PH5023 – Monte Carlo Radiation Transfer
Non-assessed lab sessions on fortran programming
Kenny Wood
kw25@st-andrews.ac.uk

During the lab sessions each student will write their own codes to do the problems below. Please ask lots of questions in the lab as we want to make sure that by the end of the lab sessions everyone is comfortable programming in fortran. If you do not have any programming experience, please look over the web links to introductory fortran tutorials that I have placed on the PH5023 website.

Using the information given in the lectures on sampling from probability distributions, the rejection technique, and the basic structure of a Monte Carlo scattering code, write computer programs to compute the following:

1. Write a Monte Carlo code to calculate a value of π using the "darts" rejection technique described in the lecture notes.
2. Write a code to sample random optical depths from the probability distribution function $P(\tau) = \exp(-\tau)$ and plot a histogram of the number of random optical depths versus τ.
3. Write a Monte Carlo radiation transfer code to simulate emission from an isotropic point source at the origin of a uniform density sphere. Assume the Monte Carlo photon packets are scattered isotropically on their random walks through the sphere. For the case where the scattering albedo is unity, compute as a function of the sphere's radial optical depth, τ_r, the average number of scatterings per Monte Carlo packet, $\langle N_{scatt} \rangle$. Plot $\langle N_{scatt} \rangle$ versus τ_r and compare this with the analytic approximation $\langle N_{scatt} \rangle \sim \tau_r + \tau_r^2/2$, derived in the lecture notes. Investigate radial optical depths from 0.1 up to 200. To ensure that you have a working MCRT code, please send me a figure comparing your MCRT results with the analytic expression.

The subroutine for generating pseudo random numbers, ran2.f from *Numerical Recipes*, is provided on the website. As a guide, the code for calculating π should be around a dozen lines (excluding ran2.f), and the Monte Carlo scattering code (again excluding ran2.f) can be written in less than sixty lines. Be careful when declaring variables, especially any counters and think about whether single or double precision is required.