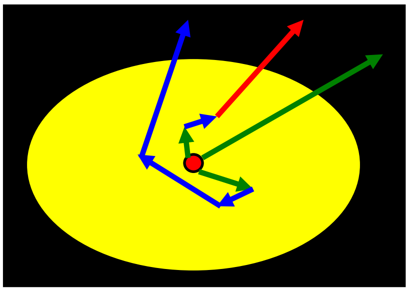
Monte Carlo Radiation Transfer I

- Monte Carlo "Photons" and interactions
- Sampling from probability distributions
- Optical depths, isotropic emission, scattering

Monte Carlo Basics

- Emit luminosity (power) packet
- Packet travels some distance
- Something happens…



- Scattering, absorption, re-emission

Luminosity Packets

Total luminosity = L (Watts, J/s, erg/s)

Each packet carries energy $E_i = L\ \Delta t\ /\ N$

N = number of Monte Carlo packets

Δt is time interval over which simulation being performed. Not computer time, but physical time and allows deposited packets to be equated to absorbed energy, for example.

Physical properties of medium do not change during Δt

MC packet represents N_γ real photons, where $N_\gamma = E_i\ /\ h\nu_i$

Luminosity Packets

Monte Carlo luminosity packet moving in direction θ contributes to the specific intensity:

$$I_\nu = \frac{dE_\nu}{\cos\theta\, dA\, dt\, d\nu\, d\Omega}$$

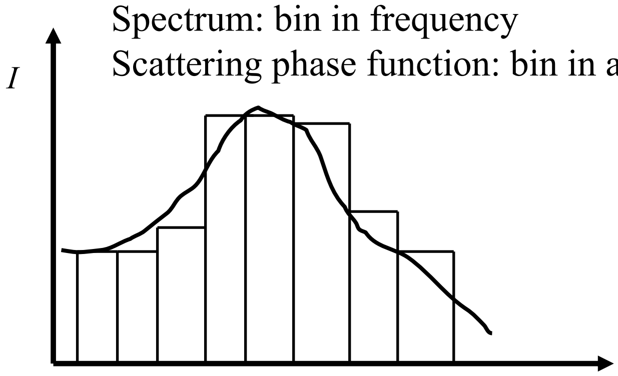
$$\Delta I_\nu = \frac{E_i}{\cos\theta\, \Delta A\, \Delta t\, \Delta\nu\, \Delta\Omega}$$

E_i → Energy of i-th packet

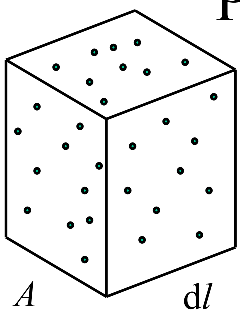
I_ν is a ***distribution function***. MC works with ***discrete*** energies. Create a histogram by binning the packets according to their directions, frequencies, etc, enables us to simulate a distribution function:

Spectrum: bin in frequency

Scattering phase function: bin in angle



Photon Interactions



Volume = A dl

Number density n (units Length^{-3})

Cross section σ (units Length^2)

Energy removed from beam per particle $/t$ / ν / dΩ= I_ν σ

Total energy absorbed/scattered from beam /sec

$$= I_\nu \, \sigma \, n \, A \, \mathrm{d}l$$

Total energy absorbed/scattered from beam /sec/area

$$= I_\nu \, \sigma \, n \, \mathrm{d}l$$

Intensity differential over dl is dI_ν = - I_ν n σ dl. Therefore

$$I_\nu (l) = I_\nu (0) \exp(-n \sigma l)$$

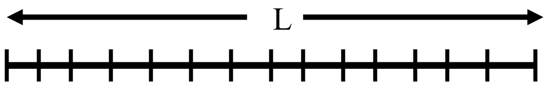
Fraction scattered or absorbed / length = n σ

n σ = volume absorption coefficient = ρ κ

Mean free path = 1/(nσ) = average dist between interactions

Probability of interaction in dl is n σ dl

Probability of traveling dl without interaction is 1 – n σ dl



N segments of length *L* / *N*

Probability of traveling L is

$$P(L) = (1 - n \sigma L / N)(1 - n \sigma L / N) \ldots$$

$$= (1 - n \sigma L / N)^N = \exp(-n \sigma L) \text{ (as N -> infty)}$$

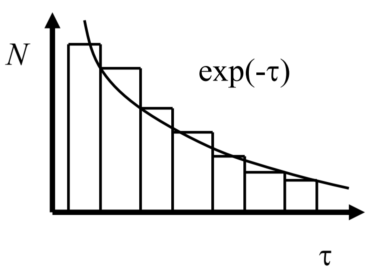
$$P(L) = \exp(-\tau)$$

$\tau = n \sigma L$ = number of mean free paths in distance L

Probability Distribution Function

PDF for photons to travel τ is exp(-τ).

If we pick τ uniformly over the range 0 to infinity we will not reproduce exp(-τ). Want to pick lots of small τ and fewer large τ. Same with a scattering phase function: want to get the correct number of photons scattered into different directions, forward and back scattering, etc.



