CATAM Part II - 14.6 - Isolating Integrals for Geodesic Motion

Introduction

Throughout this project I code in python, making use of the SymPy package for symbolic mathematics. The GraviPy package provides data structures to store and manipulate the kinds of tensors that appear in general relativity. We'll only need it to compute the Christoffel symbols. SciPy will be used to perform numerical integration. We use the +— signature.

Question 1

We use the equivalent action

$$\mathcal{S} = \int \mathcal{L} d\tau$$

$$\mathcal{L} = g_{ij}\dot{x}^i\dot{x}^j = g_{tt}\dot{t}^2 + 2g_{t\phi}\dot{t}\dot{\phi} + g_{\phi\phi}\dot{\phi}^2 + g_{rr}\dot{r}^2 + g_{\theta\theta}\dot{\theta}^2$$

where the dot denotes differentiation with respect to τ. The conserved quantities come from the lack of dependence of $\mathcal{L}$ on t, ϕ.

No t dependence gives

$$\frac{\partial \mathcal{L}}{\partial t} = 2g_{tt}\dot{t} + 2g_{t\phi}\dot{\phi} = 2E \tag{1}$$

for E constant, and similarly no ϕ dependence gives

$$\frac{\partial \mathcal{L}}{\partial \phi} = 2g_{t\phi}\dot{t} + 2g_{\phi\phi}\dot{\phi} = -2L_z \tag{2}$$

for L_z constant. Rewriting these for later use we have:

$$E = g_{tt}\dot{t} + g_{t\phi}\dot{\phi} \tag{3}$$

$$L_z = -(g_{t\phi}\dot{t} + g_{\phi\phi}\dot{\phi}) \tag{4}$$

We get a further conserved quantity from no τ dependence

$$\mathcal{L} - \dot{x}^i \frac{\partial \mathcal{L}}{\partial \dot{x}^i} = -\mathcal{L} = -1 \tag{5}$$

where $\mathcal{L} = 1$ is by timelikeness [1]

(3) and (4) give a system of 2 equations for $\dot{t}, \dot{\phi}$ which can be solved to give

$$\dot{t} = \frac{Eg_{\phi\phi} + L_z g_{t\phi}}{g_{tt}g_{\phi\phi} - g_{t\phi}^2} \tag{6}$$

$$\dot{\phi} = -\frac{Eg_{t\phi} + L_z g_{tt}}{g_{tt}g_{\phi\phi} - g_{t\phi}^2} \tag{7}$$

which can be substituted into $\mathcal{L} = 1$ to give (after some simple but tedious algebra):

$$g_{rr}\dot{r}^2 + g_{\theta\theta}\dot{\theta}^2 = -V_{eff}(r, \theta, E, L_z) \tag{8}$$

$$V_{eff}(r, \theta, E, L_z) = -1 + \frac{E^2 g_{\phi\phi} + L_z^2 g_{tt} + 2ELg_{t\phi}}{g_{tt}g_{\phi\phi} - g_{t\phi}^2}$$

[1] reading ahead and noting the +— signature

Question 2

Computing the Christoffel symbols and returning latex is pretty straightforward using GraviPy, and was done using $compute\_kerr\_christoffel.py$. The non zero Christoffel symbols up to symmetry in the lower indices are displayed below.

We make some attempt to simplify the expressions through substitutions of Σ and Δ post computation, however our program certainly misses some. This won't matter for numerical calculation later. Here's what we get out:

