Simon-Mars tensor in Curzon-Chazy spacetime

This worksheet demonstrates a few capabilities of SageManifolds (version 1.0, as included in SageMath 7.5) in computations regarding the Curzon-Chazy spacetime. It implements the computation of the Simon-Mars tensor of Curzon-Chazy spacetime used in the article arXiv:1412.6542.

Click here to download the worksheet file (ipynb format). To run it, you must start SageMath with the Jupyter notebook, via the command `sage -n jupyter`

NB: a version of SageMath at least equal to 7.5 is required to run this worksheet:

In [1]:
```
version()
```

Out[1]: 'SageMath version 7.5.1, Release Date: 2017-01-15'

First we set up the notebook to display mathematical objects using LaTeX rendering:

In [2]:
```
%display latex
```

Spacetime manifold

We declare the Curzon-Chazy spacetime as a 4-dimensional manifold:

In [3]:
```
M = Manifold(4, 'M', latex_name=r'\mathcal{M}')
print(M)
```

```
4-dimensional differentiable manifold M
```

We introduce the coordinates (t, r, y, ϕ) with y related to the standard **Weyl-Papapetrou coordinates** (t, r, θ, ϕ) by $y = \cos\theta$:

In [4]:
```
X.<t,r,y,ph> = M.chart(r't r:(0,+oo) y:(-1,1) ph:(0,2*pi):\phi')
print(X) ; X
```

```
Chart (M, (t, r, y, ph))
```

Out[4]: $(\mathcal{M}, (t, r, y, \phi))$

Metric tensor

We declare the only parameter of the Curzon-Chazy spacetime, which is the mass m as a symbolic variable:

In [5]:
```
var('m')
```

Out[5]: m

Without any loss of generality, we set m to some specific value (this amounts simply to fixing some length scale):

In [6]:
```
m = 12
```

Let us introduce the spacetime metric g and set its components in the coordinate frame associated with Weyl-Papapetrou coordinates:

In [7]:
```
g = M.lorentzian_metric('g')
g[0,0] = - exp(-2*m/r)
g[1,1] = exp(2*m/r-m^2*(1-y^2)/r^2)
g[2,2] = exp(2*m/r-m^2*(1-y^2)/r^2)*r^2/(1-y^2)
g[3,3] = exp(2*m/r)*r^2*(1-y^2)
```

In [8]:
```
g[:]
```

Out[8]:

$$\begin{pmatrix} -e^{\left(-\frac{24}{r}\right)} & 0 & 0 & 0 \\ 0 & e^{\left(\frac{144\,(y^2-1)}{r^2}+\frac{24}{r}\right)} & 0 & 0 \\ 0 & 0 & -\frac{r^2 e^{\left(\frac{144\,(y^2-1)}{r^2}+\frac{24}{r}\right)}}{y^2-1} & 0 \\ 0 & 0 & 0 & -\left(y^2-1\right) r^2 e^{\frac{24}{r}} \end{pmatrix}$$

The Levi-Civita connection ∇ associated with g:

In [9]:
```
nab = g.connection() ; print(nab)
```

```
Levi-Civita connection nabla_g associated with the Lorentzian metric g
on the 4-dimensional differentiable manifold M
```

As a check, we verify that the covariant derivative of g with respect to ∇ vanishes identically:

In [10]:
```
nab(g).display()
```

Out[10]: $\nabla_g g = 0$

Killing vector

The default vector frame on the spacetime manifold is the coordinate basis associated with Weyl-Papapetrou coordinates:

In [11]:
```
M.default_frame() is X.frame()
```

Out[11]: True

In [12]:
```
X.frame()
```

Out[12]: $\left(\mathcal{M}, \left(\frac{\partial}{\partial t}, \frac{\partial}{\partial r}, \frac{\partial}{\partial y}, \frac{\partial}{\partial \phi}\right)\right)$

Let us consider the first vector field of this frame:

In [13]:
```
xi = X.frame()[0] ; xi
```

Out[13]: $\frac{\partial}{\partial t}$

In [14]:
```
print(xi)
```

```
Vector field d/dt on the 4-dimensional differentiable manifold M
```

The 1-form associated to it by metric duality is

In [15]:
```
xi_form = xi.down(g)
xi_form.set_name('xi_form', r'\underline{\xi}')
print(xi_form) ; xi_form.display()
```

```
1-form xi_form on the 4-dimensional differentiable manifold M
```

Out[15]:
$$\underline{\xi} = -e^{\left(-\frac{24}{r}\right)} \mathrm{d}t$$

Its covariant derivative is

In [16]:
```
nab_xi = nab(xi_form)
print(nab_xi) ; nab_xi.display()
```

```
Tensor field nabla_g(xi_form) of type (0,2) on the 4-dimensional differ
entiable manifold M
```

Out[16]:
$$\nabla_g \underline{\xi} = -\frac{12\, e^{\left(-\frac{24}{r}\right)}}{r^2} \mathrm{d}t \otimes \mathrm{d}r + \frac{12\, e^{\left(-\frac{24}{r}\right)}}{r^2} \mathrm{d}r \otimes \mathrm{d}t$$

Let us check that the Killing equation is satisfied:

In [17]:
```
nab_xi.symmetrize().display()
```

Out[17]: 0

Equivalently, we check that the Lie derivative of the metric along ξ vanishes:

In [18]:
```
g.lie_der(xi).display()
```

Out[18]: 0

Thank to Killing equation, $\nabla_g \underline{\xi}$ is antisymmetric. We may therefore define a 2-form by $F := -\nabla_g \xi$. Here we enforce the antisymmetry by calling the function `antisymmetrize()` on `nab_xi`:

In [19]:
```
F = - nab_xi.antisymmetrize()
F.set_name('F')
print(F)
F.display()
```

```
2-form F on the 4-dimensional differentiable manifold M
```

Out[19]:
$$F = \frac{12\, e^{\left(-\frac{24}{r}\right)}}{r^2} \mathrm{d}t \wedge \mathrm{d}r$$

We check that

In [20]:
```
F == - nab_xi
```

Out[20]: True

The squared norm of the Killing vector is

In [21]:
```
lamb = - g(xi,xi)
lamb.set_name('lambda', r'\lambda')
print(lamb)
lamb.display()
```

```
Scalar field lambda on the 4-dimensional differentiable manifold M
```

Out[21]:
$$\begin{array}{llcl} \lambda: & \mathcal{M} & \longrightarrow & \mathbb{R} \\ & (t, r, y, \phi) & \longmapsto & e^{\left(-\frac{24}{r}\right)} \end{array}$$

Instead of invoking $g(\xi, \xi)$, we could have evaluated λ by means of the 1-form $\underline{\xi}$ acting on the vector field ξ:

In [22]:
```
lamb == - xi_form(xi)
```

Out[22]: True

or we could have used index notation in the form $\lambda = -\xi_a \xi^a$:

In [23]:
```
lamb == - ( xi_form['_a']*xi['^a'] )
```

Out[23]: True

Curvature

The Riemann curvature tensor associated with g is

In [24]:
```
Riem = g.riemann()
print(Riem)
```

```
Tensor field Riem(g) of type (1,3) on the 4-dimensional differentiable
manifold M
```

The component $R^0{}_{101} = R^t{}_{rtr}$ is

In [25]:
```
Riem[0,1,0,1]
```

Out[25]:
$$\frac{24 \left(r^2 - 72 y^2 - 12 r + 72\right)}{r^5}$$

while the component $R^2{}_{323} = R^y{}_{\phi y \phi}$ is

In [26]:
```
Riem[2,3,2,3]
```

Out[26]:
$$\frac{24 \left(72 y^4 e^{\left(\frac{144}{r^2}\right)} - \left(r^2 - 12 r + 144\right) y^2 e^{\left(\frac{144}{r^2}\right)} + \left(r^2 - 12 r + 72\right) e^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144 y^2}{r^2}\right)}}{r^3}$$

