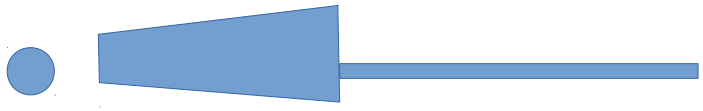
General description of current code:

1. geometry of disk-torus:

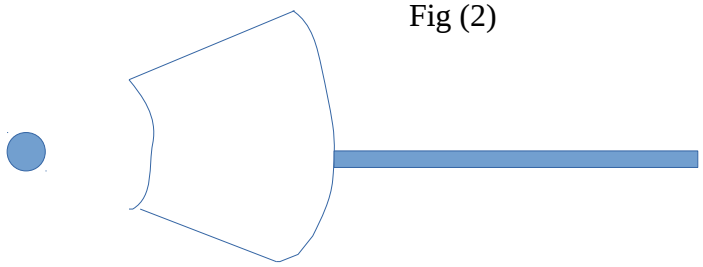
The geometry in previous Newtonian code is shown in Fig (1).

Fig (1)



Current geometry in relativistic code is shown in Fig (2).

Fig (2)



Such a new geometry of torus will allow us to easily check if photon is inside of the torus or not. This is done with “Dot Product” between photon position vector and unit vector of axis of symmetry of torus (namely, axis perpendicular to torus plane, not precession axis).

Therefore, in spherical coordinate (R, theta, phi), if photon is inside of the precessing torus, it requires:
(a) R_torus_in < R < R_torus_in, where R_torus_in/out are the inner and outer radius of the torus.
(b) |(n*k)/R| < sin(t), where n is the unit vector of torus axis, k is the photon position vector, t is the half opening angle of the torus.

2. Some additional input parameters:

Besides those original parameters in Newtonian code, there are some additional

parameters, for example, black hole mass, spin, mass accretion rate

3. What can current code do:

(a) generate the photon distribution from Novikov-Thorne disk
(b) ray-trace each photon from the disk,
(c) check whether photon enters into the torus or not.
(d) For those photon going to infinity, they will be correspondingly saved into 10 bins of theta angle (), and 4 bins of azimuthal angles (i.e., 0-29, 90-110, 180-200, 270-290 degs). Moreover, the combined gravitational and Doppler shift is taken into account for the photon energy at the infinity.

4. What is next to be done:

For those photons hitting the torus, applying the Compton scattering subroutine which is taken from Newtonian code.

5. Potential problem to be solved:

(a) there seems to be some error about finding geodesic for 0.1% of total photons (see the error report “error.txt”). This will be checked.

==========

To compile the code:

gcc sim5lib.c lfqpo.c -lm -lgsl -lgslcblas -DDEBUG -g -o mtc_uni

Current output files:

- input_spec.dat: normalized photon counts from NT disk in all direction.
  column 1: E
  column 2: F_E
- inf_spec.dat: normalized photon counts to infinity

column 1: E
column 2: F_E