# SageManifolds

## A free package for differential geometry

Éric Gourgoulhon[1], Michał Bejger[2]

[1] Laboratoire Univers et Théories (LUTH)
CNRS / Observatoire de Paris / Université Paris Diderot
92190 Meudon, France
http://luth.obspm.fr/~luthier/gourgoulhon/

[2] Centrum Astronomiczne im. M. Kopernika (CAMK)
Warsaw, Poland
http://users.camk.edu.pl/bejger/

**Asymptotic Analysis in General Relativity**
Université de Cergy-Pointoise, France
14 June 2013

# Outline

# Outline

# Sage in a few words

- Sage is a **free open-source** mathematics software
- it is based on the Python programming language
- it makes use of **many pre-existing open-sources packages**, among which
  - Maxima (symbolic calculations, since 1967 !)
  - GAP (group theory)
  - PARI/GP (number theory)
  - Singular (polynomial computations)
  - matplotlib (high quality figures)

  and provides a uniform interface to them
- William Stein (Univ. of Washington) created Sage in 2005; since then, ∼**150 developers** have joined the Sage team

**The mission**

*Create a viable free open source alternative to Magma, Maple, Mathematica and Matlab.*

# Advantages of Sage

## Sage is free

Freedom means

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

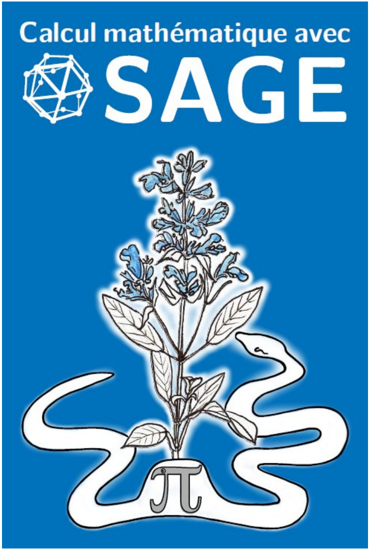
## Sage is based on Python

- no need to learn a specific syntax to use it
- easy access for students
- Python is a very powerful object oriented language, with a neat syntax

## Sage is developing and spreading fast

...sustained by an important community of developers

# The Sage book




by Paul Zimmermann et al.

Just published ! (May 2013)
Released under *Creative Commons* license:

- freely downloadable from http://sagebook.gforge.inria.fr/
- printed copies can be ordered at moderate price (10 €)

English translation in progress...

# Outline

# Existing softwares for differential geometry

**Packages for proprietary softwares:**

- xAct free package for Mathematica
- DifferentialGeometry included in Maple
- Atlas 2 for Maple
- ...

**Standalone softwares:**

- Cadabra field theory (free)
- SnapPy topology and geometry of 3-manifolds (Python) (free)
- ...

# The situation in Sage

Sage is well developed in many domains of mathematics:
number theory, group theory, linear algebra, etc.

but nothing is implemented for differential geometry, except for differential forms on an open subset of Euclidean space with a specific set of coordinates.

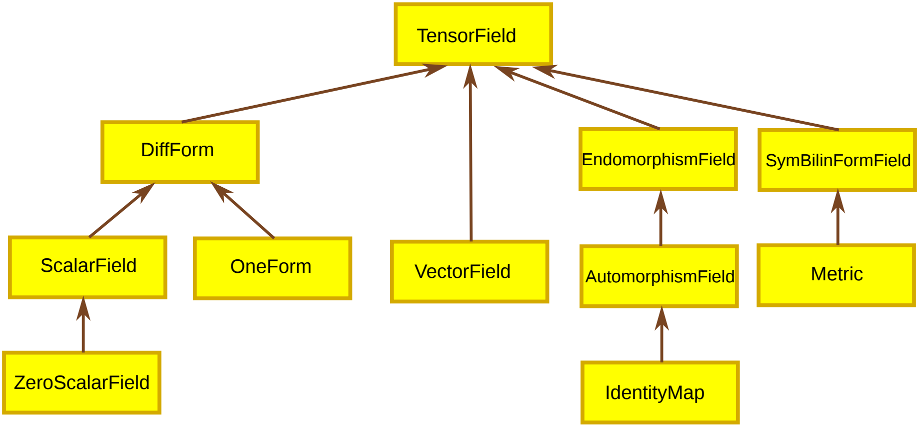
Hence the SageManifolds project

# SageManifolds

A new set of *Python classes* implementing differential geometry in Sage:

- `Manifold`: differentiable manifolds over $\mathbb{R}$
- `SubManifold`, `Curves`: submanifolds
- `Point`: points on a manifold
- `Chart`: charts
- `DiffMapping`, `Diffeomorphism`: differential mappings between manifolds
- `ScalarField`: differential mappings to $\mathbb{R}$
- `TensorField`, `VectorField`, `SymBilinFormField`, etc.: tensor fields on a manifold
- `DiffForm`, `OneForm`: $p$-forms
- `VectorFrame`, `CoordBasis`: vector frames on a manifold, including tetrads and coordinate bases
- `Components`, `CompWithSym`, etc.: components of a tensor field in a given vector frame
- `AffConnection`, `LeviCivitaConnection`: affine connections
- `Metric`: pseudo-Riemannian metrics

# Inheritance diagram of the tensor field classes



# Basic SageManifolds objects are coordinate-free

As a mapping $\mathcal{M} \to \mathbb{R}$, an object $f$ in the `ScalarField` class has different coordinate representations in different charts defined on the manifold $\mathcal{M}$.

These coordinate representations are stored as a *Python dictionary* whose keys are the names of the various charts:

$$f.\mathrm{express} = \left\{ C : F,\ \hat{C} : \hat{F}, \ldots \right\}$$

with $f(p) = F(\underbrace{x^1, \ldots, x^n}_{\text{chart } C}) = \hat{F}(\underbrace{\hat{x}^1, \ldots, \hat{x}^n}_{\text{chart } \hat{C}}) = \ldots$

# Basic SageManifolds objects are coordinate-free

An object $\boldsymbol{T}$ in the TensorField class has different set of components $T^{i\dots}{}_{j\dots}$ in different vector frames, each component being itself an object of the ScalarField field class, since

$$T^{i\dots}{}_{j\dots} = \boldsymbol{T}(\boldsymbol{e}^i, \dots, \boldsymbol{e}_j, \dots)$$

where $(\boldsymbol{e}_j)$ stands for the vector frame and $(\boldsymbol{e}^i)$ for the dual coframe.

The various sets of components are stored as a *Python dictionary* whose keys are the names of the various vector frames:

$$\boldsymbol{T}.\text{components} = \left\{ (\boldsymbol{e}) : (T^{i\dots}{}_{j\dots}),\ (\hat{\boldsymbol{e}}) : (\hat{T}^{i\dots}{}_{j\dots}), \dots \right\}$$

# SageManifolds at work

SM_Kerr -- Sage - Mozilla Firefox

Pseudo-Riemannian metric... | Sage: Open Source Mathem... | SM_Kerr – Sage

localhost:8080/home/admin/58/

## Curvature

The Ricci tensor associated with $g$:

```
Ric = g.ricci() ; print Ric
```

```
field of symmetric bilinear forms 'Ric(g)' on the 4-dimensional manifold
'M'
```

Let us check that Kerr metric is a solution of the vacuum Einstein equation:

```
Ric == 0
```

True

```
Ric.show() # another view of the above property
```

$$\mathrm{Ric}(g) = 0$$

The Riemann curvature tensor associated with $g$:

```
R = g.riemann() ; print R
```

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

Contrary to the Ricci tensor, the Riemann tensor does not vanish; for instance, the component ${R^0}_{123}$ is

```
R[0,1,2,3]
```

$$-\frac{\left(a^{7}m-2\,a^{5}m^{2}r+a^{5}mr^{2}\right)\sin\left(\theta\right)^{5}\cos(\theta)+\left(a^{7}m+2\,a^{5}m^{2}r+6\,a^{5}mr^{2}-6\,a^{3}m^{2}r^{3}+5\,a^{3}mr^{4}\right)\sin\left(\theta\right)^{3}\cos(\theta)-2\left(a^{7}m-a^{5}mr^{2}-5\,a^{3}mr^{4}-3\,amr^{6}\right)\sin(\theta)\cos(\theta)}{a^{2}r^{6}-2\,mr^{7}+r^{8}+\left(a^{8}-2\,a^{6}mr+a^{6}r^{2}\right)\cos\left(\theta\right)^{6}+3\left(a^{6}r^{2}-2\,a^{4}mr^{3}+a^{4}r^{4}\right)\cos\left(\theta\right)^{4}+3\left(a^{4}r^{4}-2\,a^{2}mr^{5}+a^{2}r^{6}\right)\cos\left(\theta\right)^{2}}$$

## Bianchi identity

# Outline

# Perspectives

- **SageManifolds** is a work in progress ($\sim$ 14,000 lines of Python code up to now)
- A preliminary version should be released in the coming weeks at `http://sagemanifolds.obspm.fr/` (page under construction)
- *Already present:* standard tensor calculus (tensor product, contraction, symmetrization, etc.), exterior calculus, Lie derivative, affine connection, curvature, torsion, pseudo-Riemannian metric, Weyl tensor,...
- *Not implemented yet (but should be soon):* pullback and pushforward operators, Hodge duality, extrinsic geometry of submanifolds
- *To do:* convert some parts to *Cython* in order to compile them (*C* code) and increase the computational speed
- *For future releases:* symplectic forms, fibre bundles, spinors, variational calculus