$$
\begin{aligned}
\Gamma^{t}_{tr} &= -\frac{m\left(a^{2}+r^{2}\right)\left(a^{2}\cos^{2}\left(\theta\right)-r^{2}\right)}{\Delta\Sigma^{2}} \\
\Gamma^{t}_{t\theta} &= -\frac{a^{2}mr\sin\left(2\theta\right)}{\Sigma^{2}} \\
\Gamma^{t}_{r\phi} &= -\frac{am\left(2\Sigma r^{2}+a^{4}\sin^{2}\left(\theta\right)-a^{4}+a^{2}r^{2}\sin^{2}\left(\theta\right)+r^{4}\right)\sin^{2}\left(\theta\right)}{\Delta\Sigma^{2}} \\
\Gamma^{t}_{\theta\phi} &= \frac{amr\left(-\Delta\Sigma+a^{4}-2a^{2}mr+2a^{2}r^{2}-2mr^{3}+r^{4}\right)\sin\left(2\theta\right)}{\Delta\Sigma^{2}} \\
\Gamma^{r}_{tt} &= \frac{\Delta m\left(-a^{2}\cos^{2}\left(\theta\right)+r^{2}\right)}{\Sigma^{3}} \\
\Gamma^{r}_{t\phi} &= \frac{\Delta am\left(a^{2}\cos^{2}\left(\theta\right)-r^{2}\right)\sin^{2}\left(\theta\right)}{\Sigma^{3}} \\
\Gamma^{r}_{rr} &= \frac{r}{\Sigma}+\frac{m}{\Delta}-\frac{r}{\Delta} \\
\Gamma^{r}_{r\theta} &= -\frac{a^{2}\sin\left(2\theta\right)}{2\Sigma} \\
\Gamma^{r}_{\theta\theta} &= -\frac{\Delta r}{\Sigma} \\
\Gamma^{r}_{\phi\phi} &= \frac{\Delta\left(-8\Sigma^{2}r+a^{4}m\left(\cos\left(4\theta\right)-1\right)+8a^{2}mr^{2}\sin^{2}\left(\theta\right)\right)\sin^{2}\left(\theta\right)}{8\Sigma^{3}} \\
\Gamma^{\theta}_{tt} &= -\frac{a^{2}mr\sin\left(2\theta\right)}{\Sigma^{3}} \\
\Gamma^{\theta}_{t\phi} &= \frac{amr\left(a^{2}+r^{2}\right)\sin\left(2\theta\right)}{\Sigma^{3}} \\
\Gamma^{\theta}_{rr} &= \frac{a^{2}\sin\left(2\theta\right)}{2\Delta\Sigma} \\
\Gamma^{\theta}_{r\theta} &= \frac{r}{\Sigma} \\
\Gamma^{\theta}_{\theta\theta} &= -\frac{a^{2}\sin\left(2\theta\right)}{2\Sigma} \\
\Gamma^{\theta}_{\phi\phi} &= -\frac{\left(\Sigma\left(\Delta\Sigma+2a^{2}mr+2mr^{3}\right)+2a^{2}mr\left(a^{2}+r^{2}\right)\sin^{2}\left(\theta\right)\right)\sin\left(\theta\right)\cos\left(\theta\right)}{\Sigma^{3}} \\
\Gamma^{\phi}_{tr} &= \frac{am\left(-a^{2}\cos^{2}\left(\theta\right)+r^{2}\right)}{\Delta\Sigma^{2}} \\
\Gamma^{\phi}_{t\theta} &= -\frac{2amr}{\Sigma^{2}\tan\left(\theta\right)} \\
\Gamma^{\phi}_{r\phi} &= \frac{\Sigma^{2}r-2\Sigma mr^{2}-\frac{a^{4}m\left(\cos\left(4\theta\right)-1\right)}{8}-a^{2}mr^{2}\sin^{2}\left(\theta\right)}{\Delta\Sigma^{2}} \\
\Gamma^{\phi}_{\theta\phi} &= \frac{\Delta\Sigma^{2}-2\Delta\Sigma mr+2\Sigma a^{2}mr+2\Sigma mr^{3}+2a^{4}mr\sin^{2}\left(\theta\right)-4a^{2}m^{2}r^{2}+2a^{2}mr^{3}\sin^{2}\left(\theta\right)-4m^{2}r^{4}}{\Delta\Sigma^{2}\tan\left(\theta\right)}
\end{aligned}
$$

Programming Task

The file $integrate.py$ contains all our integration code. It takes as inputs initial values of the 8 dynamic variables, as well as $\Gamma, manda$, defines the ODEs and passes them to the odeint function from SciPy. We'll need to come back and fiddle with some parameters later to ensure our integration is precise enough for our purposes.

Question 3

We could set $a = 0$ in our Kerr Christoffel symbols, but find it much easier to calculate them from the metric. Using a section of $q3\_schwarzchild.py$:

$$\Gamma^t_{tr} = \frac{m}{r(-2m+r)}$$
$$\Gamma^r_{tt} = \frac{m(-2m+r)}{r^3}$$
$$\Gamma^r_{rr} = \frac{m}{r(2m-r)}$$
$$\Gamma^r_{\theta\theta} = 2m - r$$
$$\Gamma^r_{\phi\phi} = (2m-r)\sin^2(\theta)$$
$$\Gamma^\theta_{r\theta} = \frac{1}{r}$$
$$\Gamma^\theta_{\phi\phi} = -\frac{\sin(2\theta)}{2}$$
$$\Gamma^\phi_{r\phi} = \frac{1}{r}$$
$$\Gamma^\phi_{\theta\phi} = \frac{1}{\tan(\theta)}$$

