NUCL 697 Homework 5: Monte Carlo Integration

Alex Hagen

April 12th, 2016

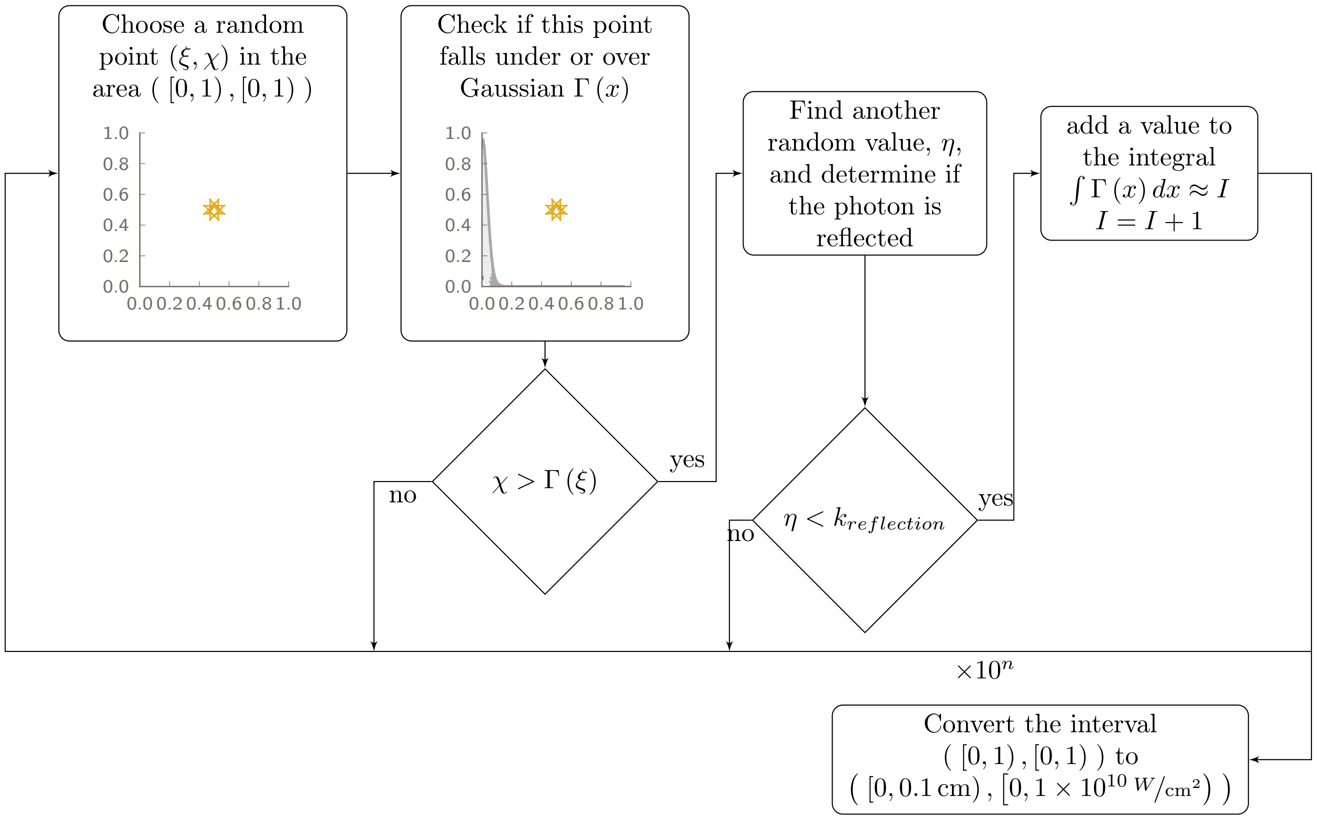
Use Monte Carlo method to calculate Nd:YAG ($1064\,\mathrm{nm}$) laser energy deposition on the surface of material with following parameters of laser beam: $1\times10^{10}\,{}^{W}\!/\!{}_{\mathrm{cm}^2}$ during $10\,\mathrm{ns}$ (square-shaped temporal distribution) Gaussian spatial distribution with radius=$1\times10^{-2}\,\mathrm{cm}$ (FWHM). Reflection coefficient of laser photons $= 0.7$. Use uniform mesh with $200\,\mathrm{cells}$ for $0.1\,\mathrm{cm}$ simulated domain. Plot results of fluence (${}^{J}\!/\!{}_{\mathrm{cm}^2}$) as a function of surface distance for 10^4, 10^6, and 10^8 photon numbers.

