Computational differential geometry with applications to gravitational physics

Éric Gourgoulhon

Laboratoire Univers et Théories (LUTH)
Observatoire de Paris, CNRS, Université PSL, Université de Paris
Meudon, France

https://luth.obspm.fr/~luthier/gourgoulhon

Yukawa Institute for Theoretical Physics

Kyoto, Japan
10 December 2020

To the memory of

Prof. Yoshiharu Eriguchi

deceased on 13 October 2020

Outline

1. Introduction
2. Differential geometry with SageMath
3. Some implementation details
4. Example 1: near-horizon geometry of the extremal Kerr black hole
5. Example 2: gravitational radiation from bodies orbiting Sgr A*
6. Example 3: images of black holes
7. Conclusions

Outline

Symbolic differential geometry and tensor calculus

Packages/modules for general purpose computer algebra systems

- xAct free package for Mathematica [J.-M. Martin-Garcia]
- Ricci free package for Mathematica [J.L. Lee]
- MathTensor package for Mathematica [S.M. Christensen & L. Parker]
- GRTensor III package for Maple [P. Musgrave, D. Pollney & K. Lake]
- DifferentialGeometry included in Maple [I.M. Anderson & E.S. Cheb-Terrab]
- Atlas 2 for Maple and Mathematica
- SageManifolds module included in SageMath

Standalone applications

- SHEEP, Classi, STensor, based on Lisp, developed in 1970's and 1980's (free) [R. d'Inverno, I. Frick, J. Åman, J. Skea, et al.]
- Cadabra (free) [K. Peeters]
- Redberry (free) [D.A. Bolotin & S.V. Poslavsky]

cf. the rather exhaustive list at http://www.xact.es/links.html

Tensor calculus software

Two kinds of **tensor computations**:

Abstract calculus (index manipulations)

- xAct/xTensor
- MathTensor
- Ricci
- Cadabra
- Redberry

Component calculus (explicit computations)

- xAct/xCoba
- Atlas 2
- DifferentialGeometry
- SageManifolds

Outline

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system

SageMath is free (GPL v2)

Freedom means

1. everybody can use it, by downloading the application from https://www.sagemath.org
2. everybody can examine the source code and improve it

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system

SageMath is free (GPL v2)

Freedom means

1. everybody can use it, by downloading the application from https://www.sagemath.org
2. everybody can examine the source code and improve it

SageMath is based on Python

- no need to learn any specific syntax to use it
- Python is a powerful *object oriented language*, with a neat syntax
- SageMath benefits from the Python ecosystem (e.g. Jupyter notebook, NumPy, Matplotlib)

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system

SageMath is free (GPL v2)

Freedom means

1. everybody can use it, by downloading the application from `https://www.sagemath.org`
2. everybody can examine the source code and improve it

SageMath is based on Python

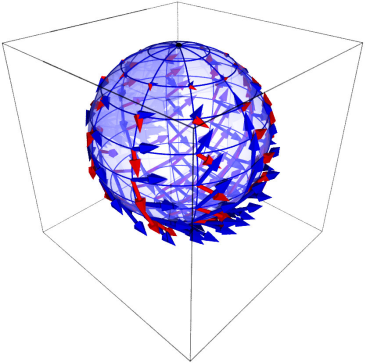
- no need to learn any specific syntax to use it
- Python is a powerful *object oriented language*, with a neat syntax
- SageMath benefits from the Python ecosystem (e.g. Jupyter notebook, NumPy, Matplotlib)

SageMath is developed by an enthusiastic community

- mostly composed of mathematicians
- welcoming newcomers

Differential geometry with SageMath

SageManifolds project: extends SageMath towards **differential geometry** and **tensor calculus**



Stereographic-coordinate frame on $\mathbb{S}^2$

- https://sagemanifolds.obspm.fr
- more than 110,000 lines of Python code
- fully included in SageMath (after review process)
- $\sim$ 25 contributors (developers and reviewers) cf. https://sagemanifolds.obspm.fr/authors.html
- dedicated mailing list
- help: https://ask.sagemath.org

Everybody is welcome to contribute
$\Longrightarrow$ visit https://sagemanifolds.obspm.fr/contrib.html

Current status

Already present (SageMath 9.2):

- differentiable manifolds: tangent spaces, vector frames, tensor fields, curves, pullback and pushforward operators, submanifolds
- vector bundles (tangent bundle, tensor bundles)
- standard tensor calculus (tensor product, contraction, symmetrization, etc.), even on non-parallelizable manifolds, and with all monoterm tensor symmetries taken into account
- Lie derivatives of tensor fields
- differential forms: exterior and interior products, exterior derivative, Hodge duality
- multivector fields: exterior and interior products, Schouten-Nijenhuis bracket
- affine connections (curvature, torsion)
- pseudo-Riemannian metrics
- computation of geodesics (numerical integration)

Current status

Already present *(cont'd)*:

- some plotting capabilities (charts, points, curves, vector fields)
- parallelization (on tensor components) of CPU demanding computations
- extrinsic geometry of pseudo-Riemannian submanifolds
- series expansions of tensor fields
- 2 symbolic backends: Pynac/Maxima (SageMath's default) and SymPy

Future prospects:

- more symbolic backends (Giac, FriCAS, ...)
- more graphical outputs
- symplectic forms, spinors, integrals on submanifolds, variational calculus, etc.
- connection with numerical relativity: use SageMath to explore numerically-generated spacetimes

Outline

SageMath approach to computer mathematics

SageMath relies on a `Parent` / `Element` scheme

Each object x on which some calculus is performed has a ***parent***, which is another SageMath object X representing the set to which x belongs.
The calculus rules on x are determined by the *algebraic structure* of X.
Conversion rules prior to an operation are defined at the level of the parents

SageMath approach to computer mathematics

SageMath relies on a `Parent` / `Element` scheme

Each object x on which some calculus is performed has a ***parent***, which is another SageMath object X representing the set to which x belongs.
The calculus rules on x are determined by the *algebraic structure* of X.
Conversion rules prior to an operation are defined at the level of the parents