Now substituting $\theta = \pi/2$, $g_{\phi\phi} = -r^2$, $g_{tt} = (1 - 2m/r)$ and $g_{t\phi} = 0$ into (8), we get:

$$V_{eff}(r, E, L_z) = -1 + \frac{E^2}{1 - 2m/r} - \frac{L^2}{r^2} = g_{rr}\dot{r}^2 + g_{\theta\theta}\dot{\theta}^2 \tag{9}$$

which has zeros as solutions of

$$r^3(E^2 - 1) + 2mr^2 - L_z^2 r + 2mL_z^2 = 0$$

which for $E = 0.97, L_z = 4, m = 1$ are $\approx 3.07, 7.61, 23.16$. By graphing, our potential is positive for $7.61 < r < 23.16$ or $2 < r < 3.07$ (ignoring $r < 2$ within schwarzchild radius, not physical for a timelike geodesic). Now a bound orbit in the equatorial plane must have $\dot{\theta} = 0$, and so (9) becomes

$$\dot{r}^2 = -(1 - 2/r)V_{eff}$$

Only some of these solutions will be bound... TODO

3a

Using (3), (4) and (9) we can calculate initial values for $\dot{t}, \dot{phi}$ and $\dot{\theta}$ respectively. Calling the function *compute_one_geodesic*() in *q3_schwarzchild.py*, we produce the following plots using parameters $r = 15, \theta = \pi/2, \dot{r} = 0, E = 0.97, L = 4$. Some fiddling around with the integration parameters was required to get a smooth enough curve, over a few orbits. Looking at the ϕ plot, we have just over 3 orbits. Calculating E, L_z, and $g_{rr}\dot{r}^2 + g_{\theta\theta}\dot{\theta}^2 + V_{eff}(r, \theta, E, L_z)$ at each point we get our 3 conservation laws to high precision, assuring us the numerical integration is accurate enough for our purposes .