Motivation

Often, accurate integration - especially of high-order functions - is difficult or time consuming when done through explicit or even implicit means. In these cases, an alternative is Monte Carlo integration. Monte Carlo (MC) leverages the properties of random numbers and the law of averages to provide results after simulating the case many times.

In this case, the situtation of an Nd:YAG laser depositing onto a material with a certain reflection coefficient is simulated. MC is great for this situation because the gaussian that approximates the spatial distribution is very high order, and because the reflection coefficient complicates things in deterministic terms.

Method


Choose a random point (ξ,χ) in the area $([0,1),[0,1))$
Check if this point falls under or over Gaussian $\Gamma(x)$
$\chi > \Gamma(\xi)$
yes
no
Find another random value, η, and determine if the photon is reflected
$\eta < k_{reflection}$
yes
no
add a value to the integral $\int \Gamma(x)\,dx \approx I$
$I = I + 1$
$\times 10^n$
Convert the interval $([0,1),[0,1))$ to $([0, 0.1\,\mathrm{cm}), [0, 1\times10^{10}\,{}^{W}\!/\!{}_{\mathrm{cm}^2}))$

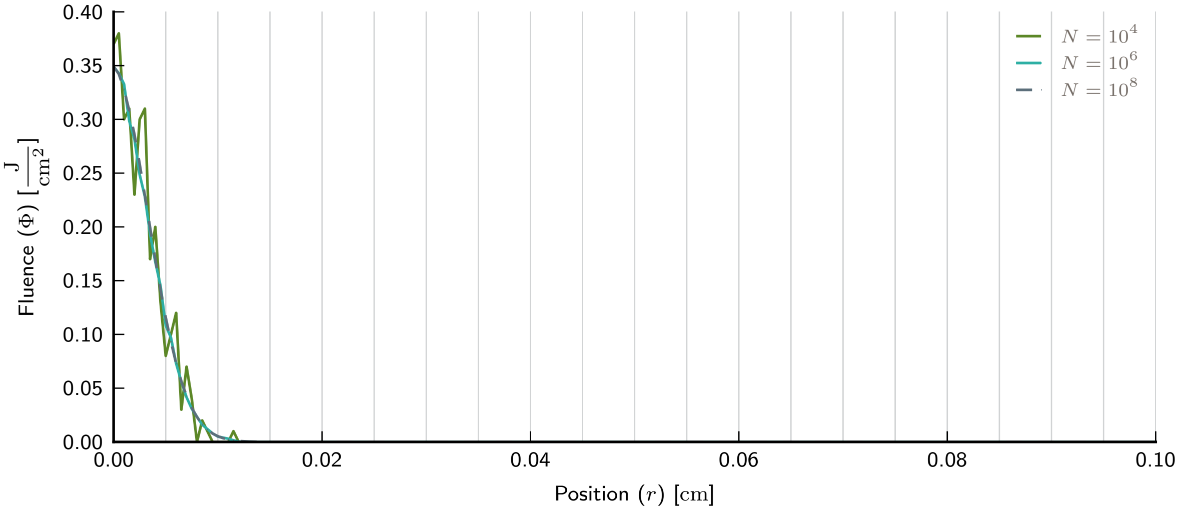


Figure 1: The Gaussian distribution of energy deposition with differing photon numbers

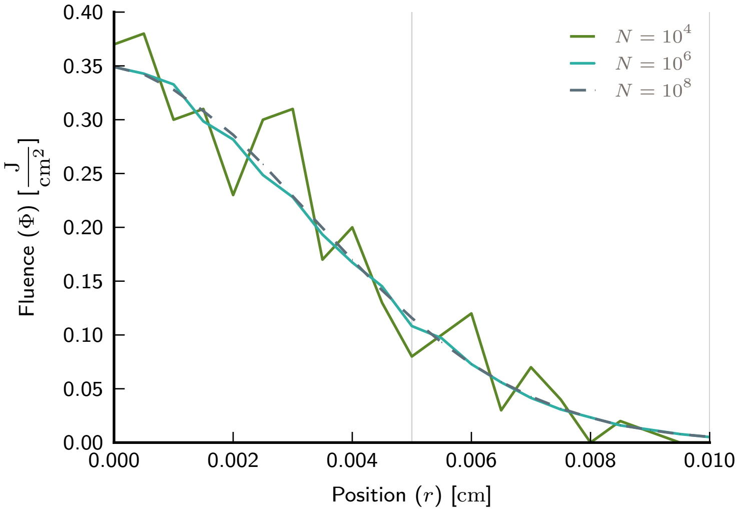


Figure 2: Closeup showing fluctuations in Gaussian distribution when using lower photon numbers

Results

As is shown in Figure 1, with the specified photon numbers, the gaussian spatial profile is recreated in every case. However, as is shown more clearly in Figure 2, the cases with low specified photon numbers (10^4 and 10^6) have some oscillation in their recreation of the gaussian spatial profile. This is a very graphic representation of the need for simulation repetition of many cases.

Code

mc.f90:

```fortran
program mc
implicit none

real(8)                                    ::    maxf, tot, sigma
real(8), allocatable, dimension(:)         ::    integral
real(8)                                    ::    ksi, chi, eta, x_r, f, f_r
integer(4)                                 ::    i, j, k, n
real(8)                                    ::    deltauni, s
integer(4)                                 ::    dim
real(8), allocatable, dimension(:)         ::    mesh
character(80)                              ::    text

! definition of some parameters
s = 1.0d-1                  ! 0.1 in cm
dim = 200                  ! number of cells

! create the uniform mesh
allocate(mesh(dim+1))
allocate(integral(dim))
deltauni = s / dble(dim)
mesh(1) = 0.0d0
do i=2, dim+1
    mesh(i) = mesh(i - 1) + deltauni
end do

! read in the number of desired particles
call getarg(1, text)
read(text,'(I10)') n
n= 10**n

! pass in the maximum value of the function
maxf = 1.0d0
! initialize the sum
tot = dble(n)
integral = 0.0d0
! initialize the standard deviation parameter
sigma = 1.0d-2 / 2.0d0 ! 1/2 of FWHM in cm

! cycle through number
do i=1, n
    ! find a mesh interval for the x cordinate that's randomly chosen
    ksi = rand()
    x_r = mesh(1) + (mesh(dim+1) - mesh(1)) * ksi

    ! determine the gaussian value in that mesh interval
    f = dexp(- x_r * x_r / sigma / sigma )

    ! create another random number which is the chosen random value
    chi = rand()
    f_r = maxf * chi

    ! now, if the value is less than the gaussian and another random number is
    ! less than the reflection coefficient, it deposits energy
    if (f_r <= f) then
        do k=1, dim+1
            if (x_r >= mesh(k) .and. x_r < mesh(k+1)) then
                eta = rand()
                if (eta < 0.70d0) then
                    integral(k) = integral(k) + 1.0d0
                end if
            end if
        end do
    end if
end do

! finally, we have an integral based on a 1x1 interval, so convert this to the
! realistic value (I_real = I_1 * 1x10^10 W/cm^2 * 10 ns / n_photon)
integral = integral * 1.0d10 * 10.0d-9 / dble(n)

! output this to stdout based on the mesh interval
do k=1, dim+1
    write(*,*) mesh(k),integral(k)
end do
end program
```