All the non-vanishing components of the Riemann tensor, taking into account the antisymmetry on the last two indices:

In [27]:
```
Riem.display_comp(only_nonredundant=True)
```

Out[27]:
```
\newcommand{\Bold}[1]{\mathbf{#1}}\begin{array}{lcl} \mathrm{Riem}\left(g\right)_{ \phantom{\, t} \, r \, t \, r }^{ \, t \
```

The Ricci tensor:

In [28]:
```
Ric = g.ricci()
print(Ric)
```

```
Field of symmetric bilinear forms Ric(g) on the 4-dimensional different
iable manifold M
```

Let us check that the Curzon-Chazy metric is a solution of the **vacuum Einstein equation**:

In [29]:
```
Ric.display()
```

Out[29]: $\mathrm{Ric}\left(g\right) = 0$

The Weyl conformal curvature tensor is

In [30]:
```
C = g.weyl()
print(C)
```

```
Tensor field C(g) of type (1,3) on the 4-dimensional differentiable man
ifold M
```

Let us exhibit two of its components ${C^0}_{123}$ and ${C^0}_{101}$:

In [31]:
```
C[0,1,2,3]
```

Out[31]: 0

In [32]:
```
C[0,1,0,1]
```

Out[32]: $$\frac{24 \left(r^{2} - 72 \, y^{2} - 12 \, r + 72\right)}{r^{5}}$$

To form the Mars-Simon tensor, we need the fully covariant (type-(0,4) tensor) form of the Weyl tensor (i.e. $C_{\alpha\beta\mu\nu} = g_{\alpha\sigma} {C^\sigma}_{\beta\mu\nu}$); we get it by lowering the first index with the metric:

In [33]:
```
Cd = C.down(g)
print(Cd)
```

```
Tensor field of type (0,4) on the 4-dimensional differentiable manifold
 M
```

The (monoterm) symmetries of this tensor are those inherited from the Weyl tensor, i.e. the antisymmetry on the last two indices (position 2 and 3, the first index being at position 0):

In [34]:
```
Cd.symmetries()
```

```
no symmetry;  antisymmetry: (2, 3)
```

Actually, Cd is also antisymmetric with respect to the first two indices (positions 0 and 1), as we can check:

In [35]:
```
Cd == Cd.antisymmetrize(0,1)
```

Out[35]: True

To take this symmetry into account explicitely, we set

In [36]:
```
Cd = Cd.antisymmetrize(0,1)
```

Hence we have now

In [37]:
```
Cd.symmetries()
```

```
no symmetry; antisymmetries: [(0, 1), (2, 3)]
```

Simon-Mars tensor

The Simon-Mars tensor with respect to the Killing vector ξ is a rank-3 tensor introduced by Marc Mars in 1999 (Class. Quantum Grav. **16**, 2507). It has the remarkable property to vanish identically if, and only if, the spacetime $(\mathcal{M}, g)$ is locally isometric to a Kerr spacetime.

Let us evaluate the Simon-Mars tensor by following the formulas given in Mars' article. The starting point is the self-dual complex 2-form associated with the Killing 2-form F, i.e. the object $\mathcal{F} := F + i\,{}^*F$, where ${}^*F$ is the Hodge dual of F:

In [38]:
```
FF = F + I * F.hodge_dual(g)
FF.set_name('FF', r'\mathcal{F}')
print(FF) ; FF.display()
```

```
2-form FF on the 4-dimensional differentiable manifold M
```

Out[38]:

$$\mathcal{F} = \frac{12\, e^{\left(-\frac{24}{r}\right)}}{r^2} \mathrm{d}t \wedge \mathrm{d}r - 12i \mathrm{d}y \wedge \mathrm{d}\phi$$

Let us check that $\mathcal{F}$ is self-dual, i.e. that it obeys ${}^*\mathcal{F} = -i\mathcal{F}$:

In [39]:
```
FF.hodge_dual(g) == - I * FF
```

Out[39]: True

Let us form the right self-dual of the Weyl tensor as follows

$$\mathcal{C}_{\alpha\beta\mu\nu} = C_{\alpha\beta\mu\nu} + \frac{i}{2}\epsilon^{\rho\sigma}{}_{\mu\nu}\, C_{\alpha\beta\rho\sigma}$$

where $\epsilon^{\rho\sigma}{}_{\mu\nu}$ is associated to the Levi-Civita tensor $\epsilon_{\rho\sigma\mu\nu}$ and is obtained by

In [40]:
```
eps = g.volume_form(2)  # 2 = the first 2 indices are contravariant
print(eps)
eps.symmetries()
```

```
Tensor field of type (2,2) on the 4-dimensional differentiable manifold
 M
no symmetry; antisymmetries: [(0, 1), (2, 3)]
```

The right self-dual Weyl tensor is then:

In [41]:
```
CC = Cd + I/2*( eps['^rs_..']*Cd['_..rs'] )
CC.set_name('CC', r'\mathcal{C}') ; print(CC)
```

```
Tensor field CC of type (0,4) on the 4-dimensional differentiable manif
old M
```

In [42]:
```
CC.symmetries()
```

```
no symmetry; antisymmetries: [(0, 1), (2, 3)]
```

In [43]:
```
CC[0,1,2,3]
```

Out[43]: $$\frac{24i\,r^2 - 1728i\,y^2 - 288i\,r + 1728i}{r^3}$$

The Ernst 1-form $\sigma_\alpha = 2\mathcal{F}_{\mu\alpha}\,\xi^\mu$ (0 = contraction on the first index of $\mathcal{F}$):

In [44]:
```
sigma = 2*FF.contract(0, xi)
```

Instead of invoking the function `contract()`, we could have used the index notation to denote the contraction:

In [45]:
```
sigma == 2*( FF['_ma']*xi['^m'] )
```

Out[45]: True

In [46]:
```
sigma.set_name('sigma', r'\sigma')
print(sigma) ; sigma.display()
```

```
1-form sigma on the 4-dimensional differentiable manifold M
```

Out[46]: $$\sigma = \frac{24\,e^{\left(-\frac{24}{r}\right)}}{r^2}\mathrm{d}r$$

The symmetric bilinear form $\gamma = \lambda\,g + \underline{\xi}\otimes\underline{\xi}$:

In [47]:
```
gamma = lamb*g + xi_form * xi_form
gamma.set_name('gamma', r'\gamma')
print(gamma) ; gamma.display()
```

```
Field of symmetric bilinear forms gamma on the 4-dimensional differenti
able manifold M
```

Out[47]: $$\gamma = e^{\left(\frac{144\,y^2}{r^2}-\frac{144}{r^2}\right)}\mathrm{d}r\otimes\mathrm{d}r + \left(-\frac{r^2 e^{\left(\frac{144\,y^2}{r^2}\right)}}{y^2 e^{\left(\frac{144}{r^2}\right)} - e^{\left(\frac{144}{r^2}\right)}}\right)\mathrm{d}y\otimes\mathrm{d}y + \left(-r^2y^2 + r^2\right)\mathrm{d}\phi\otimes\mathrm{d}\phi$$

Final computation leading to the Simon-Mars tensor:

The first part of the Simon-Mars tensor is

$$S^{(1)}_{\alpha\beta\gamma} = 4C_{\mu\alpha\nu\beta}\,\xi^\mu\,\xi^\nu\,\sigma_\gamma$$

In [48]:
```
S1 = 4*( CC.contract(0,xi).contract(1,xi) ) * sigma
print(S1)
```

```
Tensor field of type (0,3) on the 4-dimensional differentiable manifold
 M
```

The second part is the tensor

$$S^{(2)}_{\alpha\beta\gamma} = -\gamma_{\alpha\beta}\, \mathcal{C}_{\rho\gamma\mu\nu}\, \xi^{\rho}\, \mathcal{F}^{\mu\nu}$$

which we compute by using the index notation to denote the contractions:

In [49]:
```
FFuu = FF.up(g)
xiCC = CC['_.r..']*xi['^r']
S2 = gamma * ( xiCC['_.mn']*FFuu['^mn'] )
print(S2)
```

```
Tensor field of type (0,3) on the 4-dimensional differentiable manifold
 M
```

In [50]:
```
S2.symmetries()
```

```
symmetry: (0, 1); no antisymmetry
```

The Mars-Simon tensor with respect to ξ is obtained by antisymmetrizing $S^{(1)}$ and $S^{(2)}$ on their last two indices and adding them:

$$S_{\alpha\beta\gamma} = S^{(1)}_{\alpha[\beta\gamma]} + S^{(2)}_{\alpha[\beta\gamma]}$$

We use the index notation for the antisymmetrization:

In [51]:
```
S1A = S1['_a[bc]']
S2A = S2['_a[bc]']
```

An equivalent writing would have been (the last two indices being in position 1 and 2):

In [52]:
```
# S1A = S1.antisymmetrize(1,2)
# S2A = S2.antisymmetrize(1,2)
```

The Simon-Mars tensor is

In [53]:
```
S = S1A + S2A
S.set_name('S') ; print(S)
S.symmetries()
```

```
Tensor field S of type (0,3) on the 4-dimensional differentiable manifo
ld M
no symmetry; antisymmetry: (1, 2)
```

In [54]:
```
S.display()
```

Out[54]:

$$S = \frac{41472\, ye^{\left(-\frac{48}{r}\right)}}{r^6} \mathrm{d}r \otimes \mathrm{d}r \otimes \mathrm{d}y - \frac{41472\, ye^{\left(-\frac{48}{r}\right)}}{r^6} \mathrm{d}r \otimes \mathrm{d}y \otimes \mathrm{d}r - \frac{41472\, e^{\left(-\frac{48}{r}\right)}}{r^5} \mathrm{d}y \otimes \mathrm{d}r \otimes \mathrm{d}y + \frac{41472\, e^{\left(-\frac{48}{r}\right)}}{r^5} \mathrm{d}y \otimes \mathrm{d}y \otimes \mathrm{d}r + \frac{41472 \left(y^4 e^{\left(\frac{144}{r^2}\right)} - 2\, y^2 e^{\left(\frac{144}{r^2}\right)} + e^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^5} \mathrm{d}\phi \otimes \mathrm{d}r \otimes \mathrm{d}\phi - \frac{41472 \left(y^3 e^{\left(\frac{144}{r^2}\right)} - ye^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^4} \mathrm{d}\phi \otimes \mathrm{d}y \otimes \mathrm{d}\phi - \frac{41472 \left(y^4 e^{\left(\frac{144}{r^2}\right)} - 2\, y^2 e^{\left(\frac{144}{r^2}\right)} + e^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^5} \mathrm{d}\phi \otimes \mathrm{d}\phi \otimes \mathrm{d}r + \frac{41472 \left(y^3 e^{\left(\frac{144}{r^2}\right)} - ye^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^4} \mathrm{d}\phi \otimes \mathrm{d}\phi \otimes \mathrm{d}y$$

In [55]:
```
S.display_comp()
```

Out[55]:

$$S_{r\,r\,y} = \frac{41472\, ye^{\left(-\frac{48}{r}\right)}}{r^6}$$

$$S_{r\,y\,r} = -\frac{41472\, ye^{\left(-\frac{48}{r}\right)}}{r^6}$$

$$S_{y\,r\,y} = -\frac{41472\, e^{\left(-\frac{48}{r}\right)}}{r^5}$$

$$S_{y\,y\,r} = \frac{41472\, e^{\left(-\frac{48}{r}\right)}}{r^5}$$

$$S_{\phi\,r\,\phi} = \frac{41472 \left(y^4 e^{\left(\frac{144}{r^2}\right)} - 2\, y^2 e^{\left(\frac{144}{r^2}\right)} + e^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^5}$$

