README FILE OF YNOGKM

1. Introduction

This is the readme file for the code ynogkm[1], which is a public code aimed on the calculating of time-like geodesic orbits for electric charged ($\varepsilon \neq 0$) or electric charge-free ($\varepsilon = 0$) particles in a Kerr-Newmann spacetime (spin a, electric charge e). The reference for this code is:

ynogkm: A New Public Code For Calculating time-like Geodesics In The Kerr-Newmann Spacetime (Yang, Xiao-lin & Wang, Jian-cheng 2013 A&A, accepted). You are morally obligated to cite this paper in your scientific literature if you used the code in your research.

ynogkm is the direct extension of the published code **ynogk**. Its algorithm is also the same with **ynogk**. The 4 Boyer-Lindquist (B-L) coordinates (r, μ, ϕ, t) and proper time involved parameter σ are expressed as functions of parameter p, wihch is usually called as the Mino time. Where $\mu = \cos\theta$. The functions are: $r(p), \mu(p), \phi(p), t(p)$, and $\sigma(p)$. **ynogk** can not deal with the special cases with black hole spin $|a| = 1$ (If $e \neq 0$, this conditions becomes $a^2 + e^2 = 1$). This shortage has been overcome by **ynogkm**.
****

2. Sources File

The source file ynogkm.f90 contains three modules, they are:

1. Module **constants**——which defines many constants often used in the program.

2. Module **ellfunctions**——Which includes supporting subroutines and functions to com- pute Weierstrass and Jacobis elliptical functions and integrals, especially the subroutines for Carlsons integrals.

3. Module **blcoordinates**——Which contains supporting subroutines and functions to compute functions $r(p), \mu(p), \phi(p), t(p)$, and $\sigma(p)$.

4. Module **sigma2p_time2p**——Which contains routines to solve equations $\sigma(p) = \sigma_0$ and $t(p) = t_0$.

---

[1] Yun-Nan Observatory Geodesics in a Kerr-Newmann spacetime for Massive particles

****

In the sources code, a subroutine named **ynogkm** can calculate all of the functions with a given p. The header the subroutine is:

ynogkm(p|kvec, lambda, q, mve, ep, sin_ini, cos_ini, a_spin, e, r_ini, radi, mu, time, phi, sigma, cir_orbt, theta_star)

The parameters are:

a_spin, **e**—The spin a and electric charge e of the black hole.

r_ini——The initial radial coordinate of the particle.

sin_ini, **cos_ini**——Where **sin_ini** $= \sin\theta_{ini}$, **cos_ini** $= \cos\theta_{ini}$, and θ_{ini} is the initial θ coordinate of the particle.

cir_orbt——is a logical variable. If cir_orbt=.True., **ynogkm** calculates the B-L coordinates for spherical motion, and parameter theta_star ($=\theta_*$, see the discussion in section 6.2 of our paper) also should be specified, which is the θ coordinate of the turning point. If cir_orbt=.False., **ynogkm** calculates the B-L coordinates for non-spherical motion.

kvec——is an array, and kvec(1)=$k_{(r)}$, kvec(2)=$k_{(\theta)}$, kvec(3)=$k_{(\phi)}$, kvec(4)=$k_{(t)}$. The definitions of $k_{(\mu)}$ refer to Equations (91)-(95) of our paper. $k_{(\mu)}$ has the same signs with the initial four-momentum p_μ of the particle. Thus we use the signs of $k_{(\mu)}$ to determine the direction of the particle's motion, also for the signs in front of Π_r, Π_μ, Π_ξ.

lambda,q,mve,ep——are the four constants of motion, and **lambda**=$\lambda = L/E$, **q**=$q = Q/E^2$, **mve**=$m = \mu_m/E$, **ep**=$\varepsilon = \epsilon/E$. kvec, lambda, q, mve, and ep can be computed by a subroutine **lambdaqm**.

the header of **lambdaqm** is:

lambdaqm(vptl, sin_ini, cos_ini, a_spin, e, r_ini, signcharge, vobs, lambda, q, mve, ep, kvec)

where **vptl**(3) is an array contains the three physical velocities $\upsilon'_r, \upsilon'_\theta, \upsilon'_\phi$ of the particle with respect to an assumed emitter. And vptl(1)=υ'_r, vptl(2)=υ'_θ, vptl(3)=υ'_ϕ. Similarly, **vobs**(3) is an array contains the three physical velocities $\upsilon_r, \upsilon_\theta, \upsilon_\phi$ of the assumed emitter with respect to an LNRF reference. And vobs(1)=υ_r, vobs(2)=υ_θ, vobs(3)=υ_ϕ.

In the source file, the headers of functions $r(p)$ and $\mu(p)$ has the following forms:

radius(p|kvecr, lambda, q, mve, ep, a_spin, e, r_ini)
mucos(p|kp, kt, lambda, q, mve, sin_ini, cos_ini, a_spin)

where kvecr=$k_{(r)}$, kp=$k_{(\phi)}$, kt=$k_{(\theta)}$.

****

3. Six Examples

The package of the code contains 6 examples of our code applied to toy problems in the literature. Which demonstrate the utilities of our code to the reader. We show how to start one of them, the other ones are quite similar.

The example is to calculate the streamlines of stationary axisymmetric accretion flow composed by non-interacting particles falling onto a Kerr black hole (see the discussion in Section 6.5 of our paper). The source files are given in directory ./5_accretion_flow. To get start one can compile the file streamlines.f90 by following commands:

```
[@localhost 5_accretion_flow] $ g95 streamlines.f90 -o streamlines
[@localhost 5_accretion_flow] $ time ./streamlines <data.in
```

then one can use the IDL's command to draw the figure:

```
IDL> .r streamines.pro
IDL> streamlines
```

****

If you find any bugs or have any questions about ynogkm please sent an email to me. My email address is: yangxl@ynao.ac.cn.