Geometry of Killing horizons and applications to black hole physics

5. Exploring the extremal Kerr near-horizon geometry with SageMath

Éric Gourgoulhon

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

https://relativite.obspm.fr/blackholes/ihp24/

Quantum and classical fields interacting with geometry
Institut Henri Poincaré, Paris, France
March - April 2024

Outline

Outline

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system initiated by William Stein in 2005

SageMath in a few words

SageMath (*nickname:* Sage) is a **free open-source** computer algebra system initiated by William Stein in 2005

SageMath is free (GPL v2)

Freedom means

1. everybody can use it, by download 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 initiated by William Stein in 2005

SageMath is free (GPL v2)

Freedom means

1. everybody can use it, by download 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 initiated by William Stein in 2005

SageMath is free (GPL v2)

Freedom means

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

SageMath is based on Python

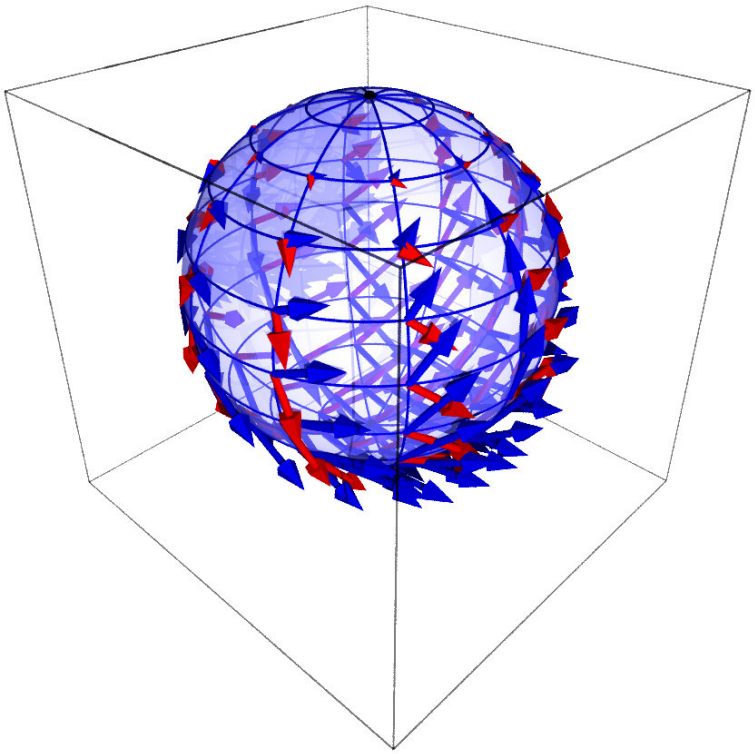
- no need to learn any specific syntax to use it
- Python is a powerful *object oriented language*, with a neat syntax
- SageMath benefits from the Python ecosystem (e.g. Jupyter notebook, NumPy, Matplotlib)

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
- $\sim$ 119,000 lines of Python code
- fully included in SageMath (after review process)
- $\sim$ 30 contributors (developers and reviewers) cf. https://sagemanifolds.obspm.fr/authors.html
- dedicated mailing list
- help desk: https://ask.sagemath.org

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

Current status

Already present (SageMath 10.2):

- 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 derivative along a vector field
- 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 by numerical integration

Current status

Already present *(cont'd)*:

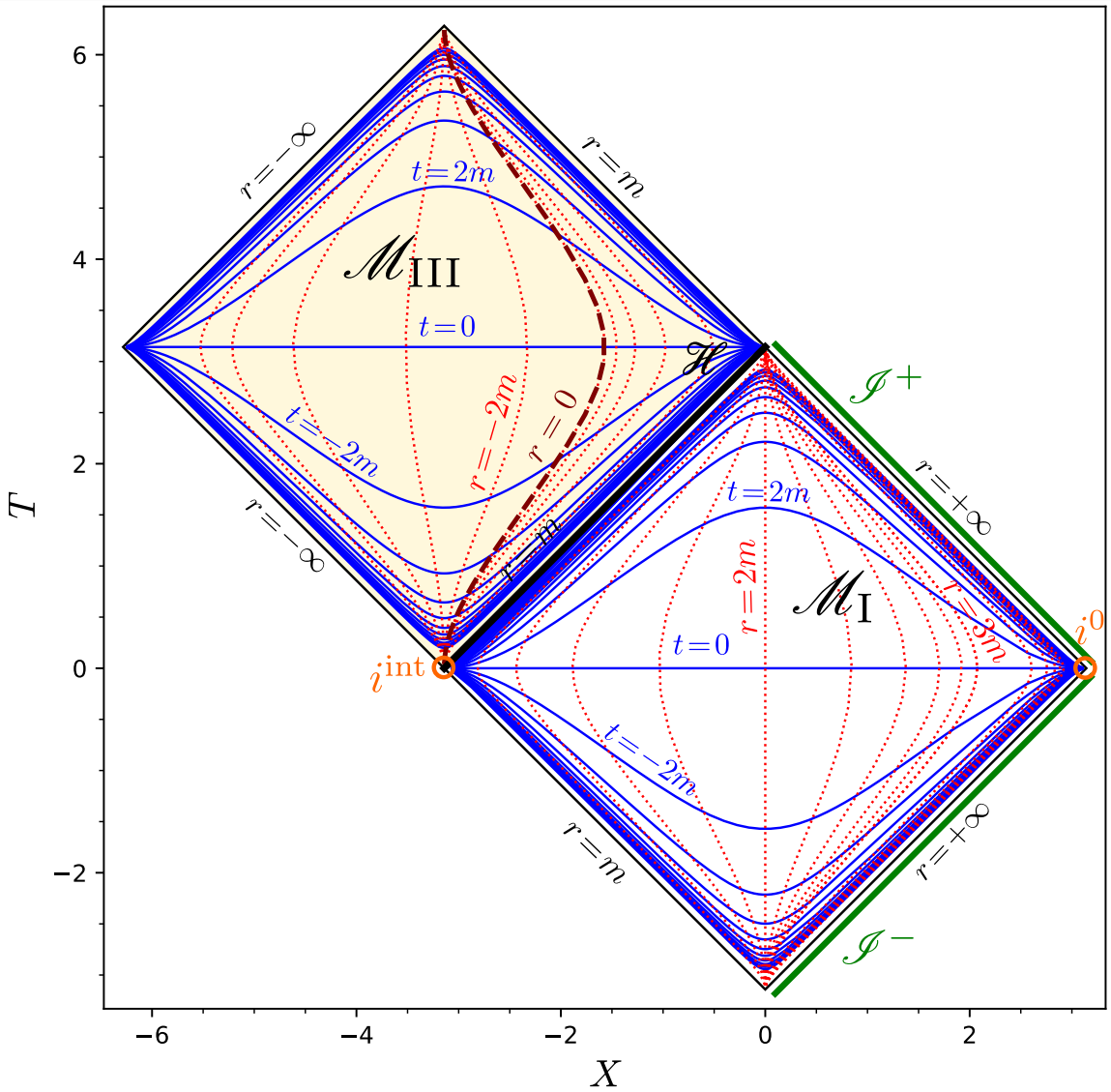
- some plotting capabilities (charts, points, curves, vector fields)
- parallelization (on tensor components) of CPU demanding computations
- extrinsic geometry of pseudo-Riemannian submanifolds
- series expansions of tensor fields
- symplectic manifolds
- 2 symbolic backends: Pynac/Maxima (SageMath's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

Future prospects:

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

Outline

Extremal Kerr spacetime

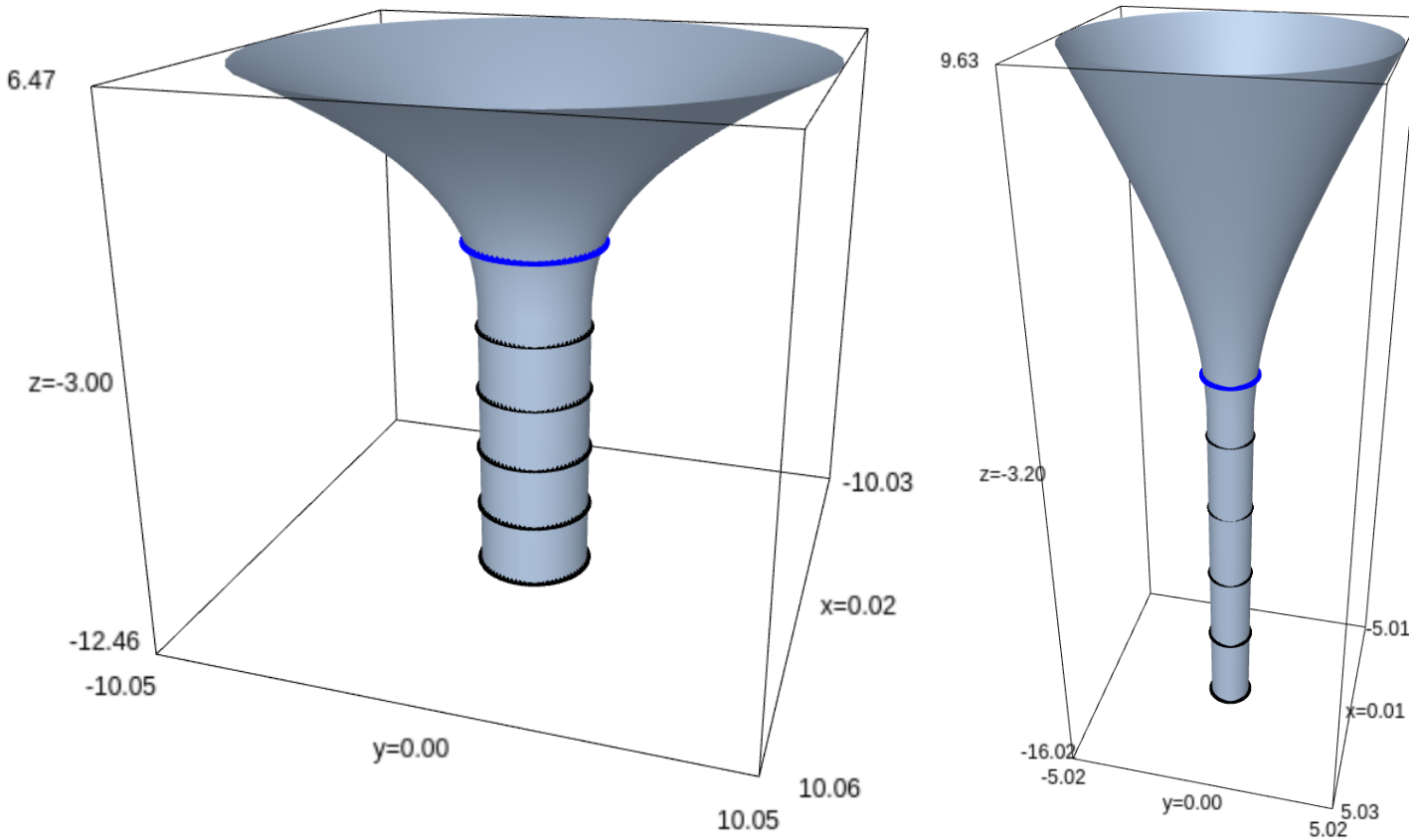


Carter-Penrose diagram of a single patch of Boyer-Lindquist coordinates (t, r, θ, φ) of extremal Kerr spacetime $(a = m)$.

https://nbviewer.jupyter.org/github/egourgoulhon/BHLectures/blob/master/sage/Kerr_extremal.ipynb

Extremal Kerr throat

Isometric embeddings of 2-surfaces $(t, \theta) = \text{const}$ in the Euclidean 3-space for $\theta = \pi/2$ (equatorial plane) (left) and $\theta = \pi/6$ (right).



Blue circle: ergosphere
Black circles: $r = 1.1\,m,\ 1.01\,m,\ 1.001\,m,\ 1.0001\,m,\ 1.00001\,m$

Near-horizon geometry of the extremal Kerr black hole

Near-horizon geometry of extremal Kerr (NHEK) discovered by Carter (1973). It has been rediscovered and studied in depth by Bardeen and Horowitz (1999).