Cumulative Distribution Function

$$\text{CDF} = \text{Area under PDF} = \int P(x)\,dx$$

Randomly choose τ, θ, λ, … so that PDF is reproduced

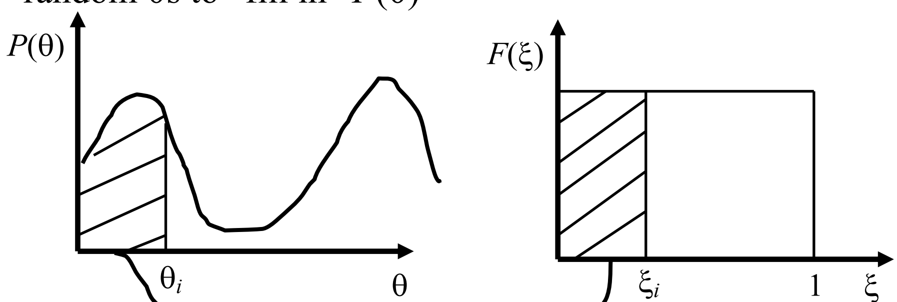
ξ is a random number uniformly chosen in range [0,1]

$$\xi = \int_a^X P(x)\,dx \Rightarrow X$$

$$\int_a^b P(x)\,dx = 1$$

This is the ***fundamental principle*** behind Monte Carlo techniques and is used to sample randomly from PDFs.

e.g., $P(\theta) = \cos\theta$ and we want to map ξ to θ. Choose random θs to "fill in" $P(\theta)$



$$\xi_i = \int_0^{\theta_i} P(\theta)\,d\theta = \sin\theta_i \Rightarrow \theta_i = \sin^{-1}\xi_i$$

Sample many random θ_i in this way and "bin" them, we will reproduce the curve $P(\theta) = \cos\theta$.

Choosing a Random Optical Depth

$P(\tau) =$ exp(-τ), i.e., photon travels τ before interaction

$$\xi = \int_0^{\tau} e^{-\tau}\, d\tau = 1 - e^{-\tau} \Rightarrow \tau = -\log(1-\xi)$$

Since ξ is in range [0,1], then (1-ξ) is also in range [0,1], so we may write:

$$\tau = -\log \xi$$

Physical distance, *L*, that the packet has traveled from:

$$\tau = \int_0^{L} n\sigma\, ds$$

Random Isotropic Direction

Solid angle is dΩ = sin θ dθ dϕ, choose (θ, ϕ) so they fill in PDFs for θ and ϕ. *P*(θ) normalized over [0, π], *P*(ϕ) normalized over [0, 2π]:

$P(\theta) = \frac{1}{2}\sin\theta$ $\qquad$ $P(\phi) = 1 / 2\pi$

Using fundamental principle from above:

$$\xi = \int_0^{\theta} P(\theta)d\theta = \frac{1}{2}\int_0^{\theta} \sin\theta\, d\theta = \frac{1}{2}(1-\cos\theta)$$

$$\xi = \int_0^{\phi} P(\phi)d\phi = \frac{1}{2\pi}\int_0^{\phi} d\phi = \frac{\phi}{2\pi}$$

$$\theta = \cos^{-1}(2\xi - 1)$$
$$\phi = 2\pi\,\xi$$

Two different ξ values!!

Use this for emitting packets isotropically from a point source, or choosing isotropic scattering direction.

Albedo

Packet gets to interaction location at randomly chosen τ, then decide whether it is scattered or absorbed. Use the ***albedo*** or ***scattering probability***. Ratio of scattering to total opacity:

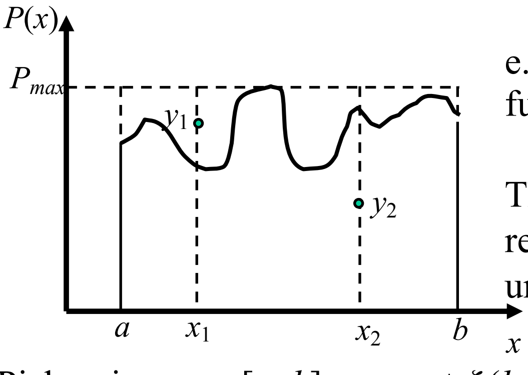
$$a = \frac{\sigma_S}{\sigma_S + \sigma_A}$$

To decide if a packet is scattered: pick a random number in range [0, 1] and scatter if $\xi < a$, otherwise packet absorbed

Now have the tools required to write a Monte Carlo radiation transfer program for isotropic scattering in a constant density slab or sphere

Rejection Method

Used when we cannot invert the PDF as in the above examples to obtain analytic formulae for random θ, λ, etc.

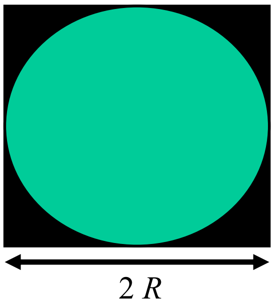


e.g., $P(x)$ can be complex function or tabulated

Throw darts at the rectangle and accept those under the curve

Pick x_1 in range $[a, b]$: $x_1 = a + \xi(b - a)$, calculate $P(x_1)$
Pick y_1 in range $[0, P_{max}]$: $y_1 = \xi\, P_{max}$
If $y_1 > P(x_1)$, reject x_1. Pick x_2, y_2 until $y_2 < P(x_2)$: accept x_2
Efficiency = Area under $P(x)$

Calculate π by the Rejection Method



Pick N random positions (x_i, y_i):
x_i in range $[-R, R]$: $x_i = (2\xi - 1)\, R$
y_i in range $[-R, R]$: $y_i = (2\xi - 1)\, R$
Reject (x_i, y_i) if $x_i^2 + y_i^2 > R^2$
Number accepted / $N = \pi R^2 / 4R^2$

$$N_A / N = \pi / 4$$

Increase accuracy (S/N): large N

FORTRAN 77:

```
do i = 1, N
  x = 2.*ran – 1.
  y = 2.*ran –1.
  if ( (x*x + y*y) .lt. 1. ) NA = NA + 1
end do
pi = 4.*NA / N
```

Lecture 12 revision quiz

- The algorithm for randomly sampling from the sin θ function has been sketched out. Work through this to get the correct normalization and sampling algorithm. Test the algorithm by writing a short computer code.
- What are some examples of functions that may be sampled efficiently/inefficiently using the rejection method?