$$S_{\phi\,y\,\phi} = -\frac{41472 \left(y^3 e^{\left(\frac{144}{r^2}\right)} - ye^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^4}$$

$$S_{\phi\,\phi\,r} = -\frac{41472 \left(y^4 e^{\left(\frac{144}{r^2}\right)} - 2\, y^2 e^{\left(\frac{144}{r^2}\right)} + e^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^5}$$

$$S_{\phi\,\phi\,y} = \frac{41472 \left(y^3 e^{\left(\frac{144}{r^2}\right)} - ye^{\left(\frac{144}{r^2}\right)}\right) e^{\left(-\frac{144\, y^2}{r^2} - \frac{48}{r}\right)}}{r^4}$$

Hence the Simon-Mars tensor is not zero: the Curzon-Chazy spacetime is not locally isomorphic to the Kerr spacetime.

Computation of the Simon-Mars scalars

First we form the "square" of the Simon-Mars tensor:

In [56]:
```
Su = S.up(g)
print(Su)
```

```
Tensor field of type (3,0) on the 4-dimensional differentiable manifold
 M
```

In [57]:
```
SS = S['_ijk']*Su['^ijk']
print(SS)
```

```
Scalar field on the 4-dimensional differentiable manifold M
```

In [58]:
```
SS.display()
```

Out[58]:

$$\begin{array}{llll} \mathcal{M} & \longrightarrow & \mathbb{R} \\ (t, r, y, \phi) & \longmapsto & -\frac{6879707136\left(y^2 e^{\left(\frac{432}{r^2}\right)} - e^{\left(\frac{432}{r^2}\right)}\right) e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r}\right)}}{r^{14}} \end{array}$$

In [59]:
```
SSE=SS.expr()
```

Then we take the real and imaginary part of this compex scalar field. Because this spacetime is spherically symmetric, we expect that the imaginary part vanishes.

In [60]:
```
SS1 = real(SSE) ; SS1
```

Out[60]:

$$-\frac{6879707136\,y^2 e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}} + \frac{6879707136\,e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}}$$

In [61]:
```
SS2 = imag(SSE) ; SS2
```

Out[61]: 0

Furthermore we scale those scalars by the ADM mass of the Curzon-Chazy spacetime, which corresponds to m:

In [62]:
```
SS1ad = m^6*SS1 ; SS1ad
```

Out[62]:

$$-\frac{20542695432781824\,y^2 e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}} + \frac{20542695432781824\,e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}}$$

And we take the log of this quantity

In [63]:
```
lSS1ad = log(SS1ad,10) ; lSS1ad
```

Out[63]:

$$\frac{\log\left(-\frac{20542695432781824\,y^2 e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}} + \frac{20542695432781824\,e^{\left(-\frac{432\,y^2}{r^2} - \frac{168}{r} + \frac{432}{r^2}\right)}}{r^{14}}\right)}{\log(10)}$$

Then we plot the value of this quantity as a function of $\rho = x = r\sqrt{1 - y^2}$ and $z = ry$, thereby producing Figure 10 of arXiv:1412.6542:

In [64]:
```
var('x z')
lSS1xzad = lSS1ad.subs(r=sqrt(x^2+z^2),
                       y = z/sqrt(x^2+z^2)).simplify_full()
lSS1xzad
```

