area A', as in Fig. 2 Right, then the expression for the expected value of x is given by (assuming the lighting direction ω' is defined by x and x' on the other object):

$$\begin{aligned} E(x) &= \int_\Omega L_i(x,\omega)\cos\theta d\omega \\ &= \int_{A'} L_o(x',\omega')V(x,x')\frac{\cos\theta\cos\theta'}{|x-x'|^2}dA' \end{aligned} \tag{15}$$

Again since $\int_{A'} p(u,v)dA = 1$(assuming the area is parameterized by u and v), then $p(u,v) = \frac{1}{A'}$, and the estimator $Y_i = L_o(x',\omega')V(x,x')\frac{\cos\theta\cos\theta'}{|x-x'|^2}A'$.

3 Generating Sampling Patterns

After seeing some examples that draw samples uniformly, we want to do something more efficiently. Remember in Eq. 10 we substitute arbitrary random distribution for the original uniform distribution. This gives us a lot of freedom to efficiently sample somewhere we are interested in. For example if we know some place has higher frequency information, then we can actually send more samples toward that region. In a environment map lighting situation, we can shoot more photons from brighter area light region than those darker region. In integrating light over surface hemisphere, we can trace more samples from lights perpendicular to the surface than those light from grazing angle.

3.1 Inversion Method

Assuming we have a uniform random number generator in hand, then we want to find a special random distribution that samples certain region more densely than other(also assume that we definitely know the distribution of importance $p(x)$ on the entire domain). The idea is to uniformly sample the CDF of $p(x)$, and invert the CDF back. Why is this true? Imagine that if certain region has higher importance, its contribution is higher, then it has larger final CDF area. If we uniformly sample on CDF, then we have bigger chances to hit the region, and we will in fact get more samples in the important region. This is called the inversion method, as illustrated in Fig. 3.

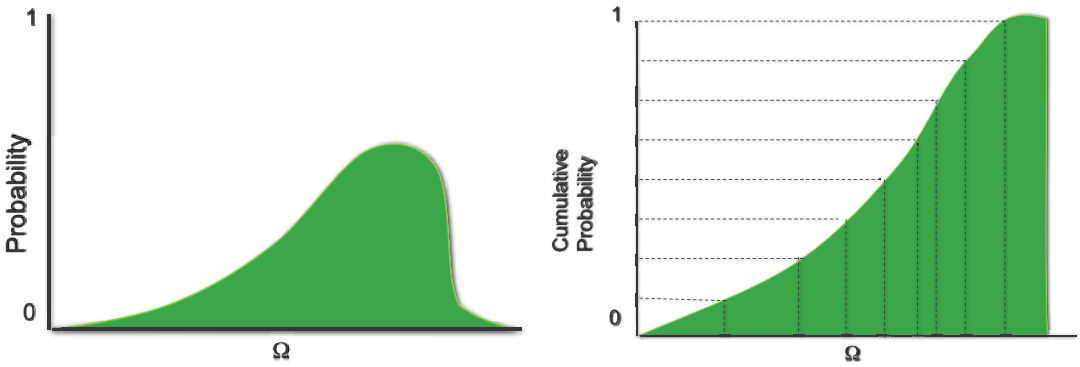

Figure 3: **Inversion Method** Left: Importance function $p(X)$ for some region Ω. Right: The CDF for $p(X)$. By sampling uniformly at the CDF, we can get dense samples at important regions in the domain Ω shown in right side of $p(x)$.

3.1.1 Example 1: Power Distribution

For example, if we know the importance function has the form of power distribution, i.e. $p(x) \propto x^n$ (which is useful in sampling Blinn micro-facet model), then the PDF of this

function is given by:

$$p(x) = cx^n$$

for some constant c. The first task is to figure out what c is, given that the integral (CDF) should equal to 1.

$$\begin{aligned} \int_0^1 cx^n &= 1 \\ c\frac{x^{n+1}}{n+1}|_0^1 &= 1 \\ \frac{c}{n+1} &= 1 \\ c &= (n+1) \end{aligned}$$

So now we know the PDF is $p(x) = (n+1)x^n$ and CDF is $P(x) = x^{n+1}$. Therefore, given uniform random variable U, the inverted importance function is $\sqrt[n+1]{U}$.