Near-horizon geometry of the extremal Kerr black hole

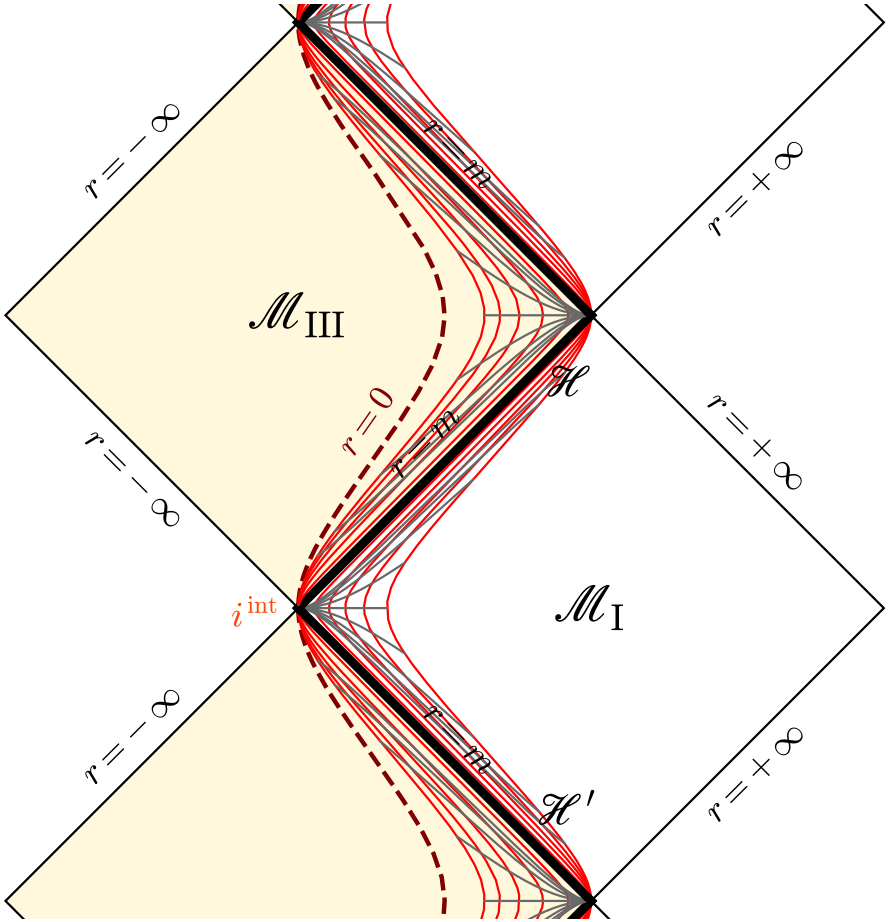
Near-horizon geometry of extremal Kerr (NHEK) discovered by Carter (1973). It has been rediscovered and studied in depth by Bardeen and Horowitz (1999).

SageMath Jupyter notebooks:

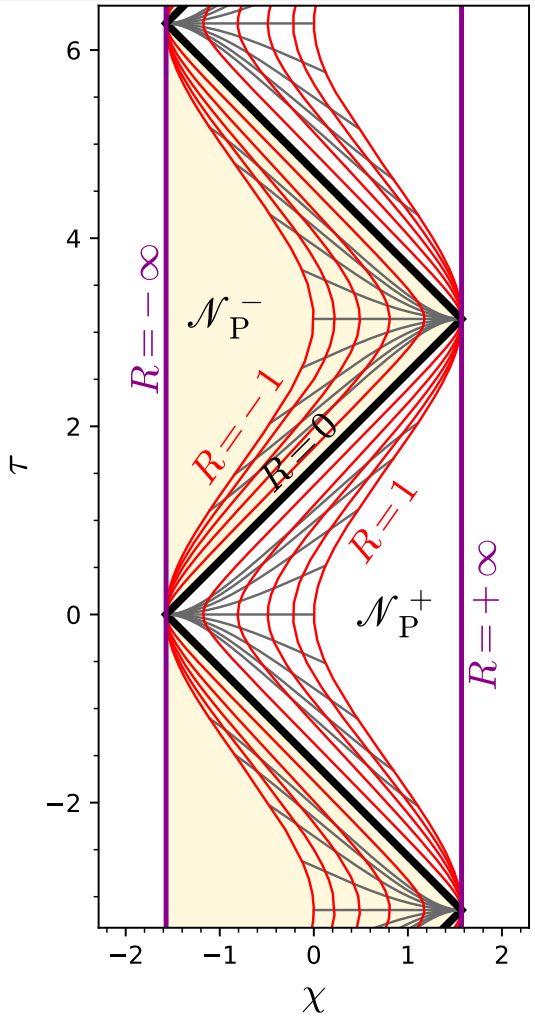
- Deriving the NHEK metric from the extremal Kerr one and investigating its symmetries:
  https://nbviewer.jupyter.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_extremal_Kerr_near_horizon.ipynb
- The NHEK spacetime for its own sake:
  https://nbviewer.jupyter.org/github/egourgoulhon/BHLectures/blob/master/sage/NHEK_spacetime.ipynb

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

Carter-Penrose diagrams generated with SageMath



Extremal Kerr



NHEK spacetime

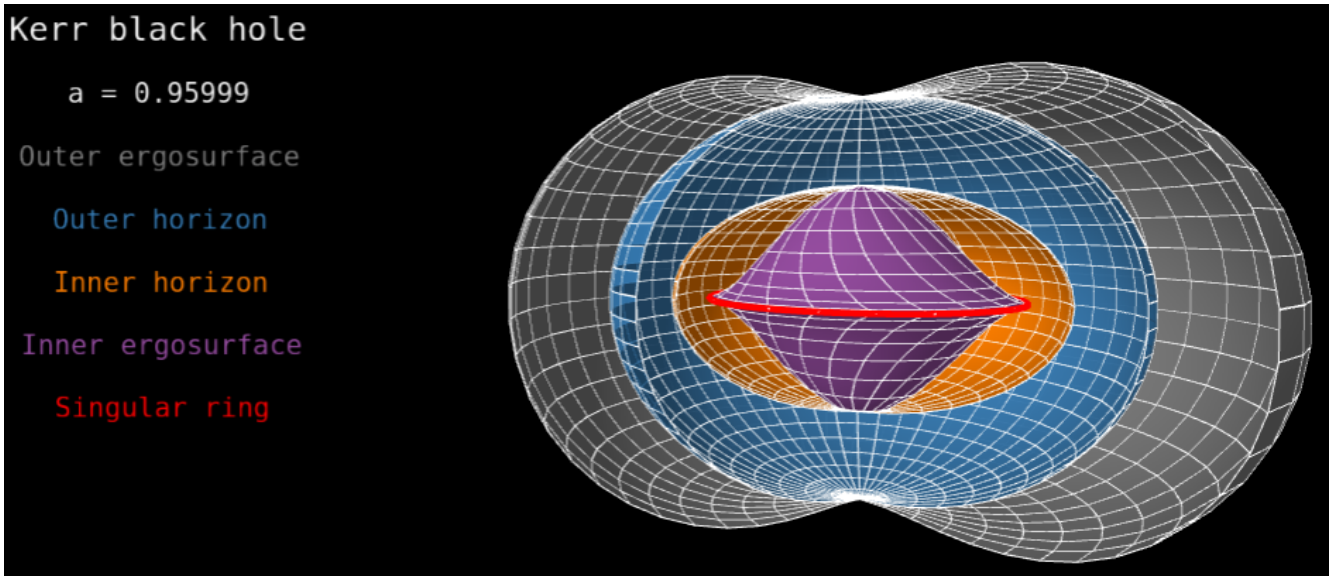
https://nbviewer.org/github/egourgoulhon/BHLectures/blob/master/sage/Kerr_extremal_extended.ipynb
https://nbviewer.org/github/egourgoulhon/BHLectures/blob/master/sage/NHEK_spacetime.ipynb

Outline

Other examples

- **Schwarzschild spacetime** (many coordinates): https://nbviewer.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_Schwarzschild.ipynb
- **Computation of geodesics in Kerr spacetime**: https://nbviewer.jupyter.org/github/BlackHolePerturbationToolkit/kerrgeodesic_gw/blob/master/Notebooks/Kerr_geodesics.ipynb
- **Tolman-Oppenheimer-Volkoff equations** (derivation of TOV system and numerical integration): https://nbviewer.org/github/sagemanifolds/SageManifolds/blob/master/Worksheets/v1.3/SM_TOV.ipynb

Animated view of horizons and ergosurfaces in Kerr spacetime



The notebook:
https://nbviewer.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_Kerr_surfaces.ipynb

The animated view:
https://sagemanifolds.obspm.fr/images/animated/Kerr_surfaces.html

Image of an accretion disk surrounding a Schwarzschild BH

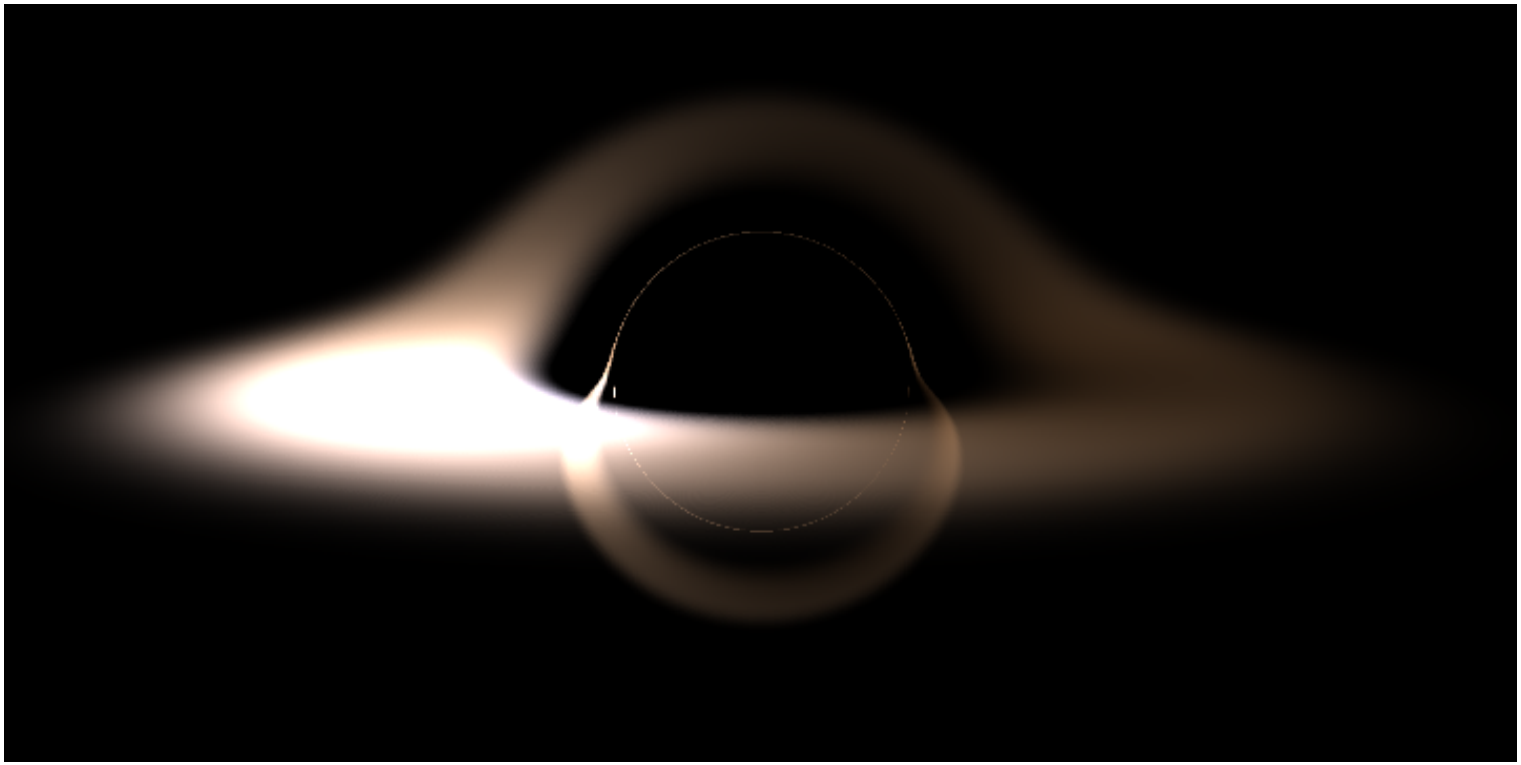


Image computed with SageMath by integrating null geodesics, cf. the notebook
https://nbviewer.jupyter.org/github/sagemanifolds/SageManifolds/blob/master/Notebooks/SM_black_hole_rendering.ipynb

Outline

Conclusions

Symbolic calculus on manifolds within the free Python-based system SageMath

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

Conclusions

Symbolic calculus on manifolds within 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 calculus on manifolds within 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 on parallelizable and non-parallelizable manifolds

Conclusions

Symbolic calculus on manifolds within 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 on parallelizable and 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 calculus on manifolds within 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 on parallelizable and 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 calculus on manifolds within 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 on parallelizable and 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 examples available at
https://sagemanifolds.obspm.fr/examples.html

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

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