CBPF-NF-014/98

Theorems of Birkhoff Type in Finsler Spaces

Solange F. Rutz

Centro Brasileiro de Pesquisas Físicas - CBPF
Departamento de Relatividades e Partículas – DRP
Rua Dr. Xavier Sigaud, 150
22290-180 – Rio de Janeiro, RJ – Brazil
E-mail: rutz@cbpfsu1.cat.cbpf.br

Abstract

In this paper we give a summary of basic definitions of Finsler geometry and a brief account of symmetry imposition for Finsler metrics, as well as a possible extension of General Relativity, in the vacuum case, for these spaces. We then make use of these results in order to generalise, for up to first order departures from the Riemannian setting, the well known Birkhoff's theorem from General Relativity. This result has been fully accomplished by means of computer algebra – actually, the very problem of explicitly determining non-Riemannian solutions for some generalised theory of gravity has only been made possible by computer algebra. The process is described in detail within the paper.

Key-words: Computer algebra; Finsler spaces; Theories of gravity.

1 Introduction.

It is well known how crucial computer algebra has been in the development of General Relativity and its applications, and, in the other way round, how important General Relativity and tensorial calculus have been to motivate and direct developments of computer algebra packages. Riemannian geometry, curved spaces and tensorial calculus were already complicated enough for even more generalised frameworks for theories of gravity to be attempted without strong motivation, either theoretical or experimental ([1]). Regarding observations, General Relativity and its Riemannian model for space-time surprised the scientific community of the time when it accurately predicted the advance of the perihelium of Mercury, an age-old problem of Newtonian theory, which could never be precisely dealt with in an Euclidean context. Also, its prediction for the bending of light in the gravitational field, later confirmed observationally, definitively settled the acceptance of the new theory, and seemed to have established a new paradigm for the geometry of the physical space.

But the advent of other geometries than Euclidean, and so the possibility of other models for physical space, has been realised ever since Gauss himself. Riemann, in his famous lecture [3], did not restrict himself to what became known as Riemannian geometry when he pushed forward the very concept of space. So the freedom is there to explore larger possibilities to model our reality and maybe find out new and yet unobserved natural behaviour such as it has been with the bending of light. Paul Finsler [2] followed the steps led by Riemann and Gauss, in the tradition of the Göttingen school, and now computer algebra came to make it possible to perform the prohibitively complex calculations that taking his geometry as model for physical space-time and gravity makes necessary.

Apart from attempts at a theoretical unification of gravitation and electromagnetic phenomena in a single geometrical framework, Finsler spaces were also considered in a purely gravitational context, either as formal propositions of new theoretical structures and field equations [4, 5, 6, 7, 8, 9, 10], or more directly concerned with exploring possible observational consequences, either in the cosmological [11] or local (solar system) setting [12, 13, 14, 15, 16, 17], prior to a full theoretical proposition. In any case, the first derivation of an explicit solution (even if just in a perturbative sense) to a Finsler gravitational field equation [18] was made possible only by means of computer algebra programming.

In this paper we present a review of an approach to Finsler gravity and also new results regarding this theory. We start by presenting a summary of basic definitions of Finsler ge-

[1] At one time [1] the attempt to formulate an unifying theory for gravity and electromagnetism led to unsuccessful attempts at Finsler [2] and other generalised geometrical modelling of natural phenomena.

ometry, together with a treatment of symmetries in these spaces, a frequently sought after technique used to model physical systems and at the same time simplifying the equations. Next we go through an argument which, by means of an analogy between Newtonian theory and General Relativity, extends Einstein's vacuum field equations to Finsler spaces. Then we show how, by means of computer algebra, the well-known Birkhoff's theorem from General Relativity [19], which establishes the unicity and time-independence of the spherically symmetric family of solutions known as Schwarzschild metrics, naturally extends itself, up to first order perturbations, to the Finsler setting.

2 Finsler Spaces.

Finsler [2] spaces are n-dimensional manifolds where the infinitesimal distance between two neighbouring points $P(x^i)$, $Q(x^i + dx^i)$ is given by

$$ds = F(x^i,\ dx^i),\ i = 1, \ldots, n, \tag{1}$$

where F is required to satisfy some properties [20], the most important being that

$$F(x^i,\ \lambda dx^i) = \lambda\ F(x^i,\ dx^i),\ \lambda > 0, \tag{2}$$

i.e., F must be positively homogeneous of degree 1 in the dx^i. A metric tensor is then defined from (1) as

$$g_{ij}(x^k,\ \dot{x}^k) = \frac{1}{2}\frac{\partial^2 F^2(x^k,\ \dot{x}^k)}{\partial \dot{x}^i\ \partial \dot{x}^j}, \tag{3}$$

where $\dot{x}^i = dx^i/dt$ are the tangent vectors to a given curve $x^i = x^i(t)$ in the manifold, or elements of the tangent space $T_n(P)$ at $P(x^i)$. Due to (2), $g_{ij}(x^k,\ \dot{x}^k)$ is homogeneous of degree zero in the second set of variables and so we can write

$$ds^2 = g_{ij}(x^k,\ dx^k)\ dx^i dx^j\ , \tag{4}$$

from where we can see that Riemannian spaces are special cases of Finsler spaces, where (3) has no directional dependence, or F^2 is quadratic in the dx^i.

The geodesics in a Finsler space can be given in a similar form to the Riemannian case as

$$\frac{d^2 x^i}{ds^2} + \gamma^i_{jk}\ \frac{dx^j}{ds}\frac{dx^k}{ds} = 0 \tag{5}$$

where γ^i_{jk} are the Christoffel symbols of second kind

$$\gamma^i_{jk} = g^{ih}\gamma_{jhk} = g^{ih}\left[\frac{1}{2}\ \left(\frac{\partial g_{hj}}{\partial x^k} + \frac{\partial g_{hk}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^h}\right)\right], \tag{6}$$

the difference with the Riemannian case being that γ^i_{jk} now depend on the directional coordinates $\dot{x}^i$ as well as x^i. Here, g^{ih} is such that $g_{ij}(x^k,\dot{x}^k)\ g^{ih}(x^k,y_k) = \delta^h_j$, and $y_i = g_{ij}(x^k,\dot{x}^k)\ \dot{x}^j$.

As in Riemannian geometry, there are many ways to define a covariant derivative in Finsler spaces, depending on the explicit expression chosen as the connection, and indeed, at least four different ones can be found in the literature [20], [21], [22], [23]. For instance, H. Rund defines the δ-derivative of a vector $X^i(x^j)$ defined along a curve $x^i(t)$ as

$$\frac{\delta X^i}{\delta t} = \frac{dX^i}{dt} + \Gamma^i_{jk}\ \dot{x}^j X^k \tag{7}$$

with

$$\Gamma^i_{jk} = \gamma^i_{jk} - C^i_{jl}\ \gamma^l_{pk}\dot{x}^p, \quad C^i_{jl} = g^{ih}\ C_{jhl} = g^{ih}\ \left(\frac{1}{2}\,\frac{\partial g_{jl}}{\partial \dot{x}^h}\right)\ .$$

Clearly, (7) reduces to the usual Riemannian covariant derivative if all the C_{jhl} are identically zero. The partial δ-derivative of $X^i(x^j)$ is then given by

$$X^i_{;j} = \frac{\partial X^i}{\partial x^j} + \Gamma^{*i}_{hj}\ X^h, \tag{8}$$

where

$$\Gamma^{*i}_{hj} = g^{ik}\ \Gamma^*_{hkj} = g^{ik}\ \left[\gamma_{hkj} - \left(C_{jki}\ \Gamma^i_{hl} + C_{khi}\ \Gamma^i_{jl} - C_{hji}\ \Gamma^i_{kl}\right)\dot{x}^l\right].$$

The relationship between the processes (7) and (8) is best expressed by the relation

$$\frac{\delta X^i}{\delta t} = X^i_{;j}\ \dot{x}^j = X^i_{;j}\ \frac{dx^j}{dt}.$$

As in Riemannian geometry, the second covariant derivatives of the deviation vector ξ^i between two close geodesics issuing from the same point P represent an invariant measure of the curvature of F_n in the neighbourhood of P, satisfying the so called geodesic deviation equation

$$\frac{\delta^2\xi^i}{\delta s^2} + K^i_{jhk}\dot{x}^j\dot{x}^h\xi^k = 0 \tag{9}$$

where $\dot{x}^i = dx^i/ds$ here denotes the tangent vectors along the geodesic, and

$$K^i_{jhk} = \left(\frac{\partial\Gamma^{*i}_{jh}}{\partial x^k} - \frac{\partial\Gamma^{*i}_{jh}}{\partial\dot{x}^l}\frac{\partial G^l}{\partial\dot{x}^k}\right) - \left(\frac{\partial\Gamma^{*i}_{jk}}{\partial x^h} - \frac{\partial\Gamma^{*i}_{jk}}{\partial\dot{x}^l}\frac{\partial G^l}{\partial\dot{x}^h}\right) + \Gamma^{*i}_{mk}\Gamma^{*m}_{jh} - \Gamma^{*i}_{mh}\Gamma^{*m}_{jk} \tag{10}$$

is the curvature tensor of F_n, G^l being given by $G^l = (1/2)\ (\gamma^l_{jk}\dot{x}^j\dot{x}^k)$. It should be noted that other choices of connections would lead to different curvature tensors, but equation (9) would remain valid and could be given in terms of them. We should also note that K^i_{jhk} actually reduces to the Riemann tensor when the space is Riemannian.

Let us now consider small point transformations

$$\bar{x}^i = x^i + \varepsilon\, \xi^i(x^j) \tag{11}$$

together with the corresponding transformations in $T_n(P)$

$$d\bar{x}^i = dx^i + \varepsilon\, \frac{\partial \xi^i}{\partial x^j}\, dx^j \tag{12}$$

which we will denote by $\delta x^i = \bar{x}^i - x^i = \varepsilon\, \xi^i(x^j)$, $\delta \dot{x}^i = \dot{\bar{x}}^i - \dot{x}^i = \varepsilon\, \partial\xi^i/\partial x^j\, \dot{x}^j$. Here ε is a small parameter which will be considered only up to first order.

In general the distance $F(x^i,\ dx^i)$ between $P(x^i)$, $Q(x^i + dx^i)$ is not preserved under (11), (12), that is, the distance $F(\bar{x}^i,\ d\bar{x}^i)$ between the correspondent points $\bar{P}(\bar{x}^i)$, $\bar{Q}(\bar{x}^i + d\bar{x}^i)$ will generally be different from $F(x^i,\ dx^i)$. But, depending on the particular form assumed by F, it may admit special types of transformations for which distances are preserved, which are then called *motions* of the space onto itself.