3.1.2 Example 2: Exponential Distribution

When rendering with participating media, it is useful to draw samples using exponential distribution. The function has the form $p(x) = ce^{-ax}$, and as before we want to find c given the function integrated to 1.

$$\int_0^\infty ce^{-ax} = -\frac{c}{a}e^{-ax}|_0^\infty = \frac{c}{a} = 1$$

thus we know $c = a$, PDF is ae^{-ax}, and CDF is:

$$P(x) = \int_0^x ae^{-au}du = 1 - e^{-ax}$$

Given uniform random variable U, we can find the inverted importance function:

$$\begin{aligned} U &= 1 - e^{-ax} \\ 1 - U &= e^{-ax} \\ \ln(1-U) &= -ax \\ -\frac{\ln(1-U)}{a} &= x \end{aligned}$$

Also note that since U is uniform random variable, so does $1 - U$, and we can further simplify the inversion by

$$X = P^{-1}(x) = -\frac{U'}{a}$$

3.1.3 Example 3: Sampling A Unit Disk

A unit disk is given by two parameters, $0 \leq r \leq 1$ and $0 \leq \theta \leq 2\pi$. It seems trivial that we can simply sample r and θ uniformly, but we will get the wrong answer by that. Imagine that if we fix certain range of θ and see for every equally spaced region in r, how will the areas differ, then we will immediately understand the area is proportional to the squared distance r^2, as shown in Fig. 4. The problem is since area is small around the center, then if we draw uniformly from r, then in fact we densely sample around the center, which is not what we want. The correct way is drawing samples $r = \sqrt{U_1}$ and $\theta = 2\pi U_2$ from two uniform random variables U_1 and U_2.

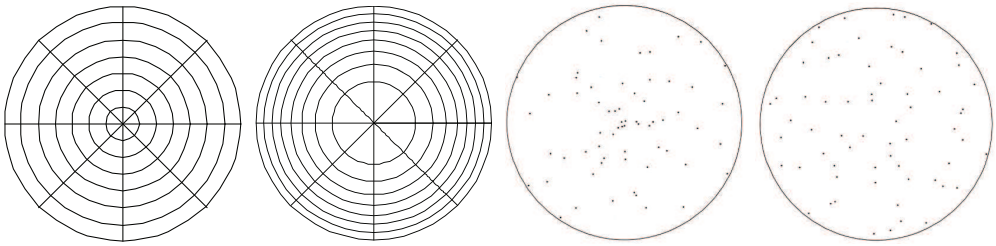

Figure 4: **Sampling a unit disk** From left to right: Area with uniform r, area with uniform $\sqrt{r}$, sampling with uniform r, sampling with uniform $\sqrt{r}$.

3.1.4 Summary

Inversion method is a powerful tool to find the correct importance sampling distribution. However it requires two things:

1. $P(x)$, the integral of $p(x)$
2. The inversion $P^{-1}(x)$

Both are in general difficult to meet, but we could still use less strong assumption to approximate the function(better than nothing!). In face of these difficulties, there is another popular method called Rejection Method to circumvent the problem.

3.2 Rejection Method

The rejection method is a technique for generating samples according to a function's distribution without needing to do either integration nor inversion; it is essentially a dart-throwing approach. The method is very simple just by drawing samples from a rectangle, and then checking if each random sample is within the desired region, as shown in two example in Fig. 5. The efficiency depends on:

$$\frac{Area\ of\ function}{Area\ of\ rectangle}$$

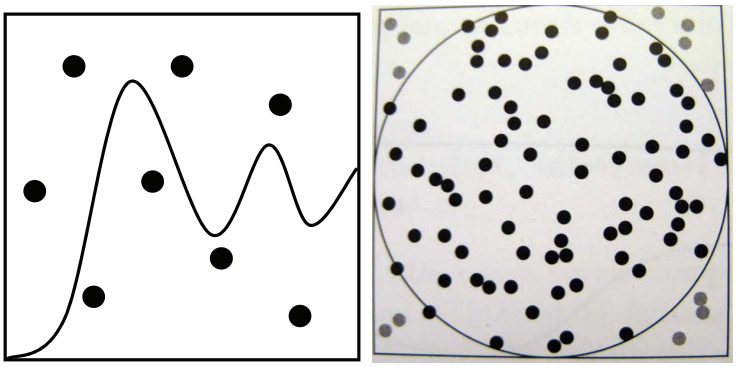

Figure 5: **Rejection Method** The method repeatedly draws random sample, and checks either to accept or to reject such sample.

For the unit disk sampling, the efficiency is $\frac{\pi r^2}{4r^2} = \frac{\pi}{4} \approx 78.5\%$. Clearly rejection isn't as efficient as inversion method, given that we know how to find a good approximation to the importance function. Nevertheless, since rejection method is simple, it can be used as a debugging tool to verify the correctness of other method.

4 Variance Reduction Technique

Monte Carlo estimator is a powerful tool to compute the integral of certain function, and its convergence is independent of dimensionality. However, the major problem is that it is slow. Variance decreases at the rate $O(\frac{1}{n})$ and error decreases at $O(n^{-\frac{1}{2}})$, which means that increasing the number of samples removes noise slowly. In this section, we introduce two important techniques that directly control the effectiveness of sampling.