Example: $x + y$ with x and y having different parents

```
sage: x = 4 ; x.parent()
Integer Ring
sage: y = 4/3 ; y.parent()
Rational Field
sage: s = x + y ; s.parent()
Rational Field
sage: y.parent().has_coerce_map_from(x.parent())
True
```

SageMath approach to computer mathematics

SageMath relies on a Parent / Element scheme

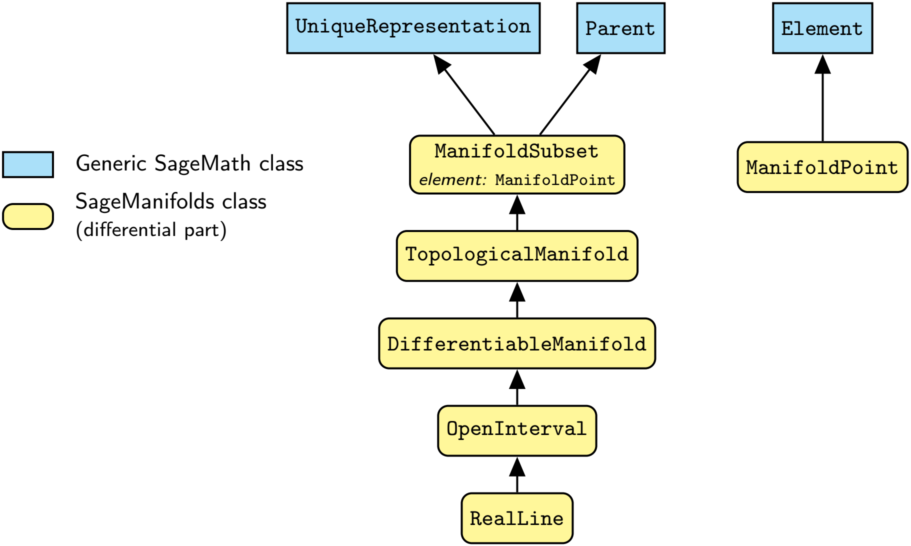
Each object x on which some calculus is performed has a ***parent***, which is another SageMath object X representing the set to which x belongs.
The calculus rules on x are determined by the *algebraic structure* of X.
Conversion rules prior to an operation are defined at the level of the parents

Example: $x + y$ with x and y having different parents

```
sage: x = 4 ; x.parent()
Integer Ring
sage: y = 4/3 ; y.parent()
Rational Field
sage: s = x + y ; s.parent()
Rational Field
sage: y.parent().has_coerce_map_from(x.parent())
True
```

This approach is similar to that of Magma and is different from that of Mathematica, in which everything is a tree of symbols

Implementing manifolds and their subsets



Implementing coordinate charts

Given a (topological) manifold M of dimension $n \geq 1$, a ***coordinate chart*** is a homeomorphism $\varphi : U \to V$, where U is an open subset of M and V is an open subset of $\mathbb{R}^n$.

Implementing coordinate charts

Given a (topological) manifold M of dimension $n \geq 1$, a ***coordinate chart*** is a homeomorphism $\varphi : U \to V$, where U is an open subset of M and V is an open subset of $\mathbb{R}^n$.

In general, more than one chart is required to cover the manifold:

Examples

- at least 2 charts are necessary to cover the n-dimensional sphere $\mathbb{S}^n$ $(n \geq 1)$ and the torus $\mathbb{T}^2$
- at least 3 charts are necessary to cover the real projective plane $\mathbb{RP}^2$

Implementing coordinate charts

Given a (topological) manifold M of dimension $n \geq 1$, a ***coordinate chart*** is a homeomorphism $\varphi : U \to V$, where U is an open subset of M and V is an open subset of $\mathbb{R}^n$.

In general, more than one chart is required to cover the manifold:

Examples

- at least 2 charts are necessary to cover the n-dimensional sphere $\mathbb{S}^n$ $(n \geq 1)$ and the torus $\mathbb{T}^2$
- at least 3 charts are necessary to cover the real projective plane $\mathbb{RP}^2$

In SageMath, an arbitrary number of charts can be introduced

To fully specify the manifold, one shall also provide the *transition maps* on overlapping chart domains (SageMath class `CoordChange`)

Implementing scalar fields

A **scalar field** on manifold M is a smooth map

$$\begin{array}{llll} f: & M & \longrightarrow & \mathbb{R} \\ & p & \longmapsto & f(p) \end{array}$$

Implementing scalar fields

A **scalar field** on manifold M is a smooth map

$$\begin{array}{rccc} f: & M & \longrightarrow & \mathbb{R} \\ & p & \longmapsto & f(p) \end{array}$$

A scalar field maps *points*, not *coordinates*, to real numbers
$\Longrightarrow$ an object f in the `ScalarField` class has different **coordinate representations** in different charts defined on M.

Implementing scalar fields

A **scalar field** on manifold M is a smooth map

$$f: \begin{array}{ccc} M & \longrightarrow & \mathbb{R} \\ p & \longmapsto & f(p) \end{array}$$

A scalar field maps *points*, not *coordinates*, to real numbers
$\Longrightarrow$ an object f in the `ScalarField` class has different **coordinate representations** in different charts defined on M.

The various coordinate representations F, $\hat{F}$, ... of f are stored as a *Python dictionary* whose keys are the charts C, $\hat{C}$, ...:

$$f.\texttt{\_express} = \left\{ C : F,\ \hat{C} : \hat{F}, \ldots \right\}$$

with $\underbrace{f(\,p\,)}_{\text{point}} = F(\underbrace{x^1,\ldots,x^n}_{\substack{\text{coord. of } p \\ \text{in chart } C}}) = \hat{F}(\underbrace{\hat{x}^1,\ldots,\hat{x}^n}_{\substack{\text{coord. of } p \\ \text{in chart } \hat{C}}}) = \ldots$

The scalar field algebra

The parent of the scalar field $f : M \to \mathbb{R}$ is the set $C^\infty(M)$ of scalar fields defined on the manifold M.