Defining

$$\bar{F}(\bar{x}^i,\ d\bar{x}^i) = F(x^i,\ dx^i) \tag{13}$$

as a new 'deformed' metric function from the original one under (11), (12), and also considering that

$$\begin{aligned} \bar{F}(\bar{x}^i,\ d\bar{x}^i) &= \bar{F}(x^i + \varepsilon\, \xi^i,\ dx^i + \varepsilon d\xi^i) \\ &= \bar{F}(x^i,\ dx^i) + \varepsilon\, \xi^i \frac{\partial \bar{F}}{\partial x^i} + \varepsilon\, d\xi^i \frac{\partial \bar{F}}{\partial (dx^i)} \end{aligned} \tag{14}$$

up to first order in ε, we can write the *Lie difference*

$$F(x^i,\ dx^i) - \bar{F}(x^i,\ dx^i) = \varepsilon\, \left(\xi^i\, \frac{\partial F}{\partial x^i} + d\xi^i\, \frac{\partial F}{\partial (dx^i)} \right), \tag{15}$$

since, from (13), (15), $\varepsilon\, F(x^i,\ dx^i) = \varepsilon\, \bar{F}(x^i,\ dx^i)$, up to first order in ε. Defining the *Lie derivative* of the metric function F under (11), (12) as $\pounds_\xi F = \lim_{\varepsilon \to 0} \left\{ \left[\bar{F}(x^i,\ dx^i) \right.\right.$ $\left.\left. - F(x^i,\ dx^i) \right] / \varepsilon \right\}$, we then have that the condition for (11), (12) to be a motion can be simply expressed as $\pounds_\xi F = 0$ or

$$\xi^i\, \frac{\partial F}{\partial x^i} + \frac{\partial \xi^i}{\partial x^j}\, dx^j\, \frac{\partial F}{\partial (dx^i)} = 0. \tag{16}$$

We may as well obtain a new 'deformed' metric tensor under (11), (12)

$$\bar{g}_{jk}(\bar{x}^i,\ \dot{\bar{x}}^i) = g_{rs}(x^i,\ \dot{x}^i)\, \frac{\partial x^r}{\partial \bar{x}^j} \frac{\partial x^s}{\partial \bar{x}^k}. \tag{17}$$

The expression for the Lie derivative of the metric tensor has already been given by various authors. For instance, Yano [24] obtains

$$\pounds_\xi g_{ij} = \xi^k \frac{\partial g_{ij}}{\partial x^k} + \frac{\partial \xi^k}{\partial x^l}\, \dot{x}^l \frac{\partial g_{ij}}{\partial \dot{x}^k} + g_{kj} \frac{\partial \xi^k}{\partial x^i} + g_{ik} \frac{\partial \xi^k}{\partial x^j}. \tag{18}$$

Yano also presents a theorem [24, theorem 3.2], equivalent to its Riemannian counterpart, in which it is assured that a necessary and sufficient condition for (11), (12) to be a motion in a Finsler space is that

$$\pounds_\xi g_{ij} = 0, \tag{19}$$

which is called the *Killing equation* in Finsler spaces.

Solving (19) is the usual method for imposition of symmetries (or invariance under a chosen group of transformations) in Riemannian spaces. But, in the Finsler context, where all geometrical objects depend on both x^i and dx^i and therefore ds not only transforms as a scalar but actually is one, it is a much simpler, and yet equivalent, task to solve (16) rather than (19). Conversely, if we have a Finsler metric which satisfies (16), it will also satisfy (19), so this method applies when considering Finsler formalisms, such as Cartan's [22], which takes the metric tensor g_{ij}, rather than the metric function F as starting point.

We define, as in Riemannian geometry, spherical symmetry to be invariance under the action of the group SO(3). In spherical coordinates, these transformations are given by

$$\begin{bmatrix} \delta r \\ \delta\theta \\ \delta\phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\varepsilon \end{bmatrix}, \quad \begin{bmatrix} \delta r \\ \delta\theta \\ \delta\phi \end{bmatrix} = \begin{bmatrix} 0 \\ \eta \sin\phi \\ \eta \dfrac{\cos\phi}{\tan\theta} \end{bmatrix}, \quad \begin{bmatrix} \delta r \\ \delta\theta \\ \delta\phi \end{bmatrix} = \begin{bmatrix} 0 \\ -\zeta\cos\phi \\ \zeta \dfrac{\sin\phi}{\tan\theta} \end{bmatrix}. \tag{20}$$

The corresponding transformations for tangent vectors are, for the parameter ε, $\delta\dot{r} = \delta\dot{\theta} = \delta\dot{\phi} = 0$, and, for the other parameters η and ζ, we find

$$\begin{bmatrix} \delta\dot{r} \\ \delta\dot{\theta} \\ \delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ \eta\cos\phi\ \dot{\phi} \\ -\eta \dfrac{\cos\phi}{\sin^2\theta}\, \dot{\theta} - \eta \dfrac{\sin\phi}{\tan\theta}\, \dot{\phi} \end{bmatrix}, \quad \begin{bmatrix} \delta\dot{r} \\ \delta\dot{\theta} \\ \delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ \zeta\sin\phi\ \dot{\phi} \\ -\zeta \dfrac{\sin\phi}{\sin^2\theta}\, \dot{\theta} + \zeta \dfrac{\cos\phi}{\tan\theta}\, \dot{\phi} \end{bmatrix}. \tag{21}$$

To determine the spherically symmetric metric in 4 dimensions, we consider a fourth coordinate, say t, to remain invariant, $\delta t = 0$ for all three parameters ε, η, ζ.

The coordinate system for $\dot{x}^i$ which best simplifies the resulting invariance equations (16) for these transformations may be obtained by first changing to coordinates $X = r\,\dot{\theta}$,

$Y = r\ \sin\theta\ \dot{\phi}$, $Z = \dot{r}$, $T = \dot{t}$, then to 'spherical' coordinates

$$\begin{aligned} X &= r\ \dot{\theta} &&= R\ \sin\alpha \sin\beta \sin\gamma\ , \\ Y &= r\ \sin\theta\ \dot{\phi} &&= R \sin\alpha \sin\beta \cos\gamma\ , \\ Z &= \dot{r} &&= R \sin\alpha \cos\beta\ , \\ T &= \dot{t} &&= R \cos\alpha\ , \end{aligned} \tag{22}$$

where $0 < R < \infty$, $0 < \alpha < \pi$, $0 < \beta < \pi$ and $0 < \gamma < 2\pi$, avoiding coordinate singularities. Inversally, we have $R = \sqrt{X^2 + Y^2 + Z^2 + T^2}$, and

$$\begin{aligned} \sin\alpha &= \frac{\sqrt{X^2+Y^2+Z^2}}{\sqrt{X^2+Y^2+Z^2+T^2}}\ , & \cos\alpha &= \frac{T}{\sqrt{X^2+Y^2+Z^2+T^2}}\ , \\ \sin\beta &= \frac{\sqrt{X^2+Y^2}}{\sqrt{X^2+Y^2+Z^2}}\ , & \cos\beta &= \frac{Z}{\sqrt{X^2+Y^2+Z^2}}\ , \\ \sin\gamma &= \frac{X}{\sqrt{X^2+Y^2}}\ , & \cos\gamma &= \frac{Y}{\sqrt{X^2+Y^2}}\ . \end{aligned} \tag{23}$$

For those coordinates, we get, for the parameter ε, $\delta R = \delta\alpha = \delta\beta = \delta\gamma = 0$, and for the parameters η and ζ we also find $\delta R = \delta\alpha = \delta\beta = 0$ and, respectively,

$$\delta\gamma = \eta\ \frac{\cos\phi}{\sin\theta}, \quad \delta\gamma = \zeta\ \frac{\sin\phi}{\sin\theta}. \tag{24}$$

The invariance equations for a function $G = G(r,\ \theta,\ \phi,\ t,\ R,\ \alpha,\ \beta,\ \gamma)$ are, for each parameter considered:

$$\frac{\partial G}{\partial\phi} = 0, \tag{25}$$

$$\sin\phi\ \frac{\partial G}{\partial\theta} + \frac{\cos\phi}{\tan\theta}\ \frac{\partial G}{\partial\phi} - \frac{\cos\phi}{\sin\theta}\ \frac{\partial G}{\partial\gamma} = 0, \tag{26}$$

$$-\cos\phi\ \frac{\partial G}{\partial\theta} + \frac{\sin\phi}{\tan\theta}\ \frac{\partial G}{\partial\phi} - \frac{\sin\phi}{\sin\theta}\ \frac{\partial G}{\partial\gamma} = 0. \tag{27}$$

The resulting invariant function is given by $G = G(r,\ t,\ R,\ \alpha,\ \beta)$, which, once required to satisfy the homogeneity condition, becomes

$$F^2(x^i,\ \dot{x}^i) = R^2\ G(r,\ t,\ \alpha,\ \beta). \tag{28}$$

In paper [25] we also discuss coordinate freedom and re-obtain (28) in the more convenient form for comparative purposes

$$ds^2 = g_{11}\ dr^2 + g_{22}\ r^2 d\Omega^2 + g_{44}\ dt^2 + 2\left(g_{12}\ dr r d\Omega + g_{14}\ dr dt + g_{24}\ r d\Omega dt\right), \tag{29}$$

where

$$d\Omega = \sqrt{d\theta^2 + \sin^2\theta\ d\phi^2},$$

and the g_{ij} are symmetric in i and j, but otherwise arbitrary functions of r, t, α, β.

3 Geometrical Extension of General Relativity.

Four-dimensional symmetric Finsler metrics are particularly well suited for geometrical generalisations of theories of gravity such as General Relativity, based on (pseudo-) Riemannian spaces with three 'space-like' coordinates and a fourth 'time-like' one. We will briefly expose here, as much as necessary, one such a generalisation, as proposed in [18].

The vacuum field equation, for instance, is proposed from an analogy, first drawn by Pirani [26], between Newton's and Einstein's theories of gravity. If we subtract, in Newton's theory, the equations of motion of two neighbouring test particles subject only to a gravitational potential $\phi(x^i)$

$$\begin{aligned} \frac{d^2x^i}{dt^2} &= -\eta^{ij}\ \frac{\partial\phi}{\partial x^j}, \\ \frac{d^2x^i}{dt^2}+\frac{d^2\xi^i}{dt^2} &= -\eta^{ij}\ \frac{\partial\phi}{\partial x^j}-\xi^k\ \eta^{ij}\ \frac{\partial^2\phi}{\partial x^j\partial x^k}, \end{aligned} \tag{30}$$

(where η^{ij} is the three dimensional Euclidean metric in Cartesian coordinates, the derivatives of ϕ are evaluated at x^i and terms of second or higher order in ξ^i are neglected), we obtain a differential equation for the deviation vector ξ^i

$$\frac{d^2\xi^i}{dt^2}+H^i_k\ \xi^k=0, \quad H^i_k=\eta^{ij}\ \frac{\partial^2\phi}{\partial x^j}\partial x^k. \tag{31}$$

We note that Laplace's equation, $\nabla^2\phi=\eta^{ij}\ \partial^2\phi/\partial x^i\partial x^j=0$, valid for the vacuum in Newton's theory, implies that the tensor H^i_k is traceless, $H=H^i_i=0$.

Now, in General Relativity, a similar situation is represented by the (Riemannian) geodesic deviation equation

$$\frac{D^2\xi^i}{Ds^2}+H^i_k\ \xi^k=0, \quad H^i_k=R^i_{jlk}\ \dot{x}^j\dot{x}^l, \tag{32}$$

where now $\dot{x}^i$ is the tangent vector to the geodesic $x^i(s)$, R^i_{jlk} is the curvature tensor and $D^2\xi^i/Ds^2$ is the second covariant derivative with respect to the element of arc ds of the deviation vector ξ^i between neighbouring geodesics, which represents in this theory the trajectories of test particles (as well as light rays) subject only to gravity. Indices now run from 1 to 4. Einstein's vacuum field equations, $R_{jl}=R^i_{jli}=0$, also implies, as Laplace's equation in the Newtonian case, that $H=H^i_i=0$.

If we take the Finsler geodesic deviation equation (9)

$$\frac{\delta^2\xi^i}{\delta s^2}+H^i_k\ \xi^k=0, \quad H^i_k=K^i_{jlk}\ \dot{x}^j\dot{x}^l, \tag{33}$$

which is very similar to (33), just taking the $\delta-$derivative as covariant derivative and K^i_{jlk} as curvature tensor, we may analogously propose that there should be an equation, to hold in the vacuum, requiring that the Finsler deviation tensor $H^i{}_k$ to be traceless

$$H = 0. \tag{34}$$

We would like to note that the deviation tensor may be expressed also in terms of one of Cartan's [22] curvature tensor, $H^i_k = R^i_{ijlk}\, \dot{x}^j \dot{x}^l$ This tensor has been named so by Berwald, who also presents it [27] explicitly in terms of the metric tensor and its derivatives as

$$H^i{}_k = 2\frac{\partial G^i}{\partial x^k} - \frac{\partial^2 G^i}{\partial x^j \partial \dot{x}^k}\dot{x}^j + 2\frac{\partial^2 G^i}{\partial \dot{x}^j \dot{x}^k} G^j - \frac{\partial G^i}{\partial \dot{x}^j}\frac{\partial G^j}{\partial \dot{x}^k}, \tag{35}$$

where the G^i are given by $G^i = 1/2\ (\gamma^i_{jk}\ \dot{x}^j \dot{x}^k)$ and γ^i_{jk} are the usual Christoffel symbols of the second kind.

4 Theorems of Birkhoff Type.

Having the general expression of the 4-dimensional spherically symmetric Finsler metric as well as a generalisation of Einstein's vacuum field equations to Finsler spaces strongly motivates one to try to determine whether or not Birkhoff's theorem remains valid within this generalised geometry. But any casual attempt to perform the necessary operations in order to obtain the explicit expression of (34), even for the slightest departure from the Riemannian framework, will show that the calculations soon get large enough to be virtually impossible to be done by hand without inevitably introducing errors in the process. Here it helps to get acquainted with computer algebra packages. We have made use of `REDUCE` 3.4 [28] in order to construct the desired expression for H. We have written a series of straightforward `REDUCE` programs that perform the operations in a sequence of natural steps. The advantage of doing so is to avoid the overload of the memory space available within the package.

It is tempting, especially when using a computer to perform the calculations, to go for the most general case of interest, here given by the general spherically symmetric 4-dimensional Finsler metric [25]. But even computer algebra packages have their limitations – particularly related to memory space ([2]) – and, also having in mind that we want

[2]Even dividing the process in its most elementary steps and also defining all components of each tensor involved as scalars, we get very early in the process expressions of more than 1 Mbyte, for which it becomes virtually impossible to make use of standard procedures within `REDUCE`, such as taking the great common denominator (`gcd, ezgcd` switches), or factorising (`factor` switch) these expressions, even when a super-computer (namely, the *Convex* from the University of London's Computer Centre) was used.

to eventually obtain an equation for which it is feasible to actually find solutions, it is convenient to begin with a rather simple first order departure from the Riemannian case, such as the one considered in this section.

But, quite apart from these difficulties, we face yet another one, which has to do with the very nature of Finsler spaces. Since, unlike Riemannian geometry, the coefficients g_{ij} in (4) depend themselves in the variables dx^i, we can express a Finsler metric in many different ways. For instance, as pointed out in section 2 and discussed in more detail in [25], we can have the alternative forms (28) and (29) for the 4-dimensional spherically symmetric Finsler metric, and *which one we decide to choose will determine the kind of perturbation to be considered.*

The most natural, and yet general, form of the 4-dimensional spherically symmetric Finsler metric *for the purpose of making comparisons with its Riemannian equivalent* is given by

$$ds^2 = g_{11}\ dr^2 + g_{22}\ r^2 d\Omega^2 + g_{44}\ dt^2 + 2\ (\ g_{12}\ dr\ rd\Omega + g_{14}\ drdt + g_{24}\ rd\Omega dt\ ), \tag{36}$$

where $d\Omega = \sqrt{d\theta^2 + \sin^2\theta\ d\phi^2}$, and $g_{ij} = g_{ij}(r, t, \alpha, \beta)$; $i,\ j = 1,\ 2,\ 4$, with α and β given by (23). But the specific form by means of which the metric [25] function F depends on its directional variables is also arbitrary, apart from the homogeneity condition (2). This very condition, combined with the (physical) idea of directions at a point of space, naturally suggests the algebraically simpler, even if less general [25] dependence on *velocities*

$$A = d\Omega/dt, \quad B = dr/dt, \tag{37}$$

where we have maintained the combination $d\Omega = \sqrt{d\theta^2 + \sin^2\theta d\phi^2}$ in order to preserve spherical symmetry.

Having made choices (36), (37), let us now consider the subsequent metric in the first order of approximation, by taking the coefficients g_{ij} to be

$$g_{ij}(r, t, A, B) = g_{ij}(r, t) + \varepsilon \frac{\partial g_{ij}}{\partial A}(r, t) A + \varepsilon \frac{\partial g_{ij}}{\partial B}(r, t) B, \tag{38}$$

where we will consider the additional terms as small compared with the Riemannian ones, *i.e.*, disregard second or higher powers of the perturbation parameter ε. Rearranging terms, we get that

$$\begin{aligned} F^2 \ =\ & f_0\ dr^2 + \varepsilon\ f_1\ dr^2 \frac{d\Omega}{dt} + \varepsilon\ f_2 \frac{dr^3}{dt} + \qquad (39) \\ & + g_0\ d\Omega^2 + \varepsilon\ g_1\ \frac{d\Omega^3}{dt} + \varepsilon\ g_2\ \frac{dr d\Omega^2}{dt} + \\ & + h_0\ dt^2 + \varepsilon\ h_1\ d\Omega dt + \varepsilon\ h_2\ drdt + \end{aligned}$$

$$
\begin{aligned}
&+\varepsilon\ k_0\ drd\Omega + \varepsilon\ k_1\ \frac{drd\Omega^2}{dt} + \varepsilon\ k_2\ \frac{dr^2 d\Omega}{dt} + \\
&+l_0\ drdt + \varepsilon\ l_1\ drd\Omega + \varepsilon\ l_2\ dr^2 + \\
&+\varepsilon\ n_0\ d\Omega dt + \varepsilon\ n_1\ d\Omega^2 + \varepsilon\ n_2 drd\Omega,
\end{aligned}
$$

$$
\begin{aligned}
= \ & [f_0 + \varepsilon l_2]\ dr^2 + [\varepsilon f_1 + \varepsilon k_2]\ dr^2 \frac{d\Omega}{dt} + [\varepsilon f_2]\ dr^2 \frac{dr}{dt} + \\
&+[g_0 + \varepsilon n_1]\ d\Omega^2 + [\varepsilon g_1]\ d\Omega^2 \frac{d\Omega}{dt} + [\varepsilon g_2 + \varepsilon k_1]\ d\Omega^2 \frac{dr}{dt} + \\
&+[h_0]\ dt^2 + [\varepsilon h_1 + \varepsilon n_0]\ d\Omega dt + [\varepsilon h_2 + l_0]\ drdt + \\
&+[\varepsilon k_0 + \varepsilon l_1 + \varepsilon n_2]\ drd\Omega.
\end{aligned}
$$

Since all undetermined functions now depend only on (r, t), we can take advantage of the same coordinate freedom as in Riemannian geometry [29], and choose r, t such that $(g_0 + \varepsilon n_1) \equiv r^2$, $(\varepsilon h_2 + l_0) \equiv 0$. Renaming functions accordingly, we get

$$
\begin{aligned}
ds^2 \ = \ & F_0 dr^2 + \varepsilon\ F_1 A dr^2 + \varepsilon\ F_2 B dr^2 + \\
&+r^2 d\Omega^2 + G_1 A d\Omega^2 + \varepsilon\ G_2 B d\Omega^2 + \\
&+H_0 dt^2 + \varepsilon\ H_1 A dt^2 + \varepsilon\ H_2 A B dt^2,
\end{aligned} \tag{40}
$$

as the most general expression of a first order departure from the Riemannian spherically symmetric metric in 4 dimensions, with choices (36), (37). We have, therefore, 8 unknown functions of (positional) coordinates r, t to determine in order to solve equation (34).

Using (40) as input into the series of programs in REDUCE as already mentioned [30], we get an expression of 1230 terms for H, consisting of a numerator of 1207 terms and a denominator of 23 terms. These are all polynomial expressions in the variables $r, t, dr, d\theta, d\phi, dt$ and the unknown functions above, where second or higher order terms in ε have been disregarded.