4.1 Importance Sampling

Recall from Eq. 10 that we can substitute arbitrary distribution $p(x)$ for uniform distribution, and redefined in Eq. 12 that $Y_i = \frac{f(X_i)}{p(X_i)}$ to improve our sampling strategy. So the question is: what is the best probability distribution to do the important sampling? The answer is: Sample according to the function itself! Why? The idea is that by concentrating work where $f(x)$ has high value, then we can compute the estimate more efficiently. Consider the case: $p(x) \propto f(x)$ or $p(x) = cf(x)$ for some constant c, then in our definition,

$$Y_i = \frac{f(X_i)}{p(X_i)} = \frac{1}{c}$$

It directly follows that $Var[Y] = 0$, a zero variance estimator! However in practice we won't able to get a perfect probability distribution so the fallback strategy for importance sampling is put more samples where $f(x)$ is bigger(or find $p(x)$ similar to $f(x)$), as shown

in Fig. 6, and this strategy is still unbiased, as in Eq. 11. One thing to note is that if the probability distribution is chosen poorly, then it is possible to increase the variance. In practice, importance sampling is one of the most frequently used variance reduction techniques, since it is easy to apply and very effective when proper sampling distribution is used.

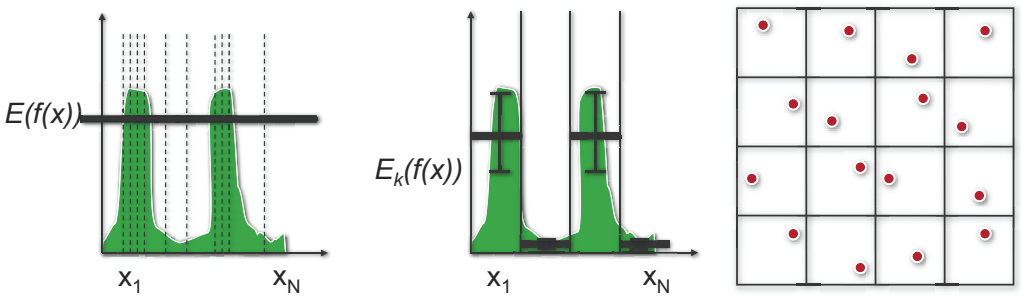

Figure 6: **Importance Sampling(Left) and Stratified Sampling Techniques(Right: 1D and 2D cases)** The two methods are used to reduce the variances we got by either sampling according to $f(x)$ or dividing regions so that variances within each region is small.

4.2 Stratified Sampling

Another popular technique is subdividing the sampling domain Ω into M non-overlapping smaller sub-domains Ω_k, which is called *stratum*, and $\bigcup_k^M \Omega_k = \Omega$. The idea is that: The function can have many discontinuities in the overall region, but if we zoom in and investigate a smaller portion, then it would be smooth, despite some regions having sharp discontinuity. If the boundary is chosen carefully, then we are able to get smaller variance within each region using fewer samples, and overall reducing the variance.

Suppose within a single stratum Ω_k we draw n_k samples, the Monte Carlo estimate is

$$F_k = \frac{1}{n_k} \sum_{j=1}^{n_k} \frac{f(X_{k,j})}{p_k(X_{k,j})}$$

where the $X_{k,j}$ is the j-th sample drawn from the k-th stratum. The overall estimate is $F_{st} = \sum_{k=1}^{M} v_k F_k$ where v_k is the volume of stratum Ω_k. For each stratum we define its mean and variance by μ_k and σ_k^2(also note that the variance from n_k samples for the basic Monte Carlo estimator F_k is $V[F_k] = \frac{\sigma_k^2}{n_k}$), then we can find the variance for the overall estimator:

$$V[F_{st}] \quad = \quad V\left[\sum_{k=1}^{M} v_k F_k\right]$$

$$
\begin{aligned}
&= \sum_{k=1}^{M} V[v_k F_k] \\
&= \sum_{k=1}^{M} v_k^2 V[F_k] \\
&= \sum_{k=1}^{M} \frac{v_k^2 \sigma_k^2}{n_k}
\end{aligned}
$$

Assuming that $n_k \propto v_k$ or $n_k = v_k n$, then the variance of the overall estimator is:

$$
\begin{aligned}
V[F_{st}] &= \sum_{k=1}^{M} \frac{v_k^2 \sigma_k^2}{n_k} \\
&= \frac{1}{n} \sum_{k=1}^{M} v_k \sigma_k^2
\end{aligned}
$$

Given the variance of stratified Monte Carlo estimator $V[F_{st}]$, we want to compare it with the estimator without stratification. From Veach's thesis(which discussed how to derive conditional variance in Eq.2.11 and Eq.2.26, note that since we draw sample $X_{j,k}$ after choosing sub-domain Ω_k, this is really a conditional probability), we have:

$$
V[F] = \frac{1}{n} \left[\sum_{k=1}^{M} v_k \sigma_k^2 + \sum_{k=1}^{M} v_k (\mu_k - I)^2 \right]
$$

What matters is the $(\mu_k - I)^2$ term, which is always non-negative. This relationship essentially tells us that if domain subdivision is taken carefully(meaning at least $\mu_k \neq I$), then stratified sampling will always give better results.