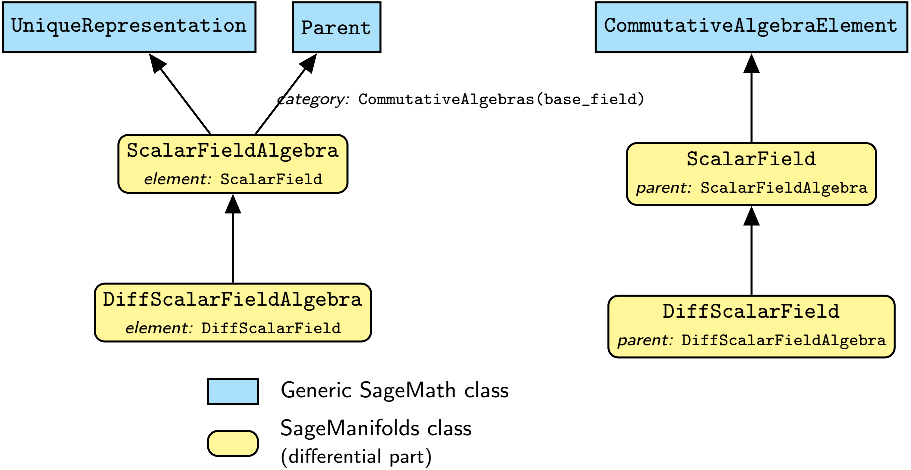
$C^\infty(M)$ has naturally the structure of a **commutative algebra over** $\mathbb{R}$:

1. it is clearly a vector space over $\mathbb{R}$
2. it is endowed with a commutative ring structure by pointwise multiplication:

$$\forall f, g \in C^\infty(M), \quad \forall p \in M, \quad (f.g)(p) := f(p)g(p)$$

The algebra $C^\infty(M)$ is implemented in SageMath via the class `ScalarFieldAlgebra`.

Scalar field classes



Set of vector fields as a $C^\infty(M)$-module

The set $\mathfrak{X}(M)$ of vector fields on a smooth manifold M is endowed with 2 algebraic structures:

1. $\mathfrak{X}(M)$ is an infinite-dimensional vector space over $\mathbb{R}$, the scalar multiplication $\mathbb{R} \times \mathfrak{X}(M) \to \mathfrak{X}(M)$, $(\lambda, \boldsymbol{v}) \mapsto \lambda\boldsymbol{v}$ being defined by

$$\forall p \in M, \quad \left.(\lambda\boldsymbol{v})\right|_p = \lambda \left.\boldsymbol{v}\right|_p$$

2. $\mathfrak{X}(M)$ is a module[1] over the ring $C^\infty(M)$, the scalar multiplication $C^\infty(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M)$, $(f, \boldsymbol{v}) \mapsto f\boldsymbol{v}$ being defined by

$$\forall p \in M, \quad \left.(f\boldsymbol{v})\right|_p = f(p) \left.\boldsymbol{v}\right|_p ,$$

the r.h.s. involving the scalar mult. by $f(p) \in \mathbb{R}$ in the vector space T_pM

In SageMath, the second structure, i.e. $\mathfrak{X}(M)$ = module over $C^\infty(M)$, is adopted to implement $\mathfrak{X}(M)$

[1] Recall that a *module over a ring* generalizes the notion of *vector space over a field*

Free modules and vector frames

$\mathfrak{X}(M)$ is a ***free module*** over $C^\infty(M)$ $\iff$ $\mathfrak{X}(M)$ admits a basis

If this occurs, then $\mathfrak{X}(M)$ is actually a ***free module of finite rank*** over $C^\infty(M)$ and $\mathrm{rank}\,\mathfrak{X}(M) = \dim M = n$.
One says then that M is a ***parallelizable*** manifold.
A basis $(\boldsymbol{e}_a)_{1\le a\le n}$ of $\mathfrak{X}(M)$ is called a ***vector frame***; for any $p \in M$, $(\boldsymbol{e}_a|_p)_{1\le a\le n}$ is a basis of the tangent vector space T_pM.
Basis expansion:

$$\forall \boldsymbol{v} \in \mathfrak{X}(M), \quad \boldsymbol{v} = v^a \boldsymbol{e}_a, \quad \text{with } v^a \in C^\infty(M) \tag{1}$$

At each point $p \in M$, Eq. (1) gives birth to an identity in the vector space T_pM:

$$\boldsymbol{v}|_p = v^a(p)\ \boldsymbol{e}_a|_p\,, \quad \text{with } v^a(p) \in \mathbb{R},$$

Example:

If U is the domain of a coordinate chart $(x^a)_{1\le a\le n}$, $\mathfrak{X}(U)$ is a free module of rank n over $C^\infty(U)$, a basis of it being the coordinate frame $(\partial/\partial x^a)_{1\le a\le n}$.

Parallelizable manifolds

M is **parallelizable** $\iff$ $\mathfrak{X}(M)$ is a free $C^\infty(M)$-module of rank n
$\iff$ M admits a global vector frame
$\iff$ the tangent bundle is trivial: $TM \simeq M \times \mathbb{R}^n$

Parallelizable manifolds

M is **parallelizable** $\iff$ $\mathfrak{X}(M)$ is a free $C^\infty(M)$-module of rank n
$\iff$ M admits a global vector frame
$\iff$ the tangent bundle is trivial: $TM \simeq M \times \mathbb{R}^n$

Examples of parallelizable manifolds

- $\mathbb{R}^n$ (global coordinate chart $\Rightarrow$ global vector frame)
- the circle $\mathbb{S}^1$ (*rem:* no global coordinate chart)
- the torus $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$
- the 3-sphere $\mathbb{S}^3 \simeq \mathrm{SU}(2)$, as any Lie group
- the 7-sphere $\mathbb{S}^7$
- any orientable 3-manifold (Steenrod theorem)

Parallelizable manifolds

M is **parallelizable** $\iff$ $\mathfrak{X}(M)$ is a free $C^\infty(M)$-module of rank n
$\iff$ M admits a global vector frame
$\iff$ the tangent bundle is trivial: $TM \simeq M \times \mathbb{R}^n$

Examples of parallelizable manifolds

- $\mathbb{R}^n$ (global coordinate chart $\Rightarrow$ global vector frame)
- the circle $\mathbb{S}^1$ (*rem:* no global coordinate chart)
- the torus $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$
- the 3-sphere $\mathbb{S}^3 \simeq \mathrm{SU}(2)$, as any Lie group
- the 7-sphere $\mathbb{S}^7$
- any orientable 3-manifold (Steenrod theorem)

Examples of non-parallelizable manifolds

- the sphere $\mathbb{S}^2$ (hairy ball theorem!) and any n-sphere $\mathbb{S}^n$ with $n \notin \{1, 3, 7\}$
- the real projective plane $\mathbb{RP}^2$

Implementing vector and tensor fields

Ultimately, in SageMath, vector fields, and more generally tensor fields, are to be described by their components w.r.t. various vector frames.

Implementing vector and tensor fields

Ultimately, in SageMath, vector fields, and more generally tensor fields, are to be described by their components w.r.t. various vector frames.

Decomposition of M into parallelizable parts

Assumption: the smooth manifold M can be covered by a finite number m of parallelizable open subsets U_i $(1 \leq i \leq m)$

Example: this holds if M is compact (finite atlas)

We then consider **restrictions** of vector fields to the parallelizable subsets U_i:

Implementing vector and tensor fields

Ultimately, in SageMath, vector fields, and more generally tensor fields, are to be described by their components w.r.t. various vector frames.

Decomposition of M into parallelizable parts

Assumption: the smooth manifold M can be covered by a finite number m of parallelizable open subsets U_i $(1 \leq i \leq m)$

Example: this holds if M is compact (finite atlas)

We then consider **restrictions** of vector fields to the parallelizable subsets U_i:

For each i, $\mathfrak{X}(U_i)$ is a free module of rank $n = \dim M$ and is implemented in SageManifolds as an instance of `VectorFieldFreeModule`, which is a subclass of `FiniteRankFreeModule`.

Each vector field $\boldsymbol{v} \in \mathfrak{X}(U_i)$ has different set of components $(v^a)_{1 \leq a \leq n}$ in different vector frames $(\boldsymbol{e}_a)_{1 \leq a \leq n}$ introduced on U_i. They are stored as a *Python dictionary* whose keys are the vector frames:

$$\boldsymbol{v}.\texttt{\_components} = \{(\boldsymbol{e}) : (v^a),\ (\hat{\boldsymbol{e}}) : (\hat{v}^a), \ldots\}$$

Tensor field storage

TensorField T

dictionary `TensorField._restrictions`

- domain 1: U_1 — `TensorFieldParal` $T|_{U_1} = T^a{}_b e_a \otimes e^b = T^{\hat{a}}{}_{\hat{b}} \varepsilon_{\hat{a}} \otimes \varepsilon^{\hat{b}} = \ldots$
- domain 2: U_2 — `TensorFieldParal` $T|_{U_2}$
- $\ldots$

dictionary `TensorFieldParal._components`

- frame 1: (e_a) — `Components` $(T^a{}_b)_{1 \le a,\, b \le n}$
- frame 2: $(\varepsilon_{\hat{a}})$ — `Components` $(T^{\hat{a}}{}_{\hat{b}})_{1 \le \hat{a},\, \hat{b} \le n}$
- $\ldots$

dictionary `Components._comp`

- $(1,1)$: `DiffScalarField` $T^1{}_1$
- $(1,2)$: `DiffScalarField` $T^1{}_2$
- $\ldots$

dictionary `DiffScalarField._express`

- chart 1: (x^a) — `ChartFunction` $T^1{}_1\left(x^1, \ldots, x^n\right)$
- chart 2: (y^a) — `ChartFunction` $T^1{}_1\left(y^1, \ldots, y^n\right)$
- $\ldots$

dictionary `ChartFunction._express`

- SR: `Expression` $x^1 \cos x^2$
- SymPy: `Basic` $x^1 \cos x^2$
- $\ldots$

Outline

Near-horizon geometry of the extremal Kerr black hole

Extremal Kerr black hole: $a = m \iff \kappa = 0$ (degenerate horizon)

Near-horizon geometry of extremal 4D Kerr is similar to $\mathrm{AdS}_2 \times \mathbb{S}^2$ geometry; it has has extended isometry group: $\mathrm{SL}(2, \mathbb{R}) \times \mathrm{U}(1)$, instead of merely $\mathbb{R} \times \mathrm{U}(1)$ for Kerr metric [Bardeen & Horowitz, PRD **60**, 104030 (1999)]

Near-horizon geometry of extremal Kerr black hole is at the basis of the **Kerr/CFT correspondence** (see [Compère, LRR **20**, 1 (2017)] for a review)

Near-horizon geometry of the extremal Kerr black hole

Extremal Kerr black hole: $a = m \iff \kappa = 0$ (degenerate horizon)

Near-horizon geometry of extremal 4D Kerr is similar to $\mathrm{AdS}_2 \times \mathbb{S}^2$ geometry; it has has extended isometry group: $\mathrm{SL}(2, \mathbb{R}) \times \mathrm{U}(1)$, instead of merely $\mathbb{R} \times \mathrm{U}(1)$ for Kerr metric [Bardeen & Horowitz, PRD **60**, 104030 (1999)]

Near-horizon geometry of extremal Kerr black hole is at the basis of the **Kerr/CFT correspondence** (see [Compère, LRR **20**, 1 (2017)] for a review)

Let us explore this geometry with a SageMath notebook:
https://nbviewer.jupyter.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_extremal_Kerr_near_horizon.ipynb

(In the nbviewer menu, click on to run an interactive version on a Binder server)

Near-horizon geometry of the extremal Kerr black hole

This notebook derives the near-horizon geometry of the extremal (i.e. maximally spinning) Kerr black hole. It is based on SageMath tools developed through the SageManifolds project.

First we set up the notebook to display maths using LaTeX rendering and to perform computations in parallel on 8 threads:

In [1]:
```
%display latex
Parallelism().set(nproc=8)
```

Spacetime manifold

We declare the Kerr spacetime (or more precisely the part of it covered by Boyer-Lindquist coordinates) as a 4-dimensional Lorentzian manifold $\mathcal{M}$:

In [2]:
```
M = Manifold(4, 'M', latex_name=r'\mathcal{M}', structure='Lorentzian')
print(M)
```

```
4-dimensional Lorentzian manifold M
```

We then introduce the standard **Boyer-Lindquist coordinates** (t, r, θ, ϕ) as a chart `BL` (for *Boyer-Lindquist*) on $\mathcal{M}$:

In [3]:
```
BL.<t,r,th,ph> = M.chart(r"t r th:(0,pi):\theta ph:(0,2*pi):periodic:\phi")
print(BL); BL
```

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

Out[3]: $(\mathcal{M}, (t, r, \theta, \phi))$

Metric tensor of the extremal Kerr spacetime

The metric is set by its components in the coordinate frame associated with Boyer-Lindquist coordinates, which is the current manifold's default frame:

In [4]:
```
m = var('m', domain='real')
a = m  # extremal Kerr
rho2 = r^2 + (a*cos(th))^2
Delta = r^2 -2*m*r + a^2
g = M.metric()
g[0,0] = -(1-2*m*r/rho2)
g[0,3] = -2*a*m*r*sin(th)^2/rho2
g[1,1], g[2,2] = rho2/Delta, rho2
g[3,3] = (r^2+a^2+2*m*r*(a*sin(th))^2/rho2)*sin(th)^2
g.display()
```

Out[4]:

$$g = \left(\frac{2\,mr}{m^2\cos(\theta)^2 + r^2} - 1\right) \mathrm{d}t \otimes \mathrm{d}t + \left(-\frac{2\,m^2 r \sin(\theta)^2}{m^2\cos(\theta)^2 + r^2}\right) \mathrm{d}t \otimes \mathrm{d}\phi + \left(\frac{m^2\cos(\theta)^2 + r^2}{m^2 - 2\,mr + r^2}\right) \mathrm{d}r \otimes \mathrm{d}r + \left(m^2\cos(\theta)^2 + r^2\right) \mathrm{d}\theta \otimes \mathrm{d}\theta + \left(-\frac{2\,m^2 r \sin(\theta)^2}{m^2\cos(\theta)^2 + r^2}\right) \mathrm{d}\phi \otimes \mathrm{d}t + \left(\frac{2\,m^3 r \sin(\theta)^2}{m^2\cos(\theta)^2 + r^2} + m^2 + r^2\right) \sin(\theta)^2 \mathrm{d}\phi \otimes \mathrm{d}\phi$$

Check that we are dealing with a solution of the vacuum Einstein equation:

In [5]:
```
g.ricci().display()
```

Out[5]: $\mathrm{Ric}(g) = 0$

Near-horizon coordinates

Let us introduce the chart `NH` of the near-horizon coordinates $(\bar{t}, \bar{r}, \theta, \bar{\phi})$:

In [6]:
```
NH.<tb,rb,th,phb> = M.chart(r"tb:\bar{t} rb:\bar{r} th:(0,pi):\theta phb:(0,2*pi):periodic:\
print(NH)
NH
```

```
Chart (M, (tb, rb, th, phb))
```

Out[6]: $(\mathcal{M}, (\bar{t}, \bar{r}, \theta, \bar{\phi}))$

Following J. Bardeen and G. T. Horowitz, Phys. Rev. D **60**, 104030 (1999), the near-horizon coordinates $(\bar{t}, \bar{r}, \theta, \bar{\phi})$ are related to the Boyer-Lindquist coordinates by

$$\bar{t} = \epsilon t, \quad \bar{r} = \frac{r-m}{\epsilon}, \quad \theta = \theta, \quad \bar{\phi} = \phi - \frac{t}{2m},$$

where ϵ is a constant parameter. The horizon of the extremal Kerr black hole is located at $r = m$, which corresponds to $\bar{r} = 0$.

We implement the above relations as a transition map from the chart `BL` to the chart `NH`:

In [7]:
```
eps = var('eps', latex_name=r'\epsilon')
BL_to_NH = BL.transition_map(NH, [eps*t, (r-m)/eps, th, ph - t/(2*m)])
BL_to_NH.display()
```

Out[7]:

$$\begin{cases} \bar{t} & = & \epsilon t \\ \bar{r} & = & -\frac{m-r}{\epsilon} \\ \theta & = & \theta \\ \bar{\phi} & = & \phi - \frac{t}{2\,m} \end{cases}$$

The inverse relation is

In [8]:
```
BL_to_NH.inverse().display()
```

Out[8]:

$$\begin{cases} t & = & \frac{\bar{t}}{\epsilon} \\ r & = & \epsilon\bar{r} + m \\ \theta & = & \theta \\ \phi & = & \frac{2\,\epsilon m\bar{\phi} + \bar{t}}{2\,\epsilon m} \end{cases}$$

The metric components with respect the coordinates $(\bar{t}, \bar{r}, \theta, \bar{\phi})$ are computed by passing the chart `NH` to the method `display()`:

In [9]:
```
g.display(NH)
```

Out[9]:

$$g = \left(-\frac{m^2\bar{r}^2\cos(\theta)^4 - \epsilon^2\bar{r}^4 - 4\,\epsilon m\bar{r}^3 - 3\,m^2\bar{r}^2 + \left(\epsilon^2\bar{r}^4 + 4\,\epsilon m\bar{r}^3 + 6\,m^2\bar{r}^2\right)\cos(\theta)^2}{4\left(\epsilon^2 m^2\bar{r}^2 + m^4\cos(\theta)^2 + 2\,\epsilon m^3\bar{r} + m^4\right)}\right)\mathrm{d}\bar{t}\otimes\mathrm{d}\bar{t}$$
$$+\left(-\frac{\epsilon m^2\bar{r}^2\sin(\theta)^4 - \left(\epsilon^3\bar{r}^4 + 4\,\epsilon^2 m\bar{r}^3 + 8\,\epsilon m^2\bar{r}^2 + 4\,m^3\bar{r}\right)\sin(\theta)^2}{2\left(\epsilon^2 m\bar{r}^2 + m^3\cos(\theta)^2 + 2\,\epsilon m^2\bar{r} + m^3\right)}\right)\mathrm{d}\bar{t}\otimes\mathrm{d}\bar{\phi}$$
$$+\left(\frac{\epsilon^2\bar{r}^2 + m^2\cos(\theta)^2 + 2\,\epsilon m\bar{r} + m^2}{\bar{r}^2}\right)\mathrm{d}\bar{r}\otimes\mathrm{d}\bar{r} + \left(\epsilon^2\bar{r}^2 + m^2\cos(\theta)^2 + 2\,\epsilon m\bar{r} + m^2\right)\mathrm{d}\theta\otimes\mathrm{d}\theta$$
$$+\left(-\frac{\epsilon m^2\bar{r}^2\sin(\theta)^4 - \left(\epsilon^3\bar{r}^4 + 4\,\epsilon^2 m\bar{r}^3 + 8\,\epsilon m^2\bar{r}^2 + 4\,m^3\bar{r}\right)\sin(\theta)^2}{2\left(\epsilon^2 m\bar{r}^2 + m^3\cos(\theta)^2 + 2\,\epsilon m^2\bar{r} + m^3\right)}\right)\mathrm{d}\bar{\phi}\otimes\mathrm{d}\bar{t}$$
$$+\left(-\frac{\epsilon^2 m^2\bar{r}^2\sin(\theta)^4 - \left(\epsilon^4\bar{r}^4 + 4\,\epsilon^3 m\bar{r}^3 + 8\,\epsilon^2 m^2\bar{r}^2 + 8\,\epsilon m^3\bar{r} + 4\,m^4\right)\sin(\theta)^2}{\epsilon^2\bar{r}^2 + m^2\cos(\theta)^2 + 2\,\epsilon m\bar{r} + m^2}\right)\mathrm{d}\bar{\phi}\otimes\mathrm{d}\bar{\phi}$$

From now on, we use the near-horizon coordinates as the default ones on the spacetime manifold:

In [10]:
```
M.set_default_chart(NH)
M.set_default_frame(NH.frame())
```

The near-horizon metric h as the limit $\epsilon \to 0$ of the Kerr metric g

Let us define the *near-horizon metric* as the metric h on $\mathcal{M}$ that is the limit $\epsilon \to 0$ of the Kerr metric g. The limit is taken by asking for a series expansion of g with respect to ϵ up to the 0-th order (i.e. keeping only ϵ^0 terms). This is acheived via the method `truncate`:

In [11]:
```
h = M.lorentzian_metric('h')
h.set(g.truncate(eps, 0))
h.display()
```

Out[11]:

$$h = \left(-\frac{\bar{r}^2 \cos(\theta)^4 + 6\,\bar{r}^2 \cos(\theta)^2 - 3\,\bar{r}^2}{4\left(m^2 \cos(\theta)^2 + m^2\right)} \right) \mathrm{d}\bar{t} \otimes \mathrm{d}\bar{t} + \left(\frac{2\,\bar{r} \sin(\theta)^2}{\cos(\theta)^2 + 1} \right) \mathrm{d}\bar{t} \otimes \mathrm{d}\bar{\phi} + \left(\frac{m^2 \cos(\theta)^2 + m^2}{\bar{r}^2} \right) \mathrm{d}\bar{r} \otimes \mathrm{d}\bar{r} + \left(m^2 \cos(\theta)^2 + m^2\right) \mathrm{d}\theta \otimes \mathrm{d}\theta + \left(\frac{2\,\bar{r} \sin(\theta)^2}{\cos(\theta)^2 + 1} \right) \mathrm{d}\bar{\phi} \otimes \mathrm{d}\bar{t} + \left(\frac{4\,m^2 \sin(\theta)^2}{\cos(\theta)^2 + 1} \right) \mathrm{d}\bar{\phi} \otimes \mathrm{d}\bar{\phi}$$

We note that the metric h is not asymptotically flat.

Killing vectors of the near-horizon geometry

Let us first consider the vector field $\eta := \frac{\partial}{\partial \bar{\phi}}$:

In [12]:
```
eta = M.vector_field(0, 0, 0, 1, name='eta', latex_name=r'\eta')
eta.display()
```

Out[12]:

$$\eta = \frac{\partial}{\partial \bar{\phi}}$$

It is a Killing vector of the near-horizon metric, since the Lie derivative of h along η vanishes:

In [13]:
```
h.lie_derivative(eta).display()
```

Out[13]: 0

This is not surprising since the components of h are independent from $\bar{\phi}$.

Similarly, we can check that $\xi_1 := \frac{\partial}{\partial \bar{t}}$ is a Killing vector of h, reflecting the independence of the components of h from $\bar{t}$:

In [14]:
```
xi1 = M.vector_field(1, 0, 0, 0, name='xi2', latex_name=r'\xi_{1}')
xi1.display()
```

Out[14]:

$$\xi_1 = \frac{\partial}{\partial \bar{t}}$$

In [15]:
```
h.lie_derivative(xi1).display()
```

Out[15]: 0

The above two Killing vectors correspond respectively to the **axisymmetry** and the **pseudo-stationarity** of the Kerr metric. A third symmetry, which is not present in the original Kerr metric, is the invariance under the **scaling** $(\bar{t}, \bar{r}) \mapsto (\alpha\bar{t}, \bar{r}/\alpha)$, as it is clear on the metric components in Out[11]. The corresponding Killing vector is

In [16]:
```
xi2 = M.vector_field(tb, -rb, 0, 0, name='xi2', latex_name=r'\xi_{2}')
xi2.display()
```

Out[16]:

$$\xi_2 = \bar{t}\frac{\partial}{\partial\bar{t}} - \bar{r}\frac{\partial}{\partial\bar{r}}$$

In [17]:
```
h.lie_derivative(xi2).display()
```

Out[17]: 0

Finally, a fourth Killing vector is

In [18]:
```
xi3 = M.vector_field(tb^2/2 + 2*m^4/rb^2, -tb*rb, 0, -2*m^2/rb,
                     name='xi3', latex_name=r'\xi_{3}')
xi3.display()
```

Out[18]:

$$\xi_3 = \left(\frac{2\,m^4}{\bar{r}^2} + \frac{1}{2}\,\bar{t}^2\right)\frac{\partial}{\partial\bar{t}} - \bar{r}\bar{t}\frac{\partial}{\partial\bar{r}} - \frac{2\,m^2}{\bar{r}}\frac{\partial}{\partial\bar{\phi}}$$

In [19]:
```
h.lie_derivative(xi3).display()
```

Out[19]: 0

Symmetry group

We have four independent Killing vectors, η, ξ_1, ξ_2 and ξ_3, which implies that the symmetry group of the near-horizon geometry is a 4-dimensional Lie group G. Let us determine G by investigating the **structure constants** of the basis $(\eta, \xi_1, \xi_2, \xi_3)$ of the Lie algebra of G. First of all, we notice that η commutes with the other Killing vectors:

In [20]:
```
for xi in [xi1, xi2, xi3]:
    show(eta.bracket(xi).display())
```

$$[\eta, \xi_1] = 0$$

$$[\eta, \xi_2] = 0$$

$$[\eta, \xi_3] = 0$$

Since η generates the rotation group $\mathrm{SO}(2) = \mathrm{U}(1)$, we may write that $G = \mathrm{U}(1) \times G_3$, where G_3 is a 3-dimensional Lie group, whose generators are (ξ_1, ξ_2, ξ_3). Let us determine the structure constants of these three vectors. We have

In [21]:
```
xi1.bracket(xi2).display()
```

Out[21]:

$$[\xi_1, \xi_2] = \frac{\partial}{\partial \bar{t}}$$

In [22]:
```
xi1.bracket(xi3).display()
```

Out[22]:

$$[\xi_1, \xi_3] = \bar{t}\frac{\partial}{\partial \bar{t}} - \bar{r}\frac{\partial}{\partial \bar{r}}$$

In [23]:
```
xi2.bracket(xi3).display()
```

Out[23]:

$$[\xi_2, \xi_3] = \left(\frac{4\,m^4 + \bar{r}^2\bar{t}^2}{2\,\bar{r}^2}\right)\frac{\partial}{\partial \bar{t}} - \bar{r}\bar{t}\frac{\partial}{\partial \bar{r}} - \frac{2\,m^2}{\bar{r}}\frac{\partial}{\partial \bar{\phi}}$$

To summarize, we have

In [24]:
```
all([xi1.bracket(xi2) == xi1,
     xi1.bracket(xi3) == xi2,
     xi2.bracket(xi3) == xi3])
```

Out[24]: True

To recognize a standard Lie algebra, let us perform a slight change of basis:

In [25]:
```
vE = -sqrt(2)*xi3
vF = sqrt(2)*xi1
vH = 2*xi2
```

We have then the following commutation relations:

In [26]:
```
all([vE.bracket(vF) == vH,
     vH.bracket(vE) == 2*vE,
     vH.bracket(vF) == -2*vF])
```

Out[26]: True

We recognize the Lie algebra $\mathfrak{sl}(2,\mathbb{R})$. Indeed, we have

In [27]:
```
sl2 = lie_algebras.sl(RR, 2)
E,F,H = sl2.gens()
all([E.bracket(F) == H,
     H.bracket(E) == 2*E,
     H.bracket(F) == -2*F])
```

Out[27]: True

Hence, we have

$$\mathrm{Lie}(G_3) = \mathfrak{sl}(2, \mathbb{R}).$$

At this stage, G_3 could be $\mathrm{SL}(2, \mathbb{R})$, $\mathrm{PSL}(2, \mathbb{R})$ or $\overline{\mathrm{SL}(2, \mathbb{R})}$ (the universal covering group of $\mathrm{SL}(2, \mathbb{R})$). One can show that actually $G_3 = \mathrm{SL}(2, \mathbb{R})$. We conclude that the full isometry group of the near-horizon geometry is

$$G = \mathrm{U}(1) \times \mathrm{SL}(2, \mathbb{R}).$$

The full notebook is available at
https://nbviewer.jupyter.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_extremal_Kerr_near_horizon.ipynb

(in the nbviewer menu, click on the icon to run an interactive version on a Binder server)

Outline

Computation of geodesics in Kerr spacetime

https://nbviewer.jupyter.org/github/BlackHolePerturbationToolkit/kerrgeodesic_gw/blob/master/Notebooks/Kerr_geodesics.ipynb

(In the nbviewer menu, click on to run an interactive version on a Binder server)

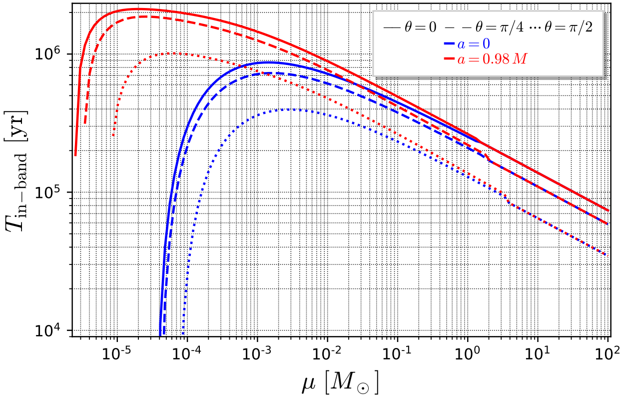
Gravitational waves from circular orbits around a Kerr black hole

https://nbviewer.jupyter.org/github/BlackHolePerturbationToolkit/kerrgeodesic_gw/blob/master/Notebooks/grav_waves_circular.ipynb

Application: Gravitational waves from bodies orbiting the Galactic Center black hole and their detectability by LISA

[Gourgoulhon, Le Tiec, Vincent & Warburton, A&A **627**, A92 (2019)]

Time in LISA band with $\mathrm{SNR}_{1\,\mathrm{yr}} \geq 10$ for an inspiralling compact object



μ: mass of the inspiralling compact object

Primordial BHs with $1M_\oplus \leq \mu \leq 5M_{\mathrm{Jup}}$ spend more than 10^6 yr in LISA band with $\mathrm{SNR}_{1\,\mathrm{yr}} \geq 10$

[Gourgoulhon, Le Tiec, Vincent & Warburton, A&A **627**, A92 (2019)]

Time in LISA band $\mathrm{SNR}_{1\,\mathrm{yr}} \geq 10$ for brown dwarfs and main-sequence stars

Results for

- inclination angle $\theta = 0$
- BH spin $a = 0$ (outside parentheses) and $a = 0.98M$ (inside parentheses)

	brown dwarf	red dwarf	Sun-type	$2.4\,M_\odot$-star
$\mu/M_\odot$	0.062	0.20	1	2.40
$\rho/\rho_\odot$	$131.$	18.8	1	0.367
$r_{0,\max}/M$	28.2 (28.0)	35.0 (34.9)	47.1 (47.0)	55.6 (55.6)
$f_{m=2}(r_{0,\max})$ [mHz]	0.105 (0.106)	0.076 (0.076)	0.049 (0.049)	0.038 (0.038)
r_{Roche}/M	7.31 (6.93)	13.3 (13.0)	34.2 (34.1)	47.6 (47.5)
$T^{\mathrm{ins}}_{\text{in-band}}$ $[10^5\ \mathrm{yr}]$	4.98 (5.55)	3.72 (3.99)	1.83 (1.89)	0.938 (0.945)

Brown dwarfs stay for $\sim 5 \times 10^5$ yr in LISA band

Outline

Image of an accretion disk around a Schwarzschild BH

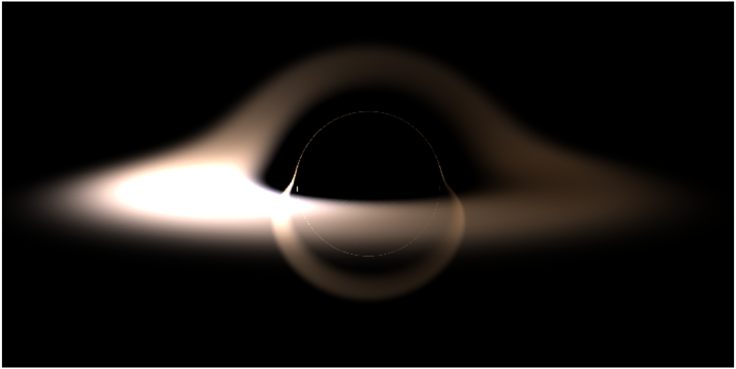
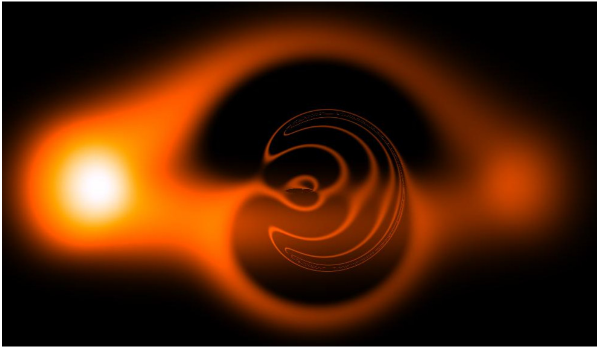
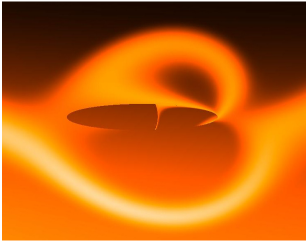


Image entirely computed with SageMath by integrating the null geodesics, cf. the notebook
https://nbviewer.jupyter.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_black_hole_rendering.ipynb

Naked rotating wormhole

Regular (singularity-free) spacetime with wormhole topology ($\mathbb{R}^2 \times \mathbb{S}^2$), sustained by exotic matter, asymptotically close a to Kerr spacetime with a naked singularity ($a > M$) and surrounded by an accretion torus





zoom on the central region

[Lamy, Gourgoulhon, Paumard & Vincent, CQG **35**, 115009 (2018)]

- Derivation of the geodesic equation: `SageMath`
- Integration of the geodesic equation: `Gyoto`

Outline

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)
- is not limited to a single coordinate chart or vector frame

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)
- is not limited to a single coordinate chart or vector frame
- runs even on non-parallelizable manifolds

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)
- is not limited to a single coordinate chart or vector frame
- runs even on non-parallelizable manifolds
- is independent of the symbolic engine (e.g. *Pynac/Maxima*, *SymPy*,...) used to perform calculus at the level of coordinate expressions

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)
- is not limited to a single coordinate chart or vector frame
- runs even on non-parallelizable manifolds
- is independent of the symbolic engine (e.g. *Pynac/Maxima*, *SymPy*,...) used to perform calculus at the level of coordinate expressions

Conclusions

Symbolic tensor calculus in the free Python-based system SageMath

- runs on fully specified smooth manifolds (described by an atlas)
- is not limited to a single coordinate chart or vector frame
- runs even on non-parallelizable manifolds
- is independent of the symbolic engine (e.g. *Pynac/Maxima*, *SymPy*,...) used to perform calculus at the level of coordinate expressions

Many more examples than shown in this talk are available at
`https://sagemanifolds.obspm.fr/examples.html`

Want to join the SageManifolds project or simply to stay tuned?

visit `https://sagemanifolds.obspm.fr/`
(download, documentation, example notebooks, mailing list)