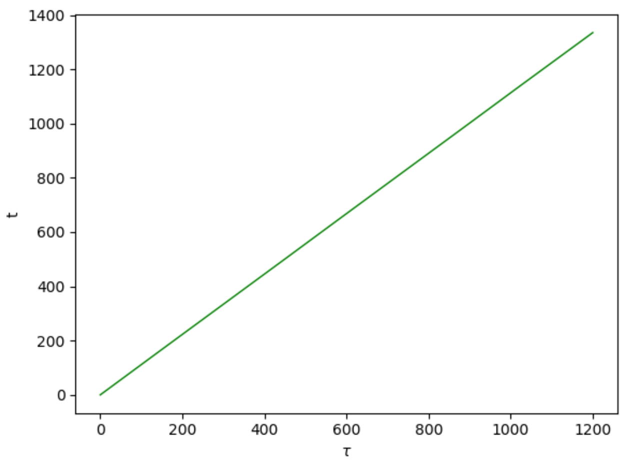

Figure 1: t

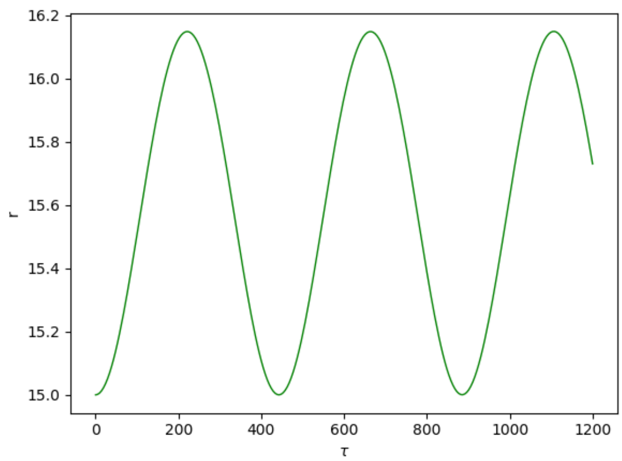

Figure 2: r

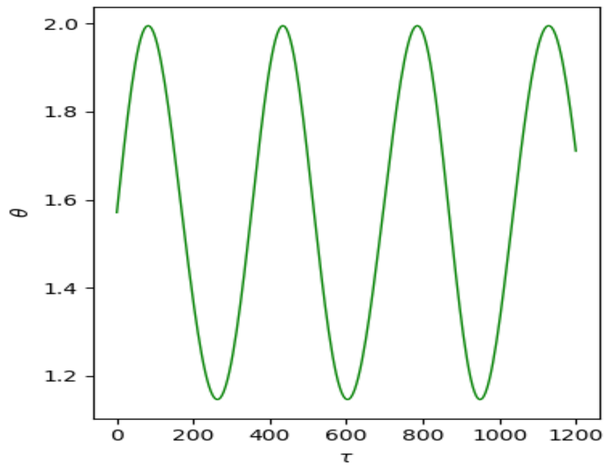

Figure 3: θ

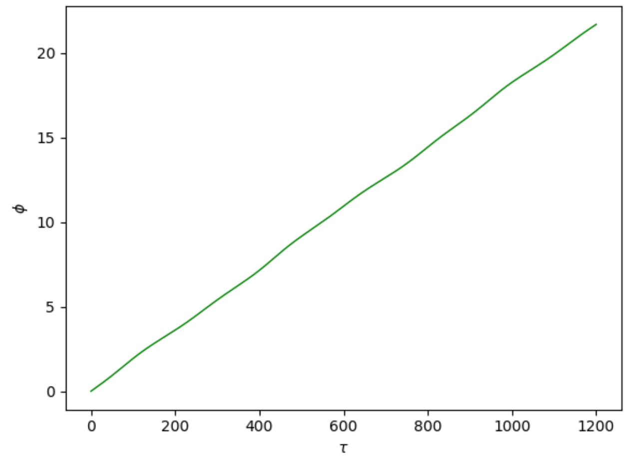

Figure 4: ϕ

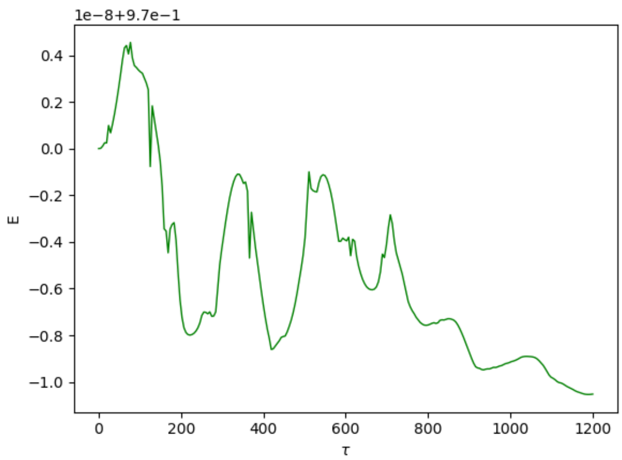

Figure 5: E

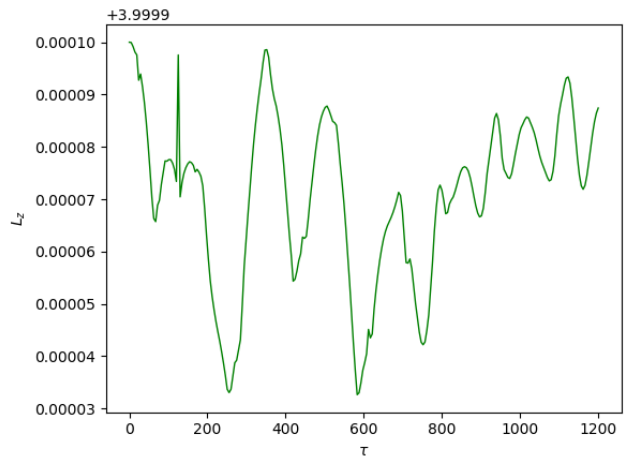

Figure 6: L

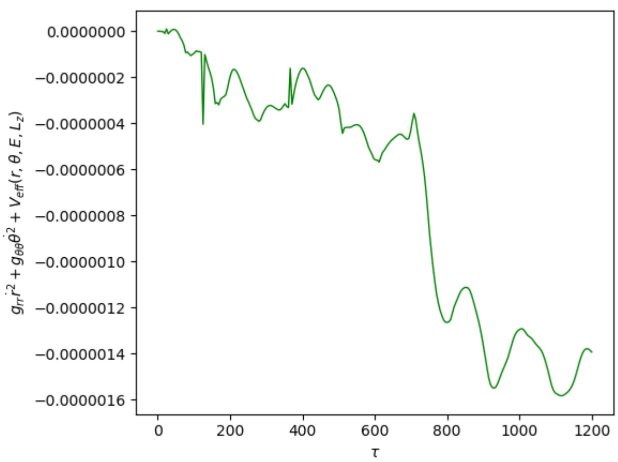

Figure 7: $g_{rr}\dot{r}^2 + g_{\theta\theta}\dot{\theta}^2 + V_{eff}(r, \theta, E, L_z)$

3b

We find as expected our code runs into issues if r is not in the range defined previously. Running the code listed under 3b in *q3.py* with several r0 values we obtain for fixed $\theta = \pi/2, \dot{r} = 0, E = 0.97, L = 4$ tje following plots. We see the values form a closed curve under these conditions, with the shape of curve depending on the initial parameters

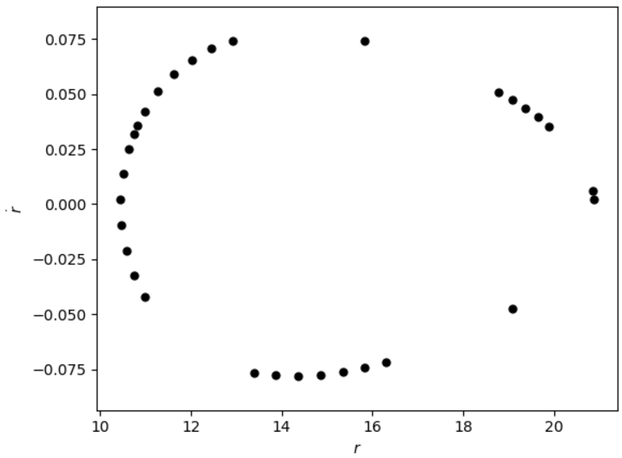

Figure 8: $r_0 = 10$

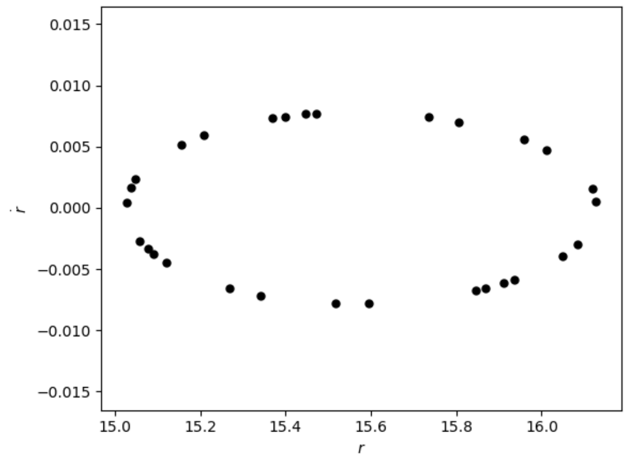

Figure 9: $r_0 = 15$

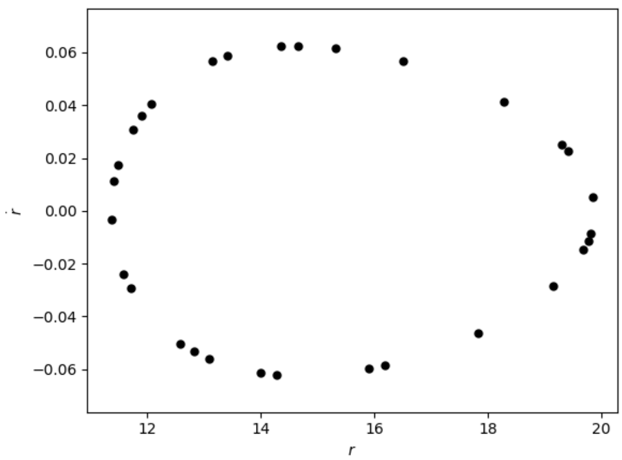

Figure 10: $r_0 = 20$

Question 4

Plotting the potential using *q4.py* and using a numerical root finder, we see positive potential for $4.513 < r < 14.564$. Now modifying *q3.py* to accomodate the Kerr metric, we produce the following Poincare maps, which all look very much like the Schwartzchild ones

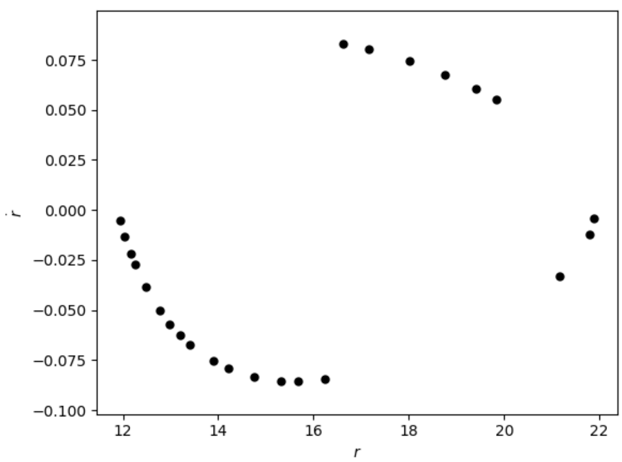

Figure 11: $r_0 = 10$

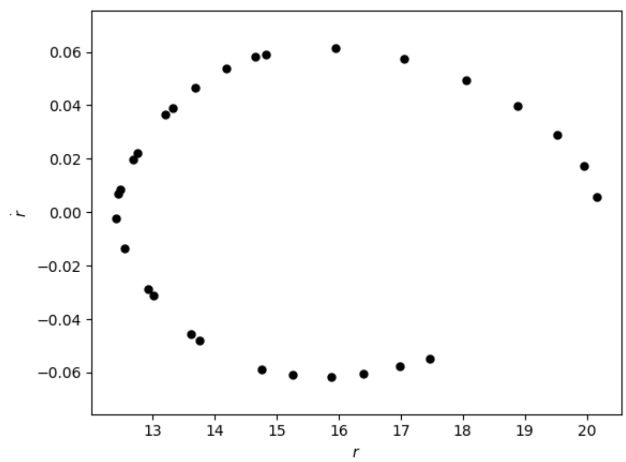

Figure 12: $r_0 = 15$

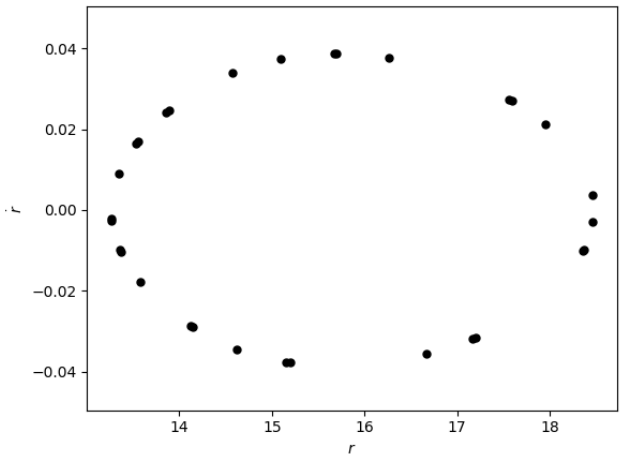

Figure 13: $r_0 = 20$

Question 5

The script *q5.py* attempts to symbolically simplify a hand computed expression for $\dot{Q}$ by symbolically calculating and substituting in $\ddot{\theta}$ in terms of $r, m, a, theta, \dot{t}, \dot{r}$ and $\dot{\phi}$ from the geodesic equation and Christoffel symbols. I couldn't manage to get this to work, perhaps due to choice of a fairly weak symbolic tool, and ended up with a horrendously complicated expression I could not simplify.

In the $a = 0$ limit Q becomes

$$Q = L_z^2 \csc^2\theta + r^2\dot{\theta}^2$$

choosing $\theta(\tau) = \pi/2$, which we are always free to do, we get

$$Q = L_z^2$$

the square of the total orbital angular momentum. In the general case general, this constant is known as the Carter constant [2] , which along with energy angular momentum, and rest mass, give enough conserved quantities to determine orbits uniquely give initial conditions. It doesn't seem immediately obvious what physical quantity it represents - a "hidden" symmetry if you will.

$$-\frac{2\left(-\Delta^2\Sigma^2\dot{\phi}^2 + \Delta\Sigma^2\dot{\theta}^2a^2 + \Delta\Sigma\delta a^2 + \Delta\Sigma\dot{\phi}^2a^4 - 2\Delta\Sigma\dot{\phi}^2a^2mr + 2\Delta\Sigma\dot{\phi}^2a^2r^2 - 2\Delta\Sigma\dot{\phi}^2mr^3 + \Delta\Sigma\dot{\phi}^2r^4 - \Delta\Sigma\dot{t}^2a^2 + 2\Delta\dot{\phi}^2a^4m\right.}{\Delta\Sigma}$$

[2]see for instance https://en.wikipedia.org/wiki/Carter˙constant