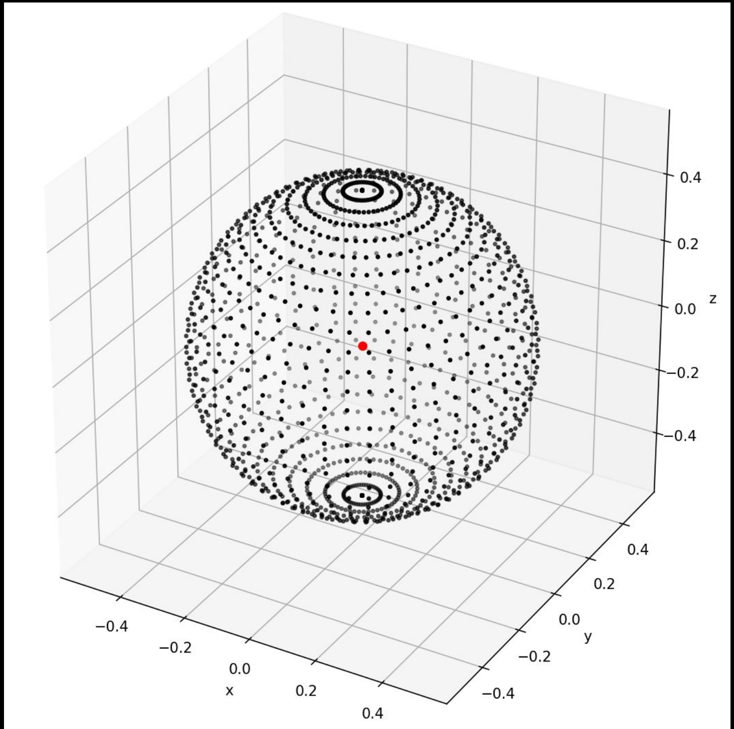
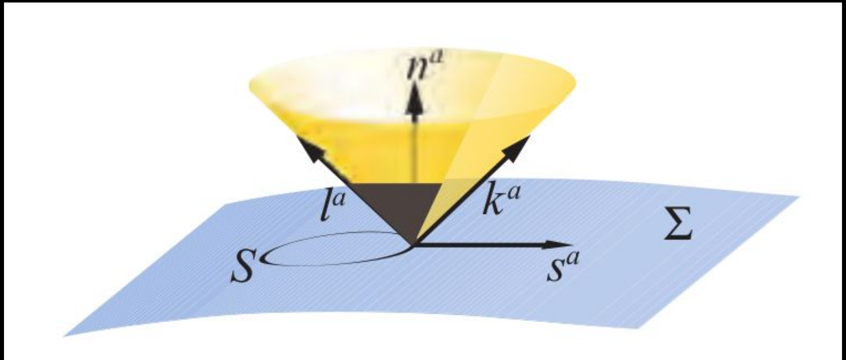
Apparent Horizon Finder



Tiago França, Queen Mary University of London

Apparent Horizon Finding
Intro

Expansion 101



Finding an apparent horizon (AH) corresponds to finding the outermost 2D trapped surface, where the "expansion", Θ, is zero. This is the surface where the area of a spherical flash of light rays emitted radially outwards will remain constant.

$$m_{\mu\nu} = \gamma_{\mu\nu} - s_\mu s_\nu, \qquad k_a = \frac{1}{\sqrt{2}}(n^a + s^a)$$

$$\Theta = m^{\mu\nu}\nabla_\mu k_\nu \;=\; \text{``}\frac{1}{\text{Area}} k^\mu \nabla_\mu \text{Area''} = 0$$

Image from T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equation on the Computer*

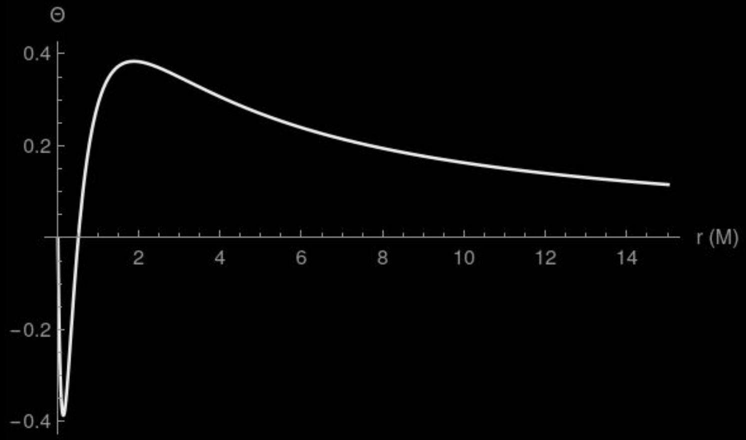
Example for Schwarzschild

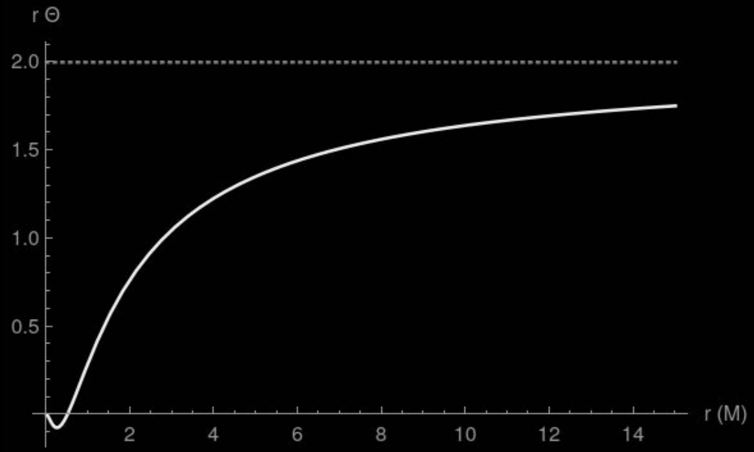
Pick a geometry and a reference surface, get the expansion.

$$1)\ dl^2 = \psi(r)^4\left(dr^2 + r^2 d\Omega^2\right)\ ,\quad \psi = 1 + \frac{M}{2r}$$

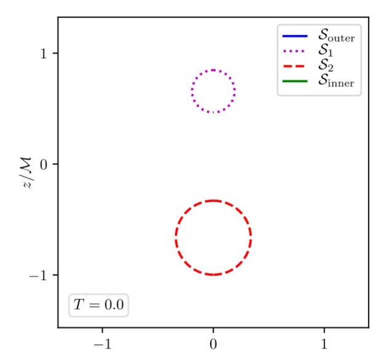
$$2)\ \text{Choose } s^i = \left(\frac{1}{\psi^2}, 0, 0\right)$$

$$\Longrightarrow\ \Theta = \frac{2}{r}\frac{1-\frac{M}{2r}}{\left(1+\frac{M}{2r}\right)^3}\ \overset{\Theta=0}{\Longrightarrow}\ r = \frac{M}{2}$$



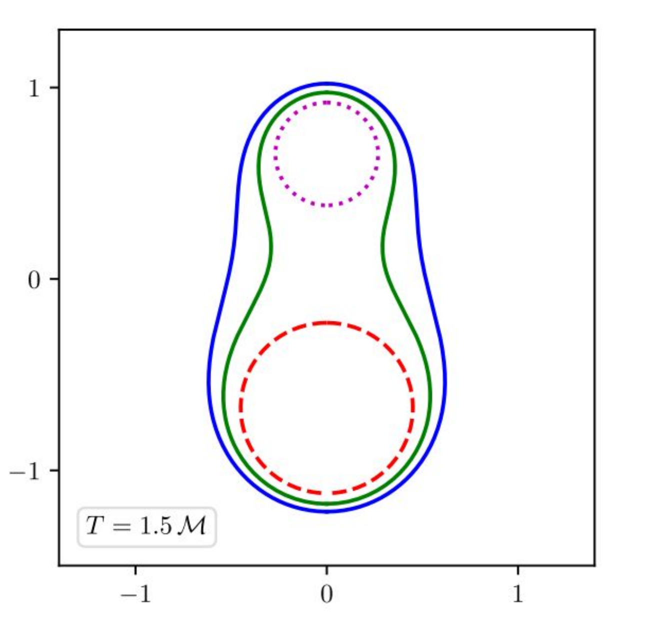


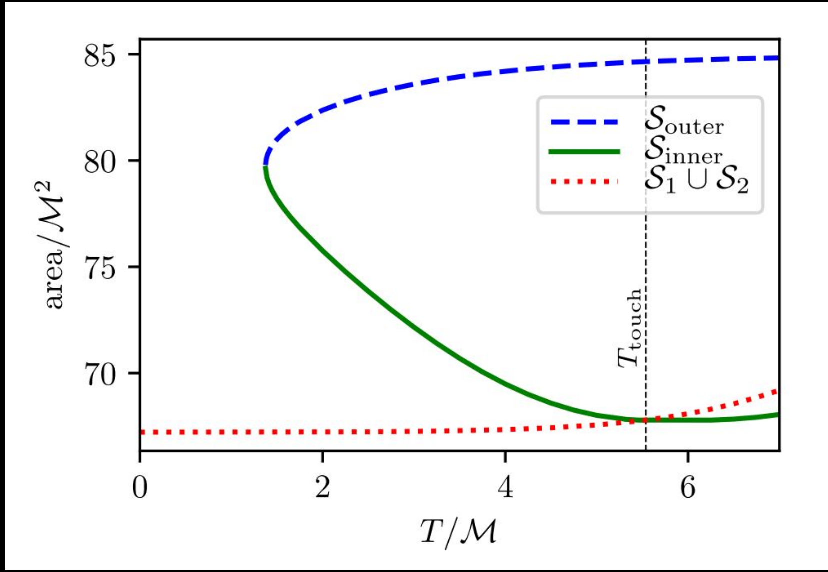
Even though we're solving for zero Expansion, multiplying by a power of the radius has better convergence properties.



In binaries, when merging, a 3rd and 4th trapped surface appear suddenly enclosing the smaller 2. The outermost one is the merged horizon. Hence, we need 3 AH for binary simulations.

(see details in 1907.00683)





Numerical Methods for Star-shaped AHs

GRchombo's AHFinder discretizes the 2D AH surface and uses a quasi-Newton method from the PETSc library to find the zero of the expansion for an ansatz of a star-shaped horizon, given some initial guess (currently set to spherical).

Newton's Method (1D):

$$f_{n+1} = f_n - \gamma \left(\Theta'\left(f_n\right)\right)^{-1} \Theta\left(f_n\right), \quad 0 < \gamma \leq 1$$

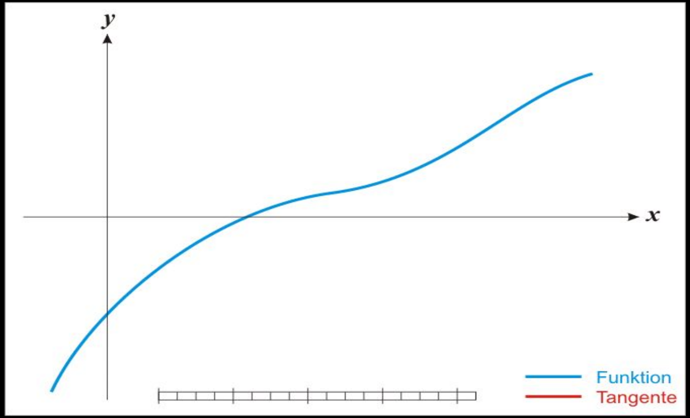
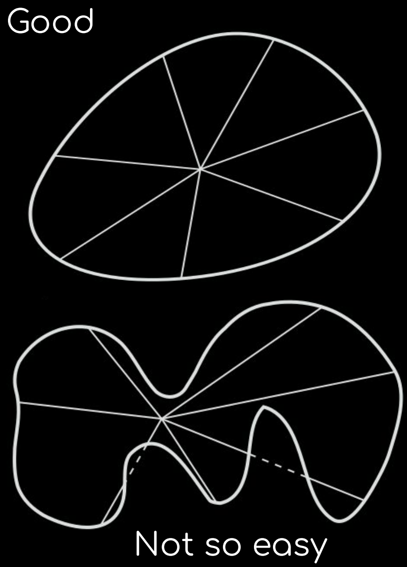
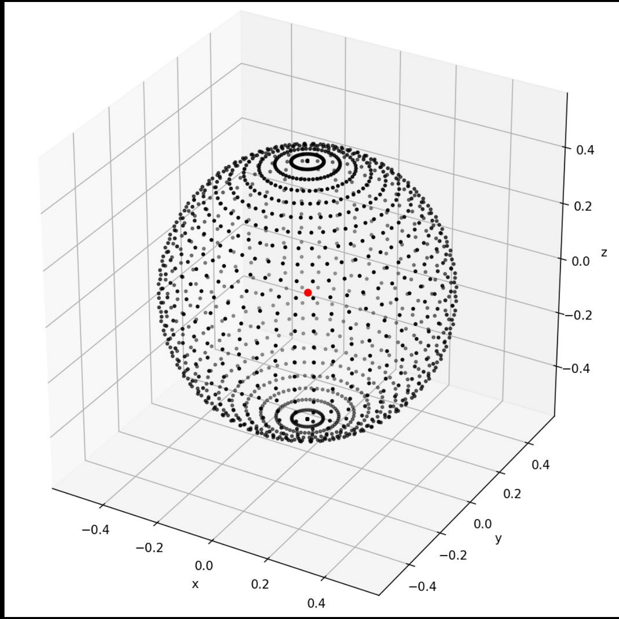


Image from Wikipedia



Images from M. Alcubierre, *Introduction to 3+1 Numerical Relativity*

Numerical Methods for Star-shaped AHs



$$\text{Ansatz: } F(\theta,\phi,r) = r - f(\theta,\phi) = 0$$

$$s^i = \frac{D^i F}{|DF|}$$

$$\text{PETSc:} \begin{cases} \text{Mix non-linear Newton-Method (SNES):} \\ \vec{f}_{n+1} = \vec{f}_n - \gamma \Delta \vec{f} \\ \Delta \vec{f} = \left(\nabla_{\vec{f}} \vec{\Theta}\left(\vec{f}_n\right)\right)^{-1} \vec{\Theta}\left(\vec{f}_n\right) \\ \text{With linear system solver (KSP):} \\ \left(\nabla_{\vec{f}} \vec{\Theta}\left(\vec{f}_n\right)\right) \Delta \vec{f} = \vec{\Theta}\left(\vec{f}_n\right) \end{cases}$$

$$\Rightarrow \text{provide } \Theta(\cdot), \nabla\Theta(\cdot), \vec{f}_0$$

AHFinder – How to use?

AHFinder Class

How to use?

1) Install PETSc and change Chombo's Make.defs.local

2) Change your Example

3) Add parameters to 'params.txt' file

4) What is the output?

Quick version: consult BinaryBH / KerrBH Examples as a reference

AHFinder – How to use

1) Install PETSc and change Chombo's Make.defs.local

- To install PESTc, it is preferable to use a cluster module. If building yourself, see below.
- Change Make.defs.local for 2D and PETSc compatibility and recompile Chombo.

Make.defs.local

```
DIM                   ?= 3
DEBUG                  = FALSE
...
cxxdbgflags            = -g
cxxoptflags            = -O3 -xCORE-AVX512
fdbgflags              = -g
foptflags              = -O3 -xCORE-AVX512
syslibflags            = -lpetsc
cxxcppflags           := ${cxxcppflags} -DUSE_AHFINDER
```

If building yourself, make sure in installation you configure it with:

- HDF5. Use '--with-hdf5-dir=' with directory as in your HDF FLAGS of Make.defs.local (without the 'include'/'lib').
- MPI (if your Chombo has MPI). Force no MPI with '--with-mpi=0' or force MPI compilers with '--with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpiifort'

AHFinder – How to use

2) Change your Example - Part I

Main_KerrBH.cpp

```
#include "BHAMR.hpp"

int runGRChombo(int argc, char *argv[])
```

...

```
    BHAMR bh_amr;
    DefaultLevelFactory<KerrBHLevel> kerr_bh_level_fact(bh_amr, sim_params);
    setupAMRObject(bh_amr, kerr_bh_level_fact);

    // Set up interpolator:
    // call this after amr object setup so grids known
    // and need it to stay in scope throughout run
    // Note: 'interpolator' needs to be in scope when bh_amr.run() is called,
    // otherwise pointer is lost
    AMRInterpolator<Lagrange<4>> interpolator(
        bh_amr, sim_params.origin, sim_params.dx, sim_params.boundary_params,
        sim_params.verbosity);
    bh_amr.set_interpolator(&interpolator);

#ifdef USE_AHFINDER
    if (sim_params.AH_activate)
    {
        AHSphericalGeometry sph(sim_params.kerr_params.center);
        bh_amr.m_ah_finder.add_ah(sph, sim_params.AH_initial_guess,
                                  sim_params.AH_params);
    }
#endif
```

- Main_Example.cpp: Change GRAMR to BHAMR (which provides AHFinder and PunctureTracker)
- Main_Example.cpp: Add an AMRInterpolator if you don’t have one yet
- Main_Example.cpp: Add an AH with a given geometry

AHFinder – How to use

2) Change your Example - Part II

SimulationParameters.hpp

```
        pp.load("kerr_center", kerr_params.center, center);

#ifdef USE_AHFINDER
        pp.load("AH_initial_guess", AH_initial_guess, 0.5 * kerr_params.mass);
#endif
    }

    KerrBH::params_t kerr_params;

#ifdef USE_AHFINDER
    double AH_initial_guess;
#endif
};
```

GNUmakefile

```
src_dirs := $(GRCHOMBO_SOURCE)/utils \
            $(GRCHOMBO_SOURCE)/simd  \
            $(GRCHOMBO_SOURCE)/CCZ4  \
            $(GRCHOMBO_SOURCE)/BoxUtils  \
            $(GRCHOMBO_SOURCE)/GRChomboCore  \
            $(GRCHOMBO_SOURCE)/TaggingCriteria  \
            $(GRCHOMBO_SOURCE)/InitialConditions/BlackHoles \
            $(GRCHOMBO_SOURCE)/BlackHoles \
            $(GRCHOMBO_SOURCE)/AMRInterpolator  \
            $(GRCHOMBO_SOURCE)/ApparentHorizonFinder

include $(CHOMBO_HOME)/mk/Make.test
```

- SimulationParameters.hpp: Add an initial guess parameter
- GNUmakefile: Add the BlackHoles, AMRInterpolator and ApparentHorizonFinder Source folders

AHFinder – How to use

2) Change your Example - Part III

KerrBHLevel.hpp

```cpp
#include "BHAMR.hpp"
#include "DefaultLevelFactory.hpp"
#include "GRAMRLevel.hpp"

class KerrBHLevel : public GRAMRLevel
{
    friend class DefaultLevelFactory<KerrBHLevel>;
    // Inherit the contructors from GRAMRLevel
    using GRAMRLevel::GRAMRLevel;

    BHAMR &m_bh_amr = dynamic_cast<BHAMR &>(m_gr_amr);
```

- ExampleLevel.hpp: include BHAMR and create a 'm_bh_amr' member
- ExampleLevel.cpp: add 'solve' to 'specificPostTimeStep' (you may have to add it to the .hpp if you don't have it already)

KerrBHLevel.cpp

```cpp
void KerrBHLevel::specificPostTimeStep()
{
    CH_TIME("KerrBHLevel::specificPostTimeStep");
#ifdef USE_AHFINDER
    if (m_p.AH_activate && m_level == m_p.AH_params.level_to_run)
        m_bh_amr.m_ah_finder.solve(m_dt, m_time, m_restart_time);
#endif
}
```

AHFinder – How to use

2) Change your Example - Part IV - Binaries

Main_BinaryBH.cpp

```cpp
    bh_amr.set_interpolator(
        &interpolator); // also sets puncture_tracker interpolator

#ifdef USE_AHFINDER
    if (sim_params.AH_activate)
    {
        AHSphericalGeometry sph1(sim_params.bh1_params.center);
        AHSphericalGeometry sph2(sim_params.bh2_params.center);

        bh_amr.m_ah_finder.add_ah(sph1, sim_params.AH_1_initial_guess,
                                  sim_params.AH_params);
        bh_amr.m_ah_finder.add_ah(sph2, sim_params.AH_2_initial_guess,
                                  sim_params.AH_params);
        bh_amr.m_ah_finder.add_ah_merger(0, 1, sim_params.AH_params);
    }
#endif
```

- Main_Example.cpp: add as many AHs as you want, including mergers. Only initial guesses need to be added (except for mergers, for which it is automatic). AHFinder::solve manages solving all the AHs.

AHFinder – How to use

3) Add parameters to 'params.txt' file

params.txt

```
#Apparent Horizon finder
AH_activate = 1
AH_num_ranks = 65
AH_num_points_u = 65
AH_num_points_v = 48
#AH_solve_interval = 1
#AH_print_interval = 1
#AH_track_center = true
#AH_predict_origin = true
#AH_level_to_run = 0
#AH_start_time = 0.
#AH_give_up_time = -1.
```

...

```
#AH_merger_search_factor = 1.
#AH_merger_pre_factor = 1.
#AH_allow_re_attempt = 0
#AH_max_fails_after_lost = -1
#AH_verbose = 1
#AH_print_geometry_data = 0
#AH_re_solve_at_restart = 0
#AH_stop_if_max_fails = 0

#AH_1_initial_guess = 0.3
#AH_2_initial_guess = 0.3

#AH_num_extra_vars = 2
#AH_extra_vars = chi d1_Ham d2_A11

AH_set_origins_to_punctures = 1
```

- params.txt: there are many AH parameters. The commented values are the default values. Consult AHFinder.hpp for more information (meaning and default values).

AHFinder – How to use

4) What is the output? Part I - Command Line

```
BinaryBH_000172.3d.hdf5  coords_AH2_0022.out  coords_AH3_0071.out  coords_AH3_0131.out
BinaryBH_000176.3d.hdf5  coords_AH2_0023.out  coords_AH3_0072.out  coords_AH3_0132.out
BinaryBH_000180.3d.hdf5  coords_AH2_0024.out  coords_AH3_0073.out  coords_AH3_0133.out
BinaryBH_000184.3d.hdf5  coords_AH2_0025.out  coords_AH3_0074.out  coords_AH3_0134.out
BinaryBH_000188.3d.hdf5  coords_AH2_0026.out  coords_AH3_0075.out  coords_AH3_0135.out
BinaryBH_000192.3d.hdf5  coords_AH2_0027.out  coords_AH3_0076.out  coords_AH3_0136.out
BinaryBH_000196.3d.hdf5  coords_AH2_0028.out  coords_AH3_0077.out  coords_AH3_0137.out
BinaryBH_000200.3d.hdf5  coords_AH2_0029.out  coords_AH3_0078.out  coords_AH3_0138.out
coords_AH1_0000.out      coords_AH3_0019.out  coords_AH3_0079.out  coords_AH3_0139.out
coords_AH1_0001.out      coords_AH3_0020.out  coords_AH3_0080.out  coords_AH3_0140.out
coords_AH1_0002.out      coords_AH3_0021.out  coords_AH3_0081.out  coords_AH3_0141.out
coords_AH1_0003.out      coords_AH3_0022.out  coords_AH3_0082.out  coords_AH3_0142.out
coords_AH1_0004.out      coords_AH3_0023.out  coords_AH3_0083.out  coords_AH3_0143.out
coords_AH1_0005.out      coords_AH3_0024.out  coords_AH3_0084.out  coords_AH3_0144.out
coords_AH1_0006.out      coords_AH3_0025.out  coords_AH3_0085.out  coords_AH3_0145.out
coords_AH1_0007.out      coords_AH3_0026.out  coords_AH3_0086.out  coords_AH3_0146.out
coords_AH1_0008.out      coords_AH3_0027.out  coords_AH3_0087.out  coords_AH3_0147.out
```

```
pout.88
pout.89
pout.9
pout.90
pout.91
pout.92
pout.93
pout.94
pout.95
pout.96
pout.97
pout.98
pout.99
slurm-17662242.out
stats_AH1.out
stats_AH2.out
stats_AH3.out
```

- Output will be a ‘coords’ file for each AH and for each step, containing to coordinates of the AH surface, and a ‘stats’ file for each AH, containing convergence information (e.g. area and spin) for all timesteps.

AHFinder – How to use

4) What is the output? Part II - ‘pout’ files

pout.0

```
GRAMRLevel::advance level 3 at time 2.23438 (22.2691 M/hr). Boxes on this rank: 8 / 28
Solving AH #0
SNES Iteration number 6
KSP Iteration number 78
Solver converged. Horizon FOUND.
SNESConvergedReason = 4
center: (6.1905, 8.00001, -1.46586e-09)
area = 16.9617
mass = 0.580898
spin = 1.77674e-05
irreducible mass = 0.580898
dimensionless spin vector = (1.7253e-07, -2.92756e-05, 8.85515e-06)
dimensionless spin in z (from equator-length integral) = 0
Solving AH #1
SNES Iteration number 6
KSP Iteration number 78
Solver converged. Horizon FOUND.
SNESConvergedReason = 4
center: (9.8095, 8.00001, -1.46832e-09)
area = 16.9627
mass = 0.580915
spin = 1.74909e-05
irreducible mass = 0.580915
dimensionless spin vector = (1.82558e-07, 2.90686e-05, -7.84557e-06)
dimensionless spin in z (from equator-length integral) = 0
BHs #0 and #1 at distance = 3.61901 > minimum distance = 2. Skipping solve for merger...
GRAMRLevel::advance level 0 at time 2.25 (22.1623 M/hr). Boxes on this rank: 1 / 1
```

- ‘pout’ files: print various information when solving for the AHs. Control with the ‘AH_verbose’ verbosity level (ranges between 0-3).

- The spin is calculated in 2 ways: one to calculate the full spin 3-vector and another method to gets the spin oriented with ‘z’. The later is more precise, and the former is only useful if you need to know the direction of the spin.

AHFinder – How to use

4) What is the output? Part III - 'coords' files

coords_AH1_000000.dat ×

# theta	phi	r	chi	dx_chi	dy_chi	dz_chi
0.0000000000	0.0000000000	0.4839762884	5.5056177192e-02	7.1150767569e-18	7.1015242298e-18	2.2786411876e-01
0.3141592654	0.0000000000	0.4839941742	5.5085085743e-02	7.0672731933e-02	-8.8091426514e-20	2.1664488191e-01
0.6283185307	0.0000000000	0.4840507182	5.5149123468e-02	1.3438728357e-01	6.6136332522e-18	1.8441576914e-01
0.9424777961	0.0000000000	0.4841351993	5.5221840971e-02	1.8513333988e-01	2.6969529041e-18	1.3407117896e-01
1.2566370614	0.0000000000	0.4842168287	5.5275258124e-02	2.1776052981e-01	4.6349642874e-18	7.0594833460e-02
1.5707963268	0.0000000000	0.4842598993	5.5293492230e-02	2.2914994366e-01	3.8556939759e-18	-2.4070510568e-18
1.8849555922	0.0000000000	0.4842168289	5.5275258178e-02	2.1776052986e-01	2.7240579584e-18	-7.0594833478e-02
2.1991148575	0.0000000000	0.4841351998	5.5221841092e-02	1.8513333999e-01	5.1228552650e-18	-1.3407117903e-01
2.5132741229	0.0000000000	0.4840507191	5.5149123687e-02	1.3438728371e-01	6.3696877634e-18	-1.8441576933e-01
2.8274333882	0.0000000000	0.4839941758	5.5085086123e-02	7.0672732060e-02	5.6785088784e-18	-2.1664488230e-01
3.1415926536	0.0000000000	0.4839762899	5.5056177544e-02	5.2177229551e-18	5.2041704279e-18	-2.2786411914e-01
0.0000000000	0.3141592654	0.4859051601	5.5495880920e-02	5.3803532810e-18	5.3668007538e-18	2.2833166757e-01
0.3141592654	0.3141592654	0.4870958848	5.5783990070e-02	6.7444169396e-02	2.1909965817e-02	2.1735972065e-01

- 'coords' files: contain the coordinate system information about the surface of the AH (spherical coordinates - theta, phi, r - above).
  With the parameters 'AH_num_write_vars = 2' and 'AH_write_vars = chi d1_chi' the example above also outputs the value of 'chi' and its derivatives at each point of the horizon. These can be diagnostic variables and include 1st or 2nd derivatives.

AHFinder – How to use

4) What is the output? Part III - stats_AH1.dat

head stats_AH1.dat

# time	file	area	mass	irreducible mass	spin	dimless spin-x	dimless spin-y
2.0000000000	4.0000000000e+00	1.6418300976e+01	5.7151703384e-01	5.7151703359e-01	3.3947220719e-05	1.0990679865e-08	-5.9239748341e-05
2.5000000000	5.0000000000e+00	1.7487064429e+01	5.8982547674e-01	5.8982547673e-01	6.7097940928e-06	-5.1749065646e-07	-1.0211252952e-05
3.0000000000	6.0000000000e+00	1.8423481166e+01	6.0541185337e-01	6.0541185326e-01	2.2251144079e-05	4.3507303577e-07	-3.5360014956e-05
3.5000000000	7.0000000000e+00	1.9195748854e+01	6.1797029939e-01	6.1797029828e-01	7.4045531728e-05	1.3623533919e-07	-1.1964564076e-04

...

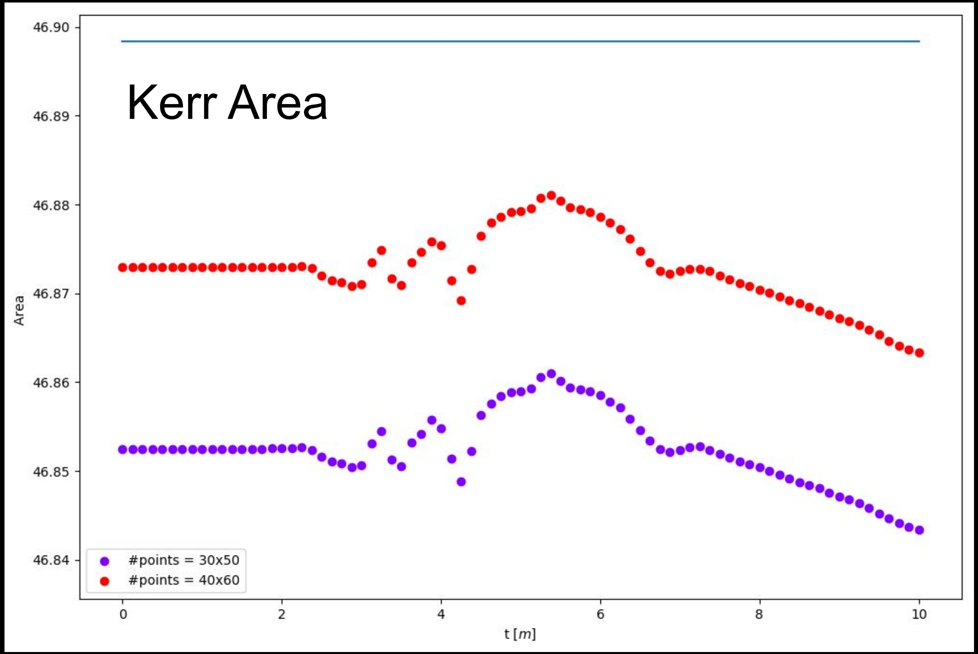
...

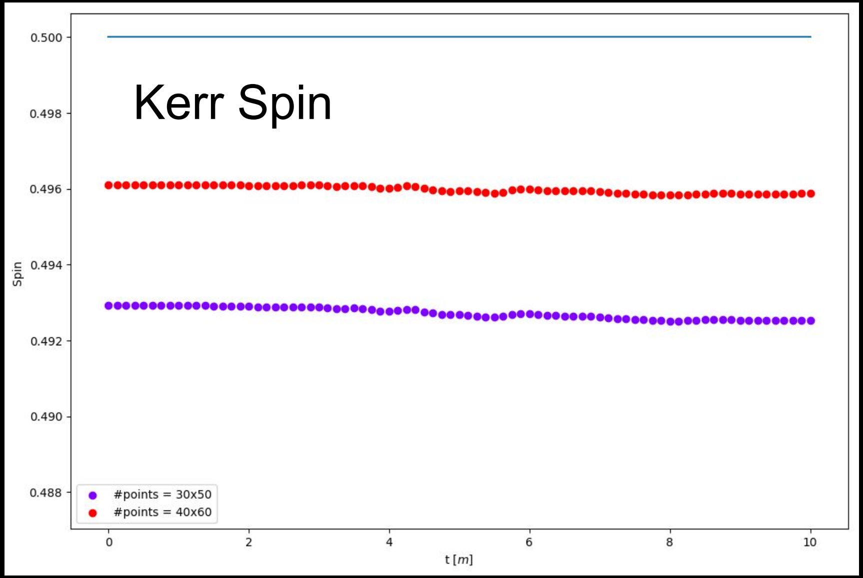
dimless spin-z	dimless spin-z-alt	origin_x	origin_y	origin_z	center_x	center_y	center_z
4.3389940102e-06	0.0000000000e+00	6.1561971694e+00	7.9999998311e+00	6.2348738671e-23	6.1468078594e+00	8.0000049371e+00	0.0000000000e+00
-4.9873397381e-06	0.0000000000e+00	6.2283372624e+00	7.9999997328e+00	1.0867085297e-22	6.2343423711e+00	7.9999910394e+00	0.0000000000e+00
-1.0015825320e-05	0.0000000000e+00	6.3160466820e+00	7.9999994787e+00	7.1447725409e-23	6.3162350431e+00	7.9999853858e+00	0.0000000000e+00
6.4701339056e-06	0.0000000000e+00	6.4135201723e+00	7.9999991857e+00	9.7917505011e-23	6.3855359687e+00	8.0000071376e+00	0.0000000000e+00

- 'stats' files: print the area and (dimensionless) spin of each found AH at each timestep (printing every 'AH_print_interval * AH_solve_interval'). Again, 'dimless spin-z-alt' is a more precise calculation if aligned with 'z', and spix-x,y,z gives the 3-vector direction.
- 'origin' is the origin of the coordinate system used in the 'coords' file (the starting point for the solver, which might not coincide with the actual center of the AH).
- 'center' is an approximate geometric center of the surface found (an approximate location of the puncture).
- 'file' is the 'coords_AH#_file.dat' file containing the coordinates of this step. This allows the user to change parameters that affect AH printing frequency (as 'AH_level_to_run', 'AH_solve_interval', etc.) without losing track of 'coords' file numbering.

AHFinder – How to use

4) What is the output? Plot using python or gnuplot

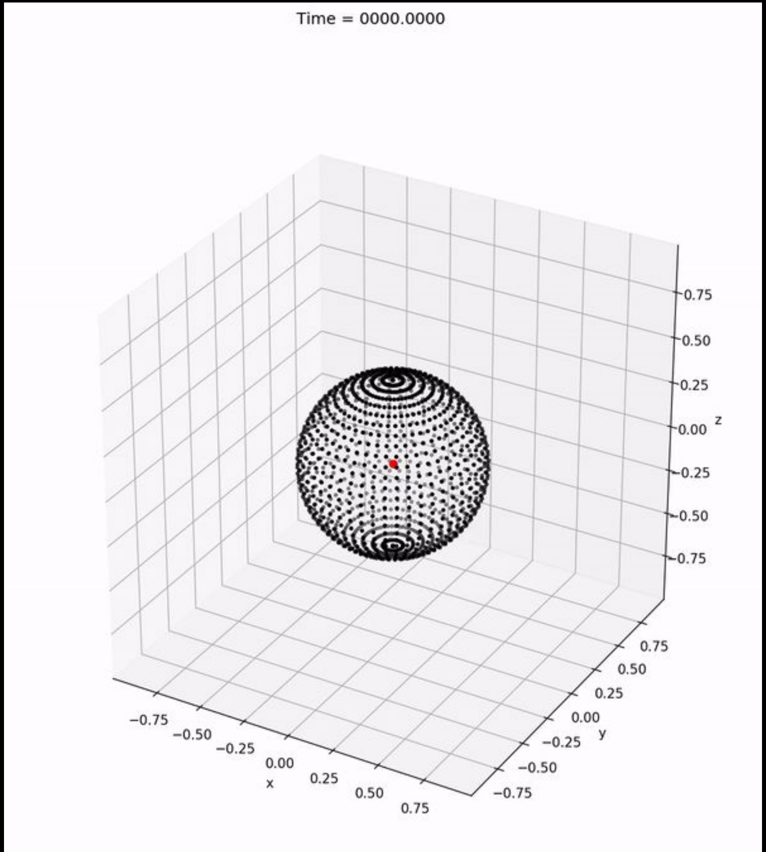




The straight line is the expected value. Increasing the number of points in the AH brings us closer to it (provided the numerical grid has enough resolution).

AHFinder – How to use

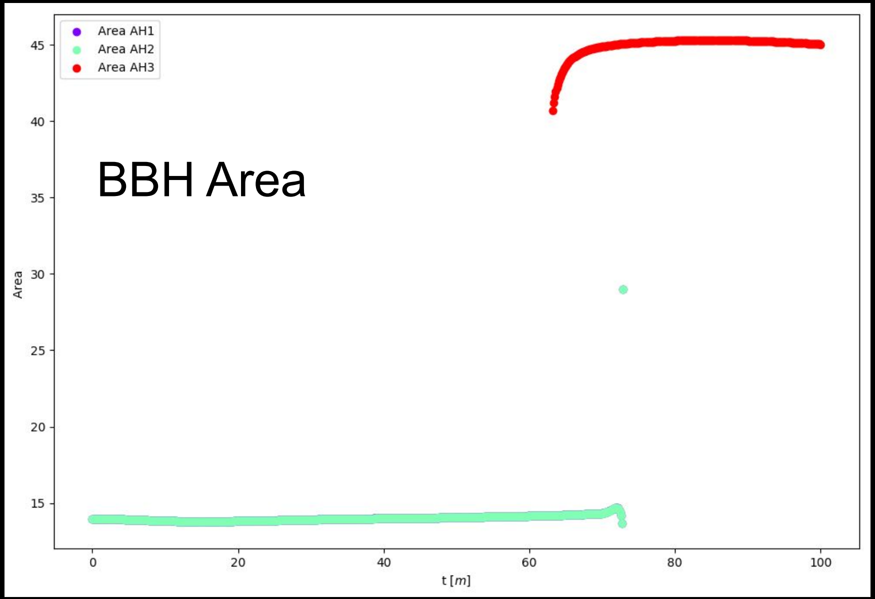
4) What is the output? Plot using python or gnuplot

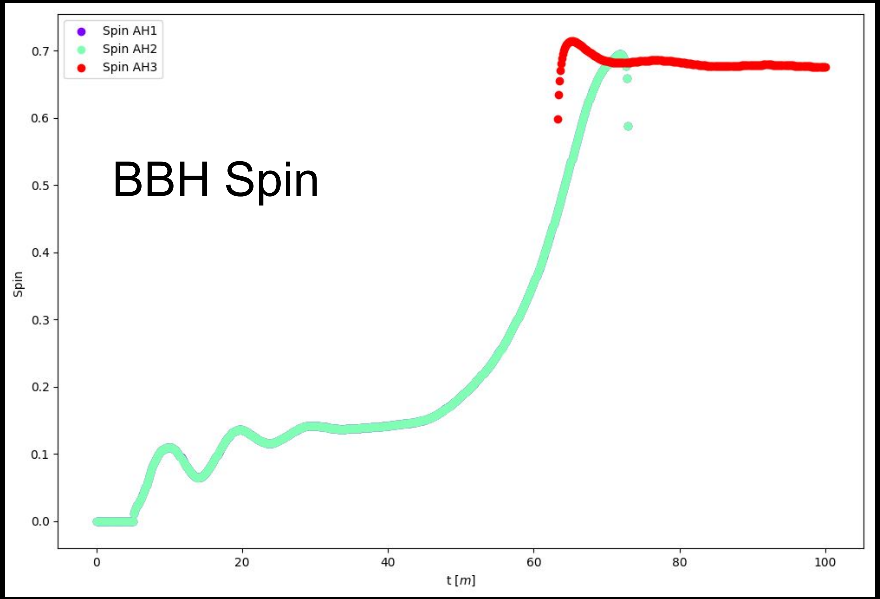


Python script example for plotting in 'Postprocessing_tools' repository.

AHFinder – How to use

4) What is the output? Similarly for BinaryBH case



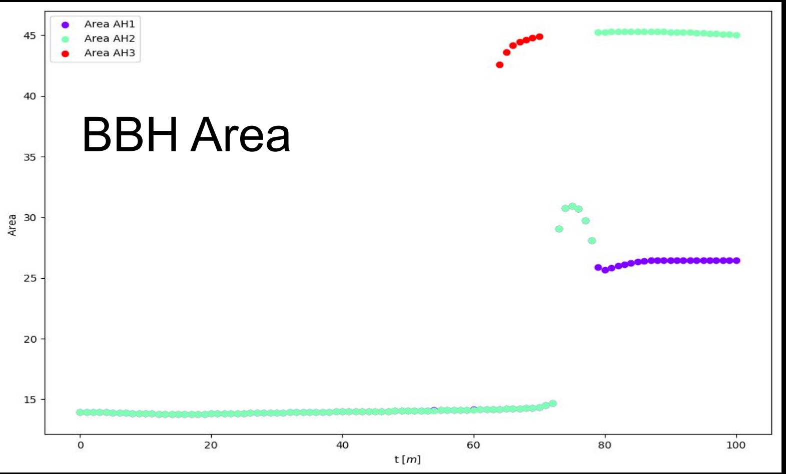


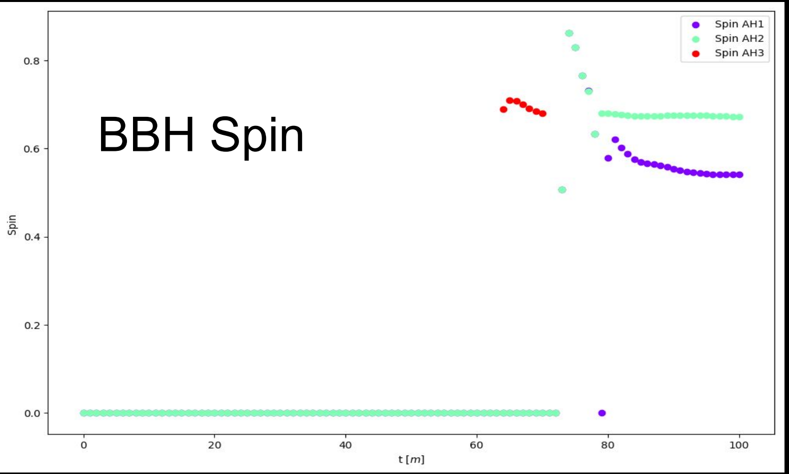
AHFinder – How to use

4) What is the output? It doesn't always go right…

In the case below, AH1 converged to the inner merged trapped surface (see slide 4), AH2 converged to the merged horizon (that should be AH3) and AH3 stopped converging. Other problems can appear as well, so pay attention to the merger stage.

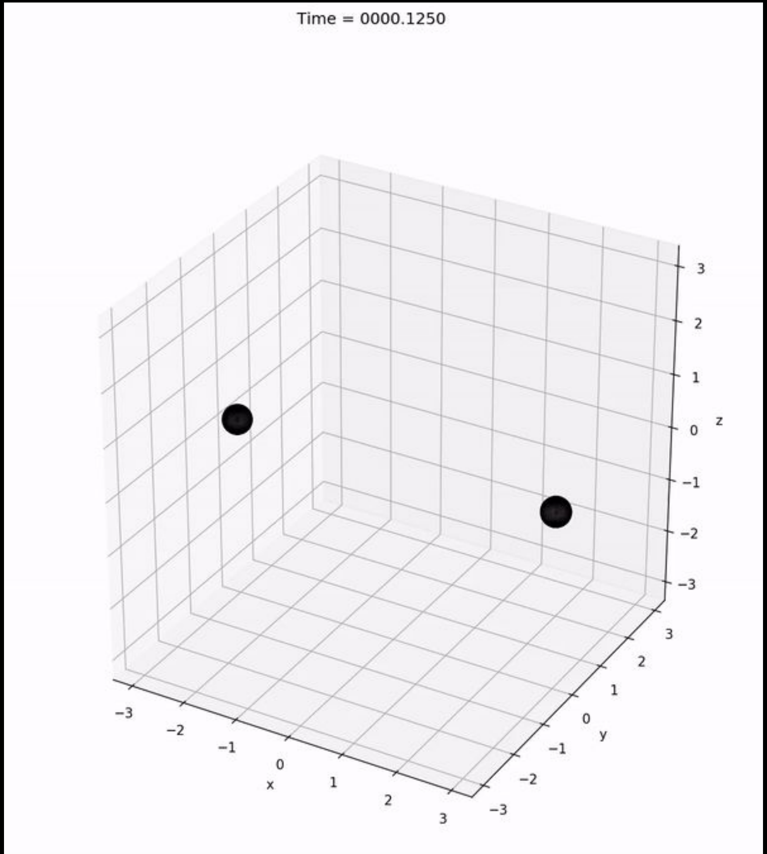
The best solution is to change the initial guess for the merger (using 'AH_merger_pre_factor'), or by changing the frequency of solving to make sure AH3 stays stable. When AH3 is stable, it's fine for AH1 and AH2 to stop converging (they should!). If they don't, one can force them to stop by deleting the last entry of their 'stats' file and restarting.





AHFinder – How to use

4) What is the output? When it goes right…



AHFinder – Extra notes

AHFinder – Advanced notes

- Finding other surfaces: the AHFinder finds a level surface of a given function and is independent of physics. The default function is 'expansion = 0', but you can define 'AHFunction' to be a new class following 'AHFunctionDefault.hpp'. See KerrBH Example to see how to use the AHFinder to look for 'chi' contours instead of the AH.

- 'Postprocessing_tools' repo: contains in ChomboTools an Example of how to run the AHFinder on a set of HDF5's and a python script for plotting the AH and related quantities.

- Writing Diagnostics on AH: if you use diagnostics in 'AH_write_vars', make sure those diagnostics are computed at 'specificPostTimeStep'. AHFinder has a 'need_diagnostics' method that determines if any "writing variable" is diagnostic. See KerrBH Example.

AHFinder – Advanced notes

- New Geometry (3D or 2D): the AHFinder is compatible with 2D and 3D codes, just write a new geometry class following the existing ('AHSphericalGeometry' for 3D and 'AHStringGeometry' for 2D) and define 'AHSurfaceGeometry' to be your new geometry class before including 'BHAMR.hpp'. For Cartoon methods, some terms in the Expansion calculation may have to be adapted for each problem (see 'AHFunctions.hpp').

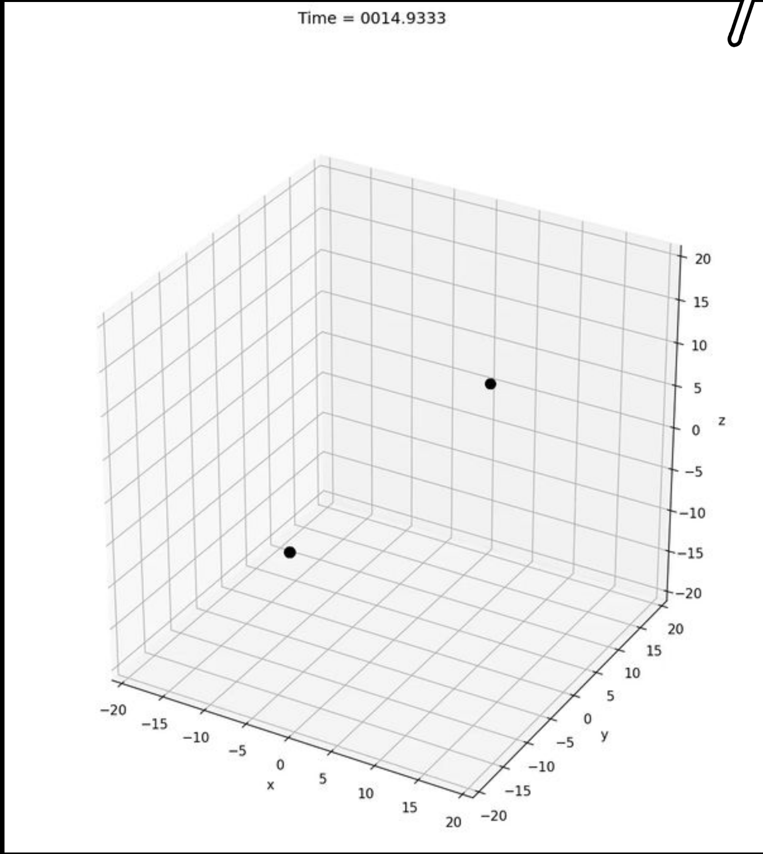
- Interpolating data: see the 'InterpolatorTest' for how to use the existing Interpolating classes to interpolate your own 1D, 2D or 3D data.

AHFinder – Examples

AHFinder – Examples

Circular and eccentric Binary Black Hole

Missing circular -
to add



AHFinder – Examples

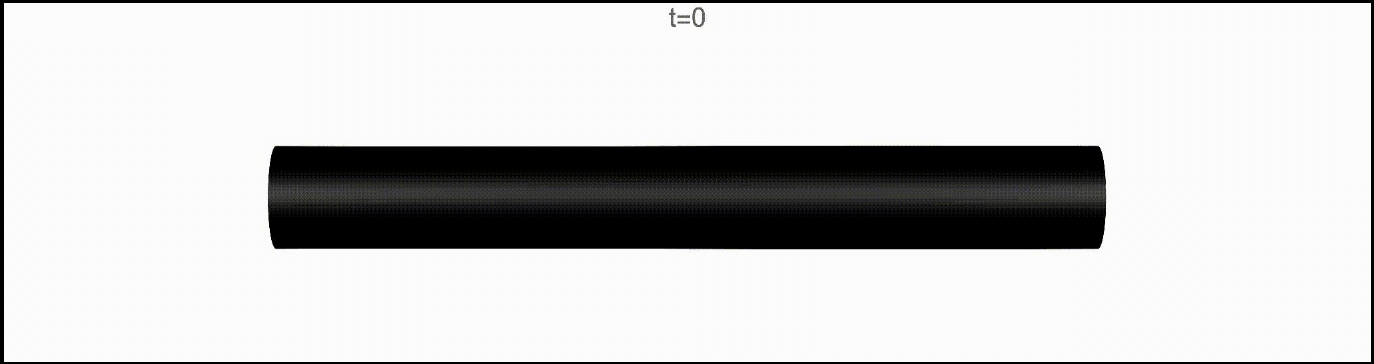
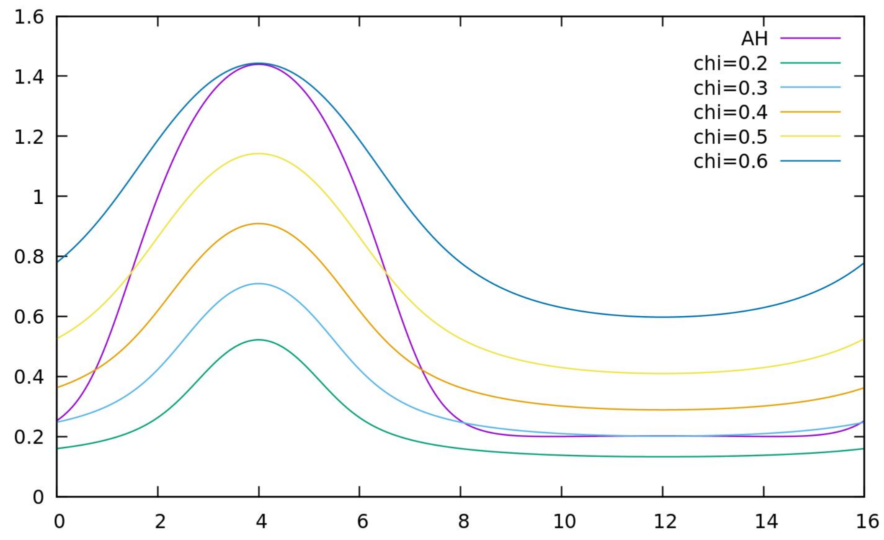


Image credit to Chenxia Gu

Example of AHFinder working in 2D (BlackString 4+1 reduced to a 2+1 simulation by Cartoon method), with non-spherical geometry (String Geometry) and being used to find both the AH and chi contours (see chi contours on the right)



AH analysis

AH analysis