First, we want to make sure that the denominator is not zero for any possible solution. This is given by

$$Q = X + \varepsilon Y, \tag{41}$$

where

$$X = 8{g_0}^3 {h_0}^3 r^3 d\Omega^2 dt^5 \tag{42}$$

and Y does involve all unknown functions. But even if Y were to be zero for some combinations of those, Q will not be zero as long as neither g_0 nor h_0 are zero. Furthermore, all terms in B do involve g_0 and/or h_0, and we can verify, by direct substitution, that Q will be made zero if either g_0 or h_0 are zero. So g_0 and h_0 cannot be zero, and that is enough to make Q nonzero too.

We can now concentrate in the numerator. Since we have expanded the functions $g_{ij}\,(x^k, dx^k)$ up to first order in dx^k, we have made its dependence in these coordinates explicit, and therefore, the remaining functions to be determined do not involve dx^k. We can then break this large polynomial expression into its coefficients independent of dx^k, by successively making each coordinate dr, dt, $d\Omega$ ([3]) to be zero, until we are left with expressions free of those coordinates, which we then call successively *eq1*, *eq2*, etc. Whenever we get an equation (that is, an expression independent of dx^i) or otherwise break into smaller parts a more complex expression, we subtract them from the original one, and carry on this process with the remaining terms. This process can always be done when perturbations of Riemannian metrics are being considered.

Explicitly in the case being dealt with here, let us first make $dr = 0$ (by making `let dr=0`). We are left with an expression of 275 terms which do not involve dr, from the 1207 terms in the numerator of H. Subtracting these from the original expression, 932 terms remain. This is our second break into full expression of H, after considering its numerator and denominator separately. Call the numerator of H by NH, and $NH1$, $NH2$, the expressions without and with dr in them, respectively. We can verify, by means of the command `factor`, which factorises its input, that $NH1$, besides not containing dr, has an overall factor dt^2, which can be eliminated from the expression, thus simplifying it. Of course, the simplified expression still has 275 terms. Let us then make dt zero (`let dt=0`) into expressions $NH1$ simplified, $NH2$. Only 24 terms in $NH1$ do not involve dt, and we can verify that it only has the remaining directional coordinates, namely $d\Omega$, as an overall factor, thus consisting on our first equation obtained from the full expression for H:

$$\begin{aligned} eq1 \;=\; & 4\left\{2\left[2\left(53g_0-52\right)f_1g_0-g_1\right]h_0-(\partial f_1/\partial r)g_0h_0r \right. \\ & \left. +106\partial g_0/\partial r f_1 h_0 r-108\partial h_0/\partial r f_1 g_0 r\right\} g_0h_0\varepsilon r^5, \end{aligned} \tag{43}$$

where the overall factor of $d\Omega^7$ has been eliminated. The expression $NH2$ has 10 terms which do not depend on dt, which we them call $NH3$. Subtracting $eq1 d\Omega^7$ from $NH1$, we get $NH4$, with 251 terms, and subtracting $NH3$ from $NH2$, we get $NH5$ with 922 terms. Expression $NH3$ has an overall factor of dr^4; once we eliminate it, we can again make dr zero (`let dr=0`) into it, thus getting

$$eq2 = -12(\partial g_0/\partial t)^2 f_1 {g_0}^3 \varepsilon r^5, \tag{44}$$

where an overall factor of $d\Omega^5$ has been eliminated. Subtracting $eq2 d\Omega^5$ from $NH3$ simplified, we get $NH6$, which has an overall factor dr which can be eliminated. Again

[3] We can verify that variables $d\theta$, $d\phi$ remain combined into $\sqrt{d\theta^2+\sin^2\theta d\phi^2}$ in the final expression for H.

making dr zero (`let dr=0`) into $NH6$ simplified, we get

$$eq3 = -12 d\Omega^4 (\partial g_0/\partial t)^2 f_2 {g_0}^3 \varepsilon r^5, \tag{45}$$

where an overall factor of $d\Omega^4$ has been eliminated. Subtracting $eq3 d\Omega^4$ from $NH6$ simplified we get $NH7$, which again has an overall factor dr, thus leading to

$$eq4 = -12 (\partial g_0/\partial t)^2 {g_0}^3 g_2 \varepsilon r^3, \tag{46}$$

which originally had $d\Omega^3$ as a factor, and, when subtracted from $NH7$ simplified, leads finally to

$$eq5 = -12 (\partial g_0/\partial t)^2 {g_0}^3 g_2 \varepsilon r^3, \tag{47}$$

from where factors $d\Omega^4$, dr have been eliminated. This way, by factorising, eliminating overall factors on dx^i and braking expressions by considering separately terms which do and do not depend on some dx^i, we have turned $NH3$, which is itself part of the whole denominator of the expression H for the metric (40), into 4 equations, namely $eq2$, $eq3$, $eq4$, $eq5$. It is completely equivalent to solving $NH3 = 0$ or the system of equations $eq2 = 0, \ldots, eq5 = 0$, since $eq2, \ldots, eq5$ are *coefficients* of $NH3$ with respect to variables dx^i.

In a completely similar way $NH4$ leads to a system of 6 equations, $eq6, \ldots, eq11$, and $NH5$ to another 29 equations, after some breaks like that which led to $NH1$ and $NH2$ in the first place. Whenever we get *two* overall factors, say dr and dt, we can break the simplified expression into three parts, namely, terms which involve dr, terms which involve dt and the remaining ones. The complete system of equations consisting of coefficients of the numerator of the expression (34) for the metric (40), can be found at the appendix.

So we have 40 non-linear differential equations for 8 unknown functions of the two variables r, t. This system is overdetermined exactly in the sense described by Wolf and Brand [31], who developed the computer algebra package `CRACK` which solves overdetermined systems of (non-linear) differential equations. But even for the purpose of using `CRACK` 40 equations are far too many; we can start by considering the smaller, simpler ones and then substituting the solutions into the remaining ones, in the hope of getting them simplified before taking them into account. If we use equations $eq1$ to $eq6$, $eq12$, $eq13$, $eq19$ to $eq25$, $eq34$ and $eq40$ into `CRACK`, we get 2 possible solutions:

$$f_1 = 0;\ f_2 = 0;\ g_1 = 0;\ g_2 = 0;\ \ g_0,\ h_0,\ h_1,\ h_2 \quad \text{free}, \tag{48}$$

and

$$g_0, h_0, f_1, h_1, h_2 \quad \text{free},$$

$$f_2 \quad \text{satisfying} \quad \left[2\left(\frac{\partial f_2}{\partial t}\right)h_0 + 2\left(\frac{\partial h_0}{\partial t}\right)f_2\right] = 0, \tag{49}$$

$$\begin{aligned} g_1 &= \left[106\left(\frac{\partial g_0}{\partial r}\right)f_1 h_0 r - \left(\frac{\partial f_1}{\partial r}\right)g_0 h_0 r + 212{g_0}^2 f_1 h_0 - 208 g_0 f_1 h_0 - \right. \\ &\left. -108\left(\frac{\partial h_0}{\partial r}\right)g_0 f_1 r\right], \\ g_2 &= \left[108\left(\frac{\partial g_0}{\partial r}\right)f_2 h_0 r - 3 f_2 h_0 r - 113\left(\frac{\partial h_0}{\partial r}\right)g_0 f_2 r + 212{g_0}^2 f_2 h_0 - 203 g_0 f_2 h_0\right], \end{aligned}$$

which we must then substitute into the remaining 23 equations. But, even substituting the simplest solution (48) into these, which leads to just 10 equations (13 are made zero), once we input those into `CRACK` the resulting system is complex enough for the process not to converge to a solution. `CRACK` keeps on dealing with ever more complex expressions, until the computer (a Sun Sparc station 10 with 90 Mbytes of RAM memory) halts on some step, possibly due to memory limitations.

We must then turn to a simpler case, at least as a first step. We know that, in order that the denominator not to be zero, g_0 and h_0 must not be zero, *i.e.*, we must have the full Riemannian part of the metric in our solutions. We also know (by the very theorem we would like to generalise to Finsler spaces, Birkhoff's theorem) that the Schwarzschild family of solutions, with a single parameter m, is the only solution, apart from coordinate transformations, of Einstein's vacuum field equations (which equation 34 generalises to Finsler spaces). Therefore, it is natural that we first try to simplify our equations by taking

$$g_0 = \left(1 - \frac{2m}{r}\right)^{-1}, \quad h_0 = -\left(1 - \frac{2m}{r}\right), \tag{50}$$

and therefore consider only solutions which are first order perturbations of this Riemannian metric. Once we are considering terms just up to first order in the perturbation parameter ε, the resulting equations will be linear differential equations on the remaining 6 undetermined functions. Once we substitute (50) into our 40 equations, we are left with just 27 equations. In particular,

$$eq6 = -12\,(\partial f_2/\partial t)\,\varepsilon r^6,$$

which already implies that $(\partial f_2/\partial t) = 0$, and we can thus further simplify the remaining equations. We then get that

$$eq8 = 24(2m - r)^2\,(\partial g_2/\partial t)\,\varepsilon r^2,$$

and so that, once we already have from the Schwarzschild metric being perturbed that $r \neq 2m$, $(\partial g_2/\partial t) = 0$. This simplifies even further the other equations, now just 20 to be considered. Of those, 3 have just $g_1(r,t)$ as unknown function, namely, $eq32$, $eq35$ and $eq36$. Applying `CRACK` on those leads to $g_1 = 0$ as only solution. But 5 equations ($eq1$, $eq7$, $eq14$, $eq26$ and $eq28$) had only f_1 and g_1 as undetermined functions, so, once considering $g_1 = 0$ and applying `CRACK` on them too, we get that $f_1 = 0$ is the only solution, too. We can also apply `CRACK` to equations $eq13$, $eq15$, $eq17$, $eq34$ and $eq37$, which have f_2 and g_2 as undetermined functions, to get that they too can only be zero as well, $g_2 = f_2 = 0$. Of the remaining 4 equations ($eq33$ had only g_2 as unknown function and is already automatically solved), 3 ($eq11$, $eq18$, $eq30$) had g_1, h_1 and h_2 as unknown functions (now just h_1 and h_2) and $eq38$ only h_1. These, once put into `CRACK` finally lead us to

$$h_1 = \varepsilon\left(1 - \frac{2m}{r}\right), \quad h_2 = 0, \tag{51}$$

which leads us to the perturbative solution

$$ds^2 = \left(1 - \frac{2m}{r}\right)^{-1} dr^2 + r^2 d\Omega^2 - \left(1 - \frac{2m}{r}\right)\left(1 + \varepsilon\frac{d\Omega}{dt}\right) dt^2. \tag{52}$$

This solution has already been found to be unique, as well as time-independent, for two other different, less general, starting expressions for the first order perturbation of the Riemannian spherically symmetric metric [18, 32], in both cases the Riemannian functions having been *obtained* rather than imposed. These are all restricted versions of a first order, perturbative, generalisation of Birkhoff's theorem to Finsler spaces, and they do suggest that, in these generalised spaces, the solution to the generalised vacuum filed equation in the spherically symmetric case[4] is unique (up to coordinate transformations) and time-independent, as in the Riemannian framework.

5 Conclusion.

In this paper we have established a perturbative version of Birkhoff's theorem from General Relativity for Finsler spaces, considering first order departures from the Riemannian 4-dimensional spherically symmetric metric which preserve its symmetry. As arbitrary choices in this process, we took the form of the Finsler 4-dimensional spherically symmetric metric closest to its Riemannian counterpart; the directional dependence of such metric to be given in terms of velocities (dr/dt), $(d\Omega/dt)$, where $d\Omega = \sqrt{d\theta^2 + \sin^2\theta d\phi^2}$;

[4]Which leads to the simplest model for the gravitational field of a star, and thus to observational predictions to orbits of planets or paths of light rays within the solar system [18].

and finally, the Riemannian part of the metric to be given by the Schwarzschild solution to Einstein's vacuum field equations.

Computer algebra played an essential role all along the process (described in detail within the paper), firstly by determining the expression of the generalised field equation for the chosen metric, then to break this equation into a system of differential equations, and finally in solving this system to find the unique time-independent solution which configures the extension of Birkhoff's theorem to the Finsler setting.

Of course, Finsler geometry does have many other fields of applications [33] besides geometrical extensions of theories of gravity, and, in the hope that computer algebra can be as helpful as it has been here, the programs developed to give Finslerian expressions from a chosen metric or connection are being made into a package, soon to be released.

Acknowledgements

I am grateful to Edgardo Ched-Terrab for the encouragement to publish this work in this special volume of Computer Physics Communications, to Antonio F.F. Teixeira for his support while I was a visitor at CBPF, to my husband Marcelo and my mother-in-law for all the help while I was writing this work. Finally, the financial support from FAPERJ is acknowledged.

References

[1] Kilmister, D. A. and Stephenson, G.: An Axiomatic Criticism of Unified Field Theories – I, II. (Il Nuovo Cimento **11** (1954), Suppl. 91-105, 118–140).

[2] Finsler, P.: Über Kurven und Flächen in allgemeinen Räumen. Dissertation, Göttingen (1918).

[3] Riemann, B.: Über die Hypothesen, welche der Geometrie zugrunde liegen. (Habilitationsvortrag, 1854. Werke, 272–287, Leipzig, 1892).

[4] Matsumoto, M.: On Einstein's Gravitational Field Equation in a Tangent Riemannian Space of a Finsler Space. (Reports on Mathematical Physics **8** (1975) 103–108).

[5] Bogoslovsky, G. Yu.: A Special-Relativistic Theory of the Locally Anisotropic Space-Time, I, II. (Il Nuovo Cimento **40 B** 1 (1977) 99–115, 116–134).

[6] Bogoslovsky, G. Yu.: From the Weyl Theory to a Locally Anisotropic Spacetime. (Classical and Quantum Gravity **9** (1992) 569–575).

[7] Ishikawa, H.: Note on Finslerian Relativity. (Journal of Mathematical Physics **22** 5 (1981) 995–1004).

[8] Tavakol, R. K. and Van der Bergh, N.: Viability Criteria for the Theories of Gravity and Finsler Spaces. (General Relativity and Gravitation **18** (1986), 849–859).

[9] Takano, Y.: Gravitational Field in Finsler Spaces. (Lettere al Nuovo Cimento **10** 17 (1974) 747–750).

[10] Matsumoto, M.: Foundations of Finsler Geometry and Special Finsler Spaces. (Kaiseisha Press, 1986).

[11] Walker, A. G.: Completely Symmetric Spaces. (Journal of the London Mathematical Society **19** (1944), 219–226).

[12] Asanov, G. S.: On Finslerian Relativity. (Il Nuovo Cimento **49** (1979), 221–246).

[13] Roxburgh, I. W. and Tavakol, R. K.: Non-Riemann Geometrizable Effects in the Gravitational One-Body Problem. (General Relativity and Gravitation **10** 4 (1979) 307–312).

[14] Coley, A. A.: Clocks and Gravity. (General Relativity and Gravitation **14** (1982), 1107–1114).

[15] Aringazin, A. K. and Asanov, G. S.: Finslerian Post-Riemannian Corrections to the Equations of Geodesics. (General Relativity and Gravitation **17** 12 (1985) 1153–1163).

[16] Roxburgh, I. W.: Finsler Spaces with Riemannian Geodesics. (General Relativity and Gravitation **23** 9 (1991) 1071–1080).

[17] Roxburgh, I. W.: Post Newtonian Limit of Finsler Space Theories of Gravity and Solar System Tests. (General Relativity and Gravitation **24** (1992) 419–431).

[18] Rutz, S. F.: A Finsler Generalisation of Einstein's Vacuum Field Equations. (General Relativity and Gravitation **25** (1993) 1139–1158).

[19] D'Inverno, R.: Introducing Einstein's Relativity. (Clarendon Press, Oxford, 1992).

[20] Rund, H.: The Differential Geometry of Finsler Spaces. (Springer-Verlag, Berlin, 1959).

[21] Berwald, L.: Über Finslersche und Cartansche Geometrie I (Mathematica, Timisoara **17** (1941), 34–55).

[22] Cartan, É.: Les Espaces de Finsler. (Actualités 79, Paris, 1934).

[23] Bao, D.; Chern, S. S.: On a Notable Connection in Finsler Geometry. (Houston Journal of Mathematics **19** 1 (1993) 135–180).

[24] Yano, K.: The Theory of Lie Derivatives and Its Applications. (North-Holland, Amsterdam, 1955).

[25] McCarthy, P. J.; Rutz, S. F.: The General Four-dimensional Spherically Symmetric Finsler Space. (General Relativity and Gravitation **25** (1993) 589–602).

[26] Pirani, F. A. E. in Lectures on General Relativity, Brandeis Summer Institute in Theoretical Physics, 1964, volume one, lecture 2. Edited by S. Deser and K. W. Ford. (Prentice-Hall, Englewood Cliffs, 1965).

[27] Berwald, L.: Projektivkrümmung allgemeiner affiner Räume und Finslersche Räume skalarer Krümmung. (Ann. Math. (2) **48**, 755–781 (1947)).

[28] Hearn, A. C.: `REDUCE` User's Manual Version 3.4. (Rand, Santa Monica, 1991).

[29] Landau, L. D. and Lifshitz, E. M.: The Classical Theory of Fields. Pergamon, Oxford, London, Paris, Frankfurt. Revised 2nd. Ed. (1962).

[30] Rutz, S. F.: Symmetry and Gravity in Finsler Spaces, PhD thesis, Queen Mary & Westfield College, University of London (1993).

[31] Wolf, T. and Brand, A.: The Computer Algebra Package `CRACK` for Investigating PDEs, `REDUCE` library: documentation in LaTeXwithin version 3.5 (1997).

[32] Rutz, S. F.: Symmetry in Finsler Spaces. (In [33], 289–300).

[33] Bao, D., Chern, S. S., Shen, Z. (editors): Finsler Geometry. (Contemporary Mathematics **196**, American Mathematical Society, Providence, 1996).

A The Equations.

```
eq1 := 4*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*(2*(53*g0 - 52)*f1*g0 - g1)*h0 - df(f1,r)*g0*h0*
r + 106*df(g0,r)*f1*h0*r - 108*df(h0,r)*f1*g0*r)*(sin(theta)**2*dphi**2 + dtheta**2)**3*g0*h0*epsilon
*r**5$

eq2 := - 12*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(sin(theta)**2*dphi**2 + dtheta**2)**2*df(g0,t)
**2*f1*g0**3*epsilon*r**5$

eq3 := - 12*(sin(theta)**2*dphi**2 + dtheta**2)**2*df(g0,t)**2*f2*g0**3*epsilon*r**5$

eq4 := - 12*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(sin(theta)**2*dphi**2 + dtheta**2)*df(g0,t)**2
*g0**3*g1*epsilon*r**3$

eq5 := - 12*(sin(theta)**2*dphi**2 + dtheta**2)*df(g0,t)**2*g0**3*g2*epsilon*dr*r**3$

eq6 := - 2*(6*df(f2,t)*g0*h0 - 5*df(g0,t)*f2*h0 + 3*df(h0,t)*f2*g0)*(sin(theta)**2*dphi**2 + dtheta
**2)**3*g0*h0*epsilon*r**6$

eq7 := 2*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*(2*(239*f1*g0 + 52*g1)*h0 + 53*df(h0,r)*f1*g0*r)
*df(g0,r)*h0**2*r + 2*(3*(318*g0 - 317)*f1*g0 + 212*g0*g1 - 209*g1)*g0*h0**3 + (df(g0,r)*h0*r - 4*df
(h0,r)*g0*r - 9*g0*h0)*df(f1,r)*g0*h0**2*r - (2*df(g0,t)*h0 + 3*df(h0,t)*g0)*df(f1,t)*g0**2*h0*r**2
- 27*(51*f1*g0 + 8*g1)*df(h0,r)*g0*h0**2*r - 2*df(f1,r,2)*g0**2*h0**3*r**2 - 6*df(f1,t,2)*g0**3*h0**
2*r**2 - 212*df(g0,t,2)*f1*g0**2*h0**2*r**2 + 106*df(g0,t)**2*f1*g0*h0**2*r**2 + 105*df(g0,t)*df(h0,
t)*f1*g0**2*h0*r**2 + 2*df(g1,r)*g0*h0**3*r - 212*df(h0,r,2)*f1*g0**2*h0**2*r**2 + 102*df(h0,r)**2*
f1*g0**2*h0*r**2 - 3*df(h0,t,2)*f1*g0**3*h0*r**2 + 3*df(h0,t)**2*f1*g0**3*r**2)*(sin(theta)**2*dphi
**2 + dtheta**2)**2*epsilon*r**3$

eq8 := - (2*(df(g0,r)*h0 + 4*df(h0,r)*g0)*df(f2,t)*g0*h0*epsilon*r**2 - (5*df(h0,r)*f2*g0*r + 16*f2
*g0*h0 + 36*g2*h0)*df(g0,t)*h0*epsilon*r + 2*(df(h0,t)*f2*epsilon*r + 2*h0)*df(h0,r)*g0**2*r + (df(
h0,t)*f2*epsilon*r - 4*h0)*df(g0,r)*g0*h0*r - 4*df(f2,r,t)*g0**2*h0**2*epsilon*r**2 - 2*df(f2,r)*df(
h0,t)*g0**2*h0*epsilon*r**2 + 24*df(g2,t)*g0*h0**2*epsilon*r + 2*df(h0,r,t)*f2*g0**2*h0*epsilon*r**2
+ 12*df(h0,t)*g0*g2*h0*epsilon*r - 8*g0**3*h0**2 + 8*g0**2*h0**2)*(sin(theta)**2*dphi**2 + dtheta**2
)**2*h0*r**3$

eq9 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*((477*f1*g0 + 104*g1)*df(h0,r)*r + df(h1,r)*g0*r -
df(h2,t)*g0*r + 106*g0*h1)*df(g0,r)*h0**2*r + 2*(53*(9*f1*g0 + 2*g1)*df(h0,t)*r + 2*df(g1,t)*h0*r +
2*h0*h2)*df(g0,t)*g0*h0*r - 12*(df(g0,t,2) + df(h0,r,2))*(159*f1*g0 + 35*g1)*g0*h0**2*r**2 - (df(h0,
t,2)*h0 - df(h0,t)**2)*(9*f1*g0 + 2*g1)*g0**2*r**2 + 2*(df(h1,r)*g0*r - df(h2,t)*g0*r - 1899*f1*g0*
h0 - 109*g0*h1 - 420*g1*h0)*df(h0,r)*g0*h0*r + (945*f1*g0 + 206*g1)*df(h0,r)**2*g0*h0*r**2 + 2*(477*
f1*g0 + 104*g1)*df(g0,t)**2*h0**2*r**2 + 424*(g0 - 1)*g0**2*h0**2*h1 - 9*df(f1,r)*df(h0,r)*g0**2*h0
**2*r**2 - 18*df(f1,t,2)*g0**3*h0**2*r**2 - 9*df(f1,t)*df(h0,t)*g0**3*h0*r**2 + 2*df(g1,r)*df(h0,r)*
g0*h0**2*r**2 - 4*df(g1,t,2)*g0**2*h0**2*r**2 - 2*df(g1,t)*df(h0,t)*g0**2*h0*r**2 - 4*df(h1,r,2)*g0
**2*h0**2*r**2 - 2*df(h1,r)*g0**2*h0**2*r + 4*df(h2,r,t)*g0**2*h0**2*r**2 + 4*df(h2,t)*g0**2*h0**2*r
)*(sin(theta)**2*dphi**2 + dtheta**2)*h0*epsilon*r$

eq10 := 2*(((4*f2*g0 + 9*g2)*df(h0,r)*h0*epsilon + df(h0,t)*g0**2)*df(g0,t)*r - (2*f2*g0 + 3*g2)*df(
h0,r,t)*g0*h0*epsilon*r - 4*df(f2,t)*df(h0,r)*g0**2*h0*epsilon*r + df(g0,r)*df(h0,r)*g0*h0*r - 2*df(
g0,t,2)*g0**2*h0*r + df(g0,t)**2*g0*h0*r - 6*df(g2,t)*df(h0,r)*g0*h0*epsilon*r - 2*df(h0,r,2)*g0**2*
h0*r + df(h0,r)**2*g0**2*r - 4*df(h0,r)*g0**2*h0)*(sin(theta)**2*dphi**2 + dtheta**2)*h0**2*r**2$

eq11 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(106*df(g0,r)*df(h0,r)*h0**2*h1*r - 212*df(g0,t,2)*
```

```
g0*h0**2*h1*r + 106*df(g0,t)**2*h0**2*h1*r + 2*df(g0,t)*df(h0,r)*h0**2*h2*r + 106*df(g0,t)*df(h0,t)*
g0*h0*h1*r - df(h0,r,t)*g0*h0**2*h2*r - 212*df(h0,r,2)*g0*h0**2*h1*r + 107*df(h0,r)**2*g0*h0*h1*r -
df(h0,r)**2*g1*h0**2*r + df(h0,r)*df(h0,t)*g0*h0*h2*r - df(h0,r)*df(h1,r)*g0*h0**2*r - 2*df(h0,r)*df
(h2,t)*g0*h0**2*r - 424*df(h0,r)*g0*h0**2*h1 + df(h0,t,2)*g0**2*h0*h1*r - 2*df(h0,t)**2*g0**2*h1*r +
3*df(h0,t)*df(h1,t)*g0**2*h0*r - 2*df(h1,t,2)*g0**2*h0**2*r)*ct*g0*h0*epsilon$

eq12 := 12*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(sin(theta)**2*dphi**2 + dtheta**2)**3*df(g0,t)*
f1*g0**2*h0*epsilon*r**6$

eq13 := 4*((212*g0 - 203)*f2*g0*h0 - 3*df(f2,r)*g0*h0*r + 108*df(g0,r)*f2*h0*r - 113*df(h0,r)*f2*g0*
r - 6*g2*h0)*(sin(theta)**2*dphi**2 + dtheta**2)**3*g0*h0*epsilon*r**5$

eq14 := 2*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((5*(169*f1*g0 + 4*g1)*h0 + 9*df(h0,r)*f1*g0*r)*df
(g0,t)*h0 + 3*(df(g0,t)*h0 - df(h0,t)*g0)*df(f1,r)*g0*h0*r - 12*(df(h0,r)*r - 2*h0)*df(f1,t)*g0**2*
h0 + 6*(2*f1*g0 - g1)*df(h0,t)*g0*h0 - 6*df(f1,r,t)*g0**2*h0**2*r - 12*df(g1,t)*g0*h0**2 - 9*df(h0,r
,t)*f1*g0**2*h0*r + 3*df(h0,r)*df(h0,t)*f1*g0**2*r)*(sin(theta)**2*dphi**2 + dtheta**2)**2*g0*epsilon
*r**4$

eq15 := 2*(3*(2*df(g0,t)*h0 - df(h0,t)*g0)*df(f2,t)*g0**2*h0*r**2 + (107*df(h0,r)*f2*g0*r + 436*f2*
g0*h0 + 642*g2*h0)*df(g0,r)*h0**2*r - 3*(df(h0,r)*r + 4*h0)*df(f2,r)*g0**2*h0**2*r - 6*(141*f2*g0 +
110*g2)*df(h0,r)*g0*h0**2*r + 4*(212*g0 - 215)*f2*g0**2*h0**3 - 6*df(f2,t,2)*g0**3*h0**2*r**2 - 208*
df(g0,t,2)*f2*g0**2*h0**2*r**2 + 101*df(g0,t)**2*f2*g0*h0**2*r**2 + 107*df(g0,t)*df(h0,t)*f2*g0**2*
h0*r**2 - 6*df(g2,r)*g0*h0**3*r - 211*df(h0,r,2)*f2*g0**2*h0**2*r**2 + 98*df(h0,r)**2*f2*g0**2*h0*r
**2 - 3*df(h0,t,2)*f2*g0**3*h0*r**2 + 3*df(h0,t)**2*f2*g0**3*r**2 + 1272*g0**2*g2*h0**3 - 1242*g0*g2
*h0**3)*(sin(theta)**2*dphi**2 + dtheta**2)**2*epsilon*r**3$

eq16 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(((27*f1*g0 + 28*g1)*df(h0,r)*r - 2*df(h1,r)*g0*r +
2*df(h2,t)*g0*r + 7596*f1*g0*h0 + 2*g0*h1 + 1680*g1*h0)*df(g0,t)*h0*r + 18*(df(g0,t)*h0 - df(h0,t)*
g0)*df(f1,r)*g0*h0*r**2 - 4*(5*df(h0,r)*r - 6*h0)*df(g1,t)*g0*h0*r - 18*(df(h0,r)*r - 2*h0)*df(f1,t)
*g0**2*h0*r + 18*(df(h0,t)*f1*g0*r - 12*h0*h2)*df(h0,r)*g0*r - 2*(df(h1,r)*g0*r - df(h2,t)*g0*r - 9*
f1*g0*h0 - 2*g0*h1 - 6*g1*h0)*df(h0,t)*g0*r - (27*f1*g0 + 10*g1)*df(h0,r,t)*g0*h0*r**2 + 2*(212*g0 -
215)*g0*h0**2*h2 - 36*df(f1,r,t)*g0**2*h0**2*r**2 + 220*df(g0,r)*h0**2*h2*r + 4*df(h1,r,t)*g0**2*h0*
r**2 - 8*df(h1,t)*g0**2*h0*r - 6*df(h2,r)*g0*h0**2*r - 4*df(h2,t,2)*g0**2*h0*r**2)*(sin(theta)**2*dphi
**2 + dtheta**2)*g0*h0*epsilon*r$

eq17 := 2*((107*(2*f2*g0 + 3*g2)*df(h0,t)*epsilon*r + 18*df(g2,t)*h0*epsilon*r + 8*g0*h0)*df(g0,t)*
g0*h0 + 2*(4*df(g0,t)*h0 - df(h0,t)*g0)*df(f2,t)*g0**2*h0*epsilon*r - (df(h0,t,2)*h0 - df(h0,t)**2)*
(2*f2*g0 + 3*g2)*g0**2*epsilon*r - 3*(142*f2*g0 + 211*g2)*df(h0,r,2)*g0*h0**2*epsilon*r + 3*(72*f2*
g0 + 107*g2)*df(g0,r)*df(h0,r)*h0**2*epsilon*r - 12*(71*f2*g0 + 105*g2)*df(h0,r)*g0*h0**2*epsilon +
3*(70*f2*g0 + 103*g2)*df(h0,r)**2*g0*h0*epsilon*r - 12*(35*f2*g0 + 52*g2)*df(g0,t,2)*g0*h0**2*epsilon
*r + 6*(34*f2*g0 + 49*g2)*df(g0,t)**2*h0**2*epsilon*r - 6*df(f2,r)*df(h0,r)*g0**2*h0**2*epsilon*r -
4*df(f2,t,2)*g0**3*h0**2*epsilon*r - 3*df(g2,r)*df(h0,r)*g0*h0**2*epsilon*r - 6*df(g2,t,2)*g0**2*h0
**2*epsilon*r - 3*df(g2,t)*df(h0,t)*g0**2*h0*epsilon*r)*(sin(theta)**2*dphi**2 + dtheta**2)*h0*r**2$

eq18 := - sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(((3*g0*h1 - 8*g1*h0)*df(h0,r)*r - 105*df(h0,t)*
g0*h2*r - 2*df(h1,r)*g0*h0*r - 4*df(h2,t)*g0*h0*r - 848*g0*h0*h1)*df(g0,t)*h0 + 3*(2*df(h0,t)*g0*h1*
r - 2*df(h1,t)*g0*h0*r + df(h2,r)*h0**2*r + 142*h0**2*h2)*df(h0,r)*g0 - (2*df(h1,r)*r + df(h2,t)*r +
2*h1)*df(h0,t)*g0**2*h0 - (3*g0*h1 - 2*g1*h0)*df(h0,r,t)*g0*h0*r - 108*df(g0,r)*df(h0,r)*h0**2*h2*r
+ 210*df(g0,t,2)*g0*h0**2*h2*r - 102*df(g0,t)**2*h0**2*h2*r + 4*df(g1,t)*df(h0,r)*g0*h0**2*r + 213*
df(h0,r,2)*g0*h0**2*h2*r - 108*df(h0,r)**2*g0*h0*h2*r + 4*df(h1,r,t)*g0**2*h0**2*r + 4*df(h1,t)*g0**
2*h0**2 + 2*df(h2,t,2)*g0**2*h0**2*r)*g0*h0*epsilon$
```

```
eq19 := 12*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(sin(theta)**2*dphi**2 + dtheta**2)**3*df(g0,t)*
f1*g0**2*h0*epsilon*r**6$

eq20 := 24*(sin(theta)**2*dphi**2 + dtheta**2)**3*df(g0,t)*f2*g0**2*h0*epsilon*r**6$

eq21 := - 6*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((2*(4*f1*g0 - 3*g1)*h0 - 5*df(h0,r)*f1*g0*r)*
df(g0,t) - 2*df(f1,r)*df(g0,t)*g0*h0*r - df(g0,r,t)*f1*g0*h0*r + df(g0,r)*df(g0,t)*f1*h0*r)*(sin(theta
)**2*dphi**2 + dtheta**2)**2*g0**2*epsilon*r**4$

eq22 := 6*((2*df(f2,r)*df(g0,t)*g0 + df(g0,r,t)*f2*g0 - 2*df(g0,r)*df(g0,t)*f2)*h0*r + (5*df(h0,r)*
f2*g0*r - 4*f2*g0*h0 + 8*g2*h0)*df(g0,t))*(sin(theta)**2*dphi**2 + dtheta**2)**2*g0**2*epsilon*r**4$

eq23 := 6*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((2*df(g1,r)*h0*r + 5*df(h0,r)*g1*r - 4*g1*h0)*df(
g0,t)*g0 + df(g0,r,t)*g0*g1*h0*r - 3*df(g0,r)*df(g0,t)*g1*h0*r)*(sin(theta)**2*dphi**2 + dtheta**2)*
g0**2*epsilon*r**2$

eq24 := 6*(df(g0,r,t)*g0*g2*h0 - 4*df(g0,r)*df(g0,t)*g2*h0 + 2*df(g0,t)*df(g2,r)*g0*h0 + 5*df(g0,t)*
df(h0,r)*g0*g2)*(sin(theta)**2*dphi**2 + dtheta**2)*g0**2*epsilon*dr*r**3$

eq25 := - 12*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(sin(theta)**2*dphi**2 + dtheta**2)**3*df(g0,t
)*f1*g0**2*h0*epsilon*r**6$

eq26 := 2*sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(4*(7*(15*f1*g0 + 8*g1)*h0 + 27*df(h0,r)*f1*g0*r)*
df(g0,r)*h0*r + (4*(10*f1*g0 - 59*g1)*df(h0,r)*h0*r - 8*(5*f1*g0 - 53*g0*g1 + 49*g1)*h0**2 - 2*df(f1
,r,2)*g0*h0**2*r**2 + 7*df(f1,t)*df(g0,t)*g0*h0*r**2 - 207*df(g0,t,2)*f1*g0*h0*r**2 + 99*df(g0,t)**2
*f1*h0*r**2 + 107*df(g0,t)*df(h0,t)*f1*g0*r**2 - 10*df(g1,r)*h0**2*r - 216*df(h0,r,2)*f1*g0*h0*r**2
+ 98*df(h0,r)**2*f1*g0*r**2)*g0 + (df(g0,r)*h0*r - 10*df(h0,r)*g0*r + 16*g0*h0)*df(f1,r)*g0*h0*r)*(
sin(theta)**2*dphi**2 + dtheta**2)**2*g0*epsilon*r**3$

eq27 := (6*((2*f2*g0 - 3*g2)*df(h0,t)*h0 - 2*df(f2,r,t)*g0*h0**2*r + df(g0,r,t)*f2*h0**2*r - 6*df(g2
,t)*h0**2 - 3*df(h0,r,t)*f2*g0*h0*r + df(h0,r)*df(h0,t)*f2*g0*r)*g0 + 6*(df(g0,r)*h0*r - 4*df(h0,r)*
g0*r + 4*g0*h0)*df(f2,t)*g0*h0 - 3*(4*df(g0,t)*h0 - df(h0,t)*g0)*df(g0,r)*f2*h0*r + 6*(2*df(g0,t)*h0
- df(h0,t)*g0)*df(f2,r)*g0*h0*r + (39*df(h0,r)*f2*g0*r + 1676*f2*g0*h0 + 102*g2*h0)*df(g0,t)*h0)*(sin
(theta)**2*dphi**2 + dtheta**2)**2*g0*epsilon*r**4$

eq28 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((2*(1899*f1*g0 + 530*g1)*h0 + (963*f1*g0 + 436*g1)*
df(h0,r)*r)*df(g0,r)*h0**2*r + 2*((477*f1*g0 + 215*g1)*df(h0,t) + 15*df(g1,t)*h0)*df(g0,t)*g0*h0*r**
2 + 9*(df(g0,r)*h0*r - 4*df(h0,r)*g0*r + 4*g0*h0)*df(f1,r)*g0*h0**2*r - 6*(2*df(h0,r)*r - h0)*df(g1,
r)*g0*h0**2*r - (1899*f1*g0 + 830*g1)*df(g0,t,2)*g0*h0**2*r**2 - 2*(963*f1*g0 + 424*g1)*df(h0,r,2)*
g0*h0**2*r**2 + 2*(477*f1*g0 + 200*g1)*df(h0,r)**2*g0*h0*r**2 + 4*(234*f1*g0 + 97*g1)*df(g0,t)**2*h0
**2*r**2 + 2*(18*f1*g0 - 509*g1)*df(h0,r)*g0*h0**2*r - 4*(9*f1*g0 - 106*g0*g1 + 115*g1)*g0*h0**3 -
18*df(f1,r,2)*g0**2*h0**3*r**2 + 9*df(f1,t)*df(g0,t)*g0**2*h0**2*r**2 - 12*df(g1,t,2)*g0**2*h0**2*r
**2 - 6*df(g1,t)*df(h0,t)*g0**2*h0*r**2 - 6*df(h0,t,2)*g0**2*g1*h0*r**2 + 6*df(h0,t)**2*g0**2*g1*r**
2)*(sin(theta)**2*dphi**2 + dtheta**2)*g0*epsilon*r$

eq29 := (((20*f2*g0 + 81*g2)*df(h0,r)*h0*epsilon*r + 24*df(g2,r)*h0**2*epsilon*r + 2*df(h0,t)*g0**2*
r + 3408*f2*g0*h0**2*epsilon + 5040*g2*h0**2*epsilon)*df(g0,t)*g0 - (6*(4*f2*g0 + 9*g2)*df(g0,t)*h0*
epsilon*r - (4*f2*g0 + 9*g2)*df(h0,t)*g0*epsilon*r - 18*df(g2,t)*g0*h0*epsilon*r - 2*df(h0,r)*g0**2*
r - 8*g0**2*h0)*df(g0,r)*h0 + 8*(df(g0,r)*h0*r - df(h0,r)*g0*r - 2*g0*h0)*df(f2,t)*g0**2*h0*epsilon
+ 8*(3*df(g0,t)*h0 - df(h0,t)*g0)*df(f2,r)*g0**2*h0*epsilon*r - 12*(3*df(h0,r)*r - 2*h0)*df(g2,t)*g0
**2*h0*epsilon + 2*(4*f2*g0 + 3*g2)*df(h0,r)*df(h0,t)*g0**2*epsilon*r + 6*(2*f2*g0 + 3*g2)*df(g0,r,t
)*g0*h0**2*epsilon*r - 4*(2*f2*g0 - 3*g2)*df(h0,t)*g0**2*h0*epsilon - 12*(f2*g0 + 2*g2)*df(h0,r,t)*
```

```
g0**2*h0*epsilon*r - 16*df(f2,r,t)*g0**3*h0**2*epsilon*r - 4*df(g0,t,2)*g0**3*h0*r + 2*df(g0,t)**2*
g0**2*h0*r - 12*df(g2,r,t)*g0**2*h0**2*epsilon*r - 6*df(g2,r)*df(h0,t)*g0**2*h0*epsilon*r - 4*df(h0,
r,2)*g0**3*h0*r + 2*df(h0,r)**2*g0**3*r)*(sin(theta)**2*dphi**2 + dtheta**2)*h0*r**2$

eq30 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((7*(15*g0*h1 + 16*g1*h0)*df(h0,r)*r - 6*df(g0,t)*h0
*h2*r + df(h1,r)*g0*h0*r + 2*df(h2,t)*g0*h0*r + 424*g0*h0*h1)*df(g0,r)*h0 + (108*(g0*h1 + g1*h0)*df(
h0,t)*r + 8*df(g1,t)*h0**2*r - 3*df(h0,r)*h0*h2*r - 3*df(h1,t)*g0*h0*r + 6*df(h2,r)*h0**2*r + 852*h0
**2*h2)*df(g0,t)*g0 + 2*(2*df(h1,r)*g0*r + df(h2,t)*g0*r + 2*g0*h1 - 214*g1*h0)*df(h0,r)*g0*h0 - (213
*g0*h1 + 208*g1*h0)*df(g0,t,2)*g0*h0*r - 2*(105*g0*h1 + 107*g1*h0)*df(h0,r,2)*g0*h0*r + 2*(54*g0*h1
+ 47*g1*h0)*df(g0,t)**2*h0*r + 3*(34*g0*h1 + 35*g1*h0)*df(h0,r)**2*g0*r + 3*df(g0,r,t)*g0*h0**2*h2*r
- 5*df(g1,r)*df(h0,r)*g0*h0**2*r - 2*df(g1,t,2)*g0**2*h0**2*r - df(g1,t)*df(h0,t)*g0**2*h0*r - df(h0
,t,2)*g0**2*g1*h0*r + df(h0,t)**2*g0**2*g1*r - 2*df(h1,r,2)*g0**2*h0**2*r - 4*df(h1,r)*g0**2*h0**2
- 4*df(h2,r,t)*g0**2*h0**2*r - 4*df(h2,t)*g0**2*h0**2)*g0*h0*epsilon$

eq31 := - 2*(2*df(f2,r,2)*g0**2*h0**2*r**2 - 3*df(f2,r)*df(g0,r)*g0*h0**2*r**2 + 10*df(f2,r)*df(h0,
r)*g0**2*h0*r**2 - 8*df(f2,r)*g0**2*h0**2*r - 7*df(f2,t)*df(g0,t)*g0**2*h0*r**2 - df(g0,r,2)*f2*g0*
h0**2*r**2 + 2*df(g0,r)**2*f2*h0**2*r**2 - 113*df(g0,r)*df(h0,r)*f2*g0*h0*r**2 - 418*df(g0,r)*f2*g0*
h0**2*r - 236*df(g0,r)*g2*h0**2*r + 207*df(g0,t,2)*f2*g0**2*h0*r**2 - 94*df(g0,t)**2*f2*g0*h0*r**2 -
107*df(g0,t)*df(h0,t)*f2*g0**2*r**2 + 14*df(g2,r)*g0*h0**2*r + 216*df(h0,r,2)*f2*g0**2*h0*r**2 - 98*
df(h0,r)**2*f2*g0**2*r**2 - 20*df(h0,r)*f2*g0**2*h0*r + 246*df(h0,r)*g0*g2*h0*r + 12*f2*g0**2*h0**2
- 424*g0**2*g2*h0**2 + 406*g0*g2*h0**2)*(sin(theta)**2*dphi**2 + dtheta**2)**2*g0*epsilon*r**3$

eq32 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*(5*df(g1,r)*h0*r + 118*df(h0,r)*g1*r + 414*g1*h0)
*df(g0,r)*g0*h0*r + 2*(7*df(g1,t)*h0 + 107*df(h0,t)*g1)*df(g0,t)*g0**2*r**2 - 4*(5*df(h0,r)*r - 4*h0
)*df(g1,r)*g0**2*h0*r - (9*f1*g0 - 176*g1)*df(g0,t)**2*g0*h0*r**2 + 4*df(g0,r,2)*g0*g1*h0**2*r**2 -
10*df(g0,r)**2*g1*h0**2*r**2 - 414*df(g0,t,2)*g0**2*g1*h0*r**2 - 4*df(g1,r,2)*g0**2*h0**2*r**2 - 432
*df(h0,r,2)*g0**2*g1*h0*r**2 + 196*df(h0,r)**2*g0**2*g1*r**2 + 40*df(h0,r)*g0**2*g1*h0*r - 24*g0**2*
g1*h0**2)*(sin(theta)**2*dphi**2 + dtheta**2)*g0*epsilon*r$

eq33 := 2*((7*df(g2,r)*h0*r + 123*df(h0,r)*g2*r + 424*g2*h0)*df(g0,r)*g0*h0 + (7*df(g2,t)*h0 + 107*
df(h0,t)*g2)*df(g0,t)*g0**2*r - (2*f2*g0 - 81*g2)*df(g0,t)**2*g0*h0*r + 3*df(g0,r,2)*g0*g2*h0**2*r -
9*df(g0,r)**2*g2*h0**2*r - 207*df(g0,t,2)*g0**2*g2*h0*r - 2*df(g2,r,2)*g0**2*h0**2*r - 10*df(g2,r)*
df(h0,r)*g0**2*h0*r - 216*df(h0,r,2)*g0**2*g2*h0*r + 98*df(h0,r)**2*g0**2*g2*r)*(sin(theta)**2*dphi
**2 + dtheta**2)*g0*epsilon*r**2$

eq34 := - sqrt(sin(theta)**2*dphi**2 + dtheta**2)*df(g0,t)**2*g0**3*g1*h0*epsilon*dr*r$

eq35 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*((9*f1*g0 + 32*g1)*df(h0,r)*r - (9*f1*g0 - 821*g1
)*h0 + 11*df(g1,r)*h0*r)*df(g0,t)*g0*h0 + ((9*f1*g0 + 14*g1)*df(g0,r,t)*h0**2*r + 18*df(f1,r)*df(g0,
t)*g0*h0**2*r - 12*df(g1,r,t)*g0*h0**2*r - 6*df(g1,r)*df(h0,t)*g0*h0*r - 18*df(h0,r,t)*g0*g1*h0*r +
6*df(h0,r)*df(h0,t)*g0*g1*r + 12*df(h0,t)*g0*g1*h0)*g0 - 3*(3*(f1*g0 + 4*g1)*df(g0,t)*h0 - 4*df(g1,t
)*g0*h0 - 2*df(h0,t)*g0*g1)*df(g0,r)*h0*r - 24*(df(h0,r)*r - h0)*df(g1,t)*g0**2*h0)*(sin(theta)**2*
dphi**2 + dtheta**2)*g0*epsilon*r**2$

eq36 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*((5*df(g1,r)*h0*r + 111*df(h0,r)*g1*r + 428*g1*h0)*
df(g0,r)*g0*h0 + (df(g1,t)*h0 + 106*df(h0,t)*g1)*df(g0,t)*g0**2*r - 4*(df(h0,r)*r + h0)*df(g1,r)*g0
**2*h0 + 2*df(g0,r,2)*g0*g1*h0**2*r - 5*df(g0,r)**2*g1*h0**2*r - 211*df(g0,t,2)*g0**2*g1*h0*r + 102*
df(g0,t)**2*g0*g1*h0*r - 2*df(g1,r,2)*g0**2*h0**2*r - 214*df(h0,r,2)*g0**2*g1*h0*r + 106*df(h0,r)**2
*g0**2*g1*r - 4*df(h0,r)*g0**2*g1*h0)*ct*g0*h0*epsilon*dr$

eq37 := - 2*(((428*f2*g0 + 851*g2)*df(h0,r,2)*g0**2*h0*r + (422*f2*g0 + 833*g2)*df(g0,t,2)*g0**2*h0
```

```
*r - 2*(103*f2*g0 + 191*g2)*df(g0,t)**2*g0*h0*r - 4*(53*f2*g0 + 102*g2)*df(h0,r)**2*g0**2*r + (4*f2*
g0 + 9*g2)*df(g0,r)**2*h0**2*r + 4*(2*f2*g0 + 103*g2)*df(h0,r)*g0**2*h0 - (2*f2*g0 + 3*g2)*df(g0,r,2
)*g0*h0**2*r)*h0 - ((218*f2*g0 + 445*g2)*df(h0,r)*r + 12*(71*f2*g0 + 105*g2)*h0 + 7*df(g2,r)*h0*r)*
df(g0,r)*g0*h0**2 - ((212*f2*g0 + 429*g2)*df(h0,t) + 25*df(g2,t)*h0)*df(g0,t)*g0**2*h0*r + (4*df(f2,
r,2)*g0*h0**3 - 2*df(f2,t)*df(g0,t)*g0*h0**2 + 2*df(g2,r,2)*h0**3 + 6*df(g2,t,2)*g0*h0**2 + 3*df(g2,
t)*df(h0,t)*g0*h0 + 3*df(h0,t,2)*g0*g2*h0 - 3*df(h0,t)**2*g0*g2)*g0**2*r - 2*(3*df(g0,r)*h0*r - 4*df
(h0,r)*g0*r - 4*g0*h0)*df(f2,r)*g0**2*h0**2 + (13*df(h0,r)*r - 8*h0)*df(g2,r)*g0**2*h0**2)*(sin(theta
)**2*dphi**2 + dtheta**2)*epsilon*r**2$

eq38 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(((2*g0*h1 + 7*g1*h0)*df(h0,r)*r + 10*df(g1,r)*h0**2
*r + 106*df(h0,t)*g0*h2*r - 2*df(h1,r)*g0*h0*r - df(h2,t)*g0*h0*r - 2*g0*h0*h1 + 856*g1*h0**2)*df(g0
,t)*g0 + ((g0*h1 - 16*g1*h0)*df(g0,t)*r + 4*df(g1,t)*g0*h0*r + 106*df(h0,r)*g0*h2*r + 2*df(h0,t)*g0*
g1*r + 3*df(h2,r)*g0*h0*r + 426*g0*h0*h2)*df(g0,r)*h0 - 2*(df(h0,r)*r + 2*h0)*df(g1,t)*g0**2*h0 - (g0
*h1 - 6*g1*h0)*df(g0,r,t)*g0*h0*r + df(g0,r,2)*g0*h0**2*h2*r - 2*df(g0,r)**2*h0**2*h2*r - 212*df(g0,
t,2)*g0**2*h0*h2*r + 107*df(g0,t)**2*g0*h0*h2*r - 4*df(g1,r,t)*g0**2*h0**2*r - 2*df(g1,r)*df(h0,t)*
g0**2*h0*r - 3*df(h0,r,t)*g0**2*g1*h0*r - 212*df(h0,r,2)*g0**2*h0*h2*r + 106*df(h0,r)**2*g0**2*h2*r
+ 2*df(h0,r)*df(h0,t)*g0**2*g1*r - 2*df(h0,t)*g0**2*g1*h0 - 2*df(h2,r,2)*g0**2*h0**2*r - 4*df(h2,r)*
g0**2*h0**2)*ct*g0*h0*epsilon$

eq39 := ((9*(2*df(g2,t)*h0 + df(h0,t)*g2)*g0 - 2*(4*f2*g0 + 39*g2)*df(g0,t)*h0)*df(g0,r)*h0*r + ((8*
f2*g0 + 93*g2)*df(h0,r)*r + 36*df(g2,r)*h0*r + 8*f2*g0*h0 + 1684*g2*h0)*df(g0,t)*g0*h0 + 2*(4*df(f2,
r)*df(g0,t)*h0**2 - 6*df(g2,r,t)*h0**2 - 3*df(g2,r)*df(h0,t)*h0 - 12*df(g2,t)*df(h0,r)*h0 - 9*df(h0,
r,t)*g2*h0 + 3*df(h0,r)*df(h0,t)*g2)*g0**2*r + 4*(f2*g0 + 6*g2)*df(g0,r,t)*g0*h0**2*r)*(sin(theta)**
2*dphi**2 + dtheta**2)*g0*epsilon*r**2$

eq40 := sqrt(sin(theta)**2*dphi**2 + dtheta**2)*(2*(df(g1,r)*h0*r + df(h0,r)*g1*r + g1*h0)*df(g0,t)*
g0 + df(g0,r,t)*g0*g1*h0*r - 3*df(g0,r)*df(g0,t)*g1*h0*r)*g0**2*h0*epsilon*dr$
```