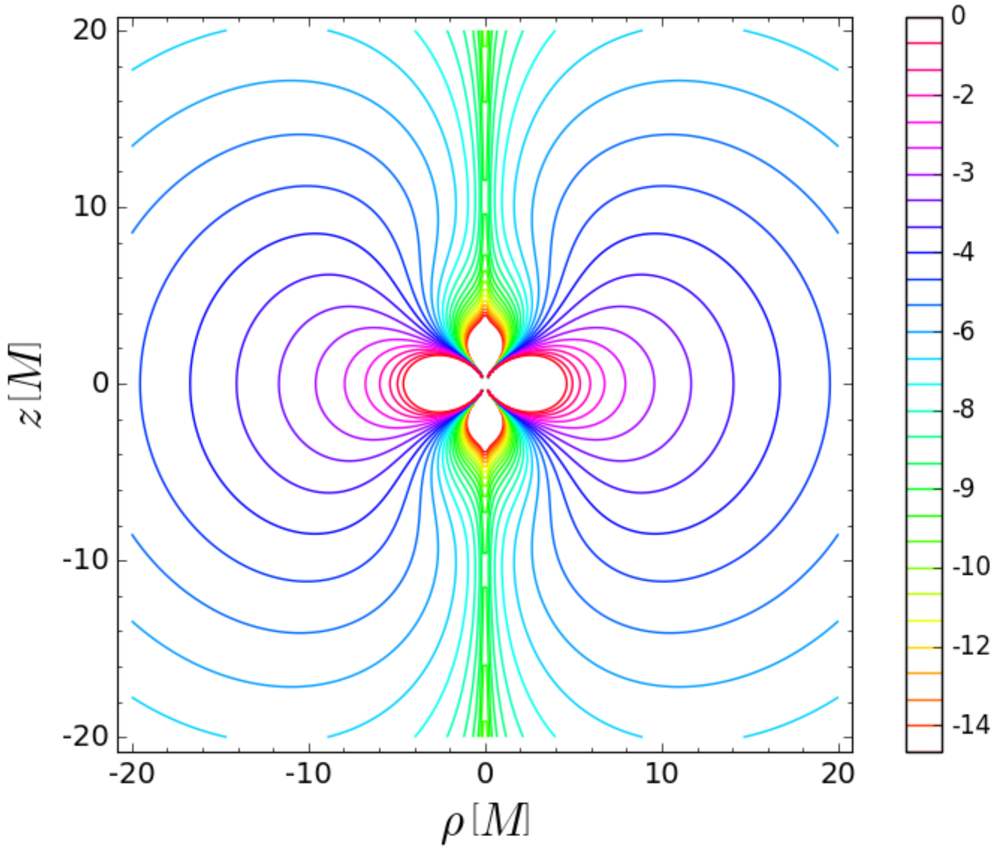
Out[64]:

$$\frac{\log\left(\frac{20542695432781824\, x^2 e^{\left(-\frac{432\, z^2}{x^4+2\, x^2z^2+z^4} - \frac{168}{\sqrt{x^2+z^2}} + \frac{432}{x^2+z^2}\right)}}{x^{16}+8\, x^{14}z^2+28\, x^{12}z^4+56\, x^{10}z^6+70\, x^8z^8+56\, x^6z^{10}+28\, x^4z^{12}+8\, x^2z^{14}+z^{16}}\right)}{\log(10)}$$

In [65]:
```
S1CC1 = contour_plot(lSS1xzad, (x,-20,20), (z,-20,20), plot_points=200,

                     fill=False, cmap='hsv', linewidths=1,
                     contours=(-14,-13.5,-13,-12.5,-12,-11.5,-11,
                               -10.5,-10,-9.5,-9,-8.5,-8,-7.5,-7,
                               -6.5,-6,-5.5,-5,-4.5,-4,-3.5,-3,-2.5,
                               -2,-1.5,-1,-0.5,0),
                     colorbar=True, colorbar_spacing='uniform',
                     colorbar_format='%1.f',
                     axes_labels=(r"$\rho\,\left[M\right]$",
                                  r"$z\,\left[M\right]$"),
                     fontsize=14)
S1CC1
```

Out[65]:



We also a viewer for 3D plots (use 'threejs' or 'jmol' for interactive 3D graphics):

In [66]:
```
viewer3D = 'tachyon' # must be 'threejs', 'jmol', 'tachyon' or None (de
fault)
```

In [67]:
```
plot3d(lSS1xzad, (x,0.12,20), (z,0.12,20), viewer=viewer3D,
       aspect_ratio=[1,1,0.05], plot_points=100,
       axes_labels=['rho', 'z', 'log(beta)'])
```

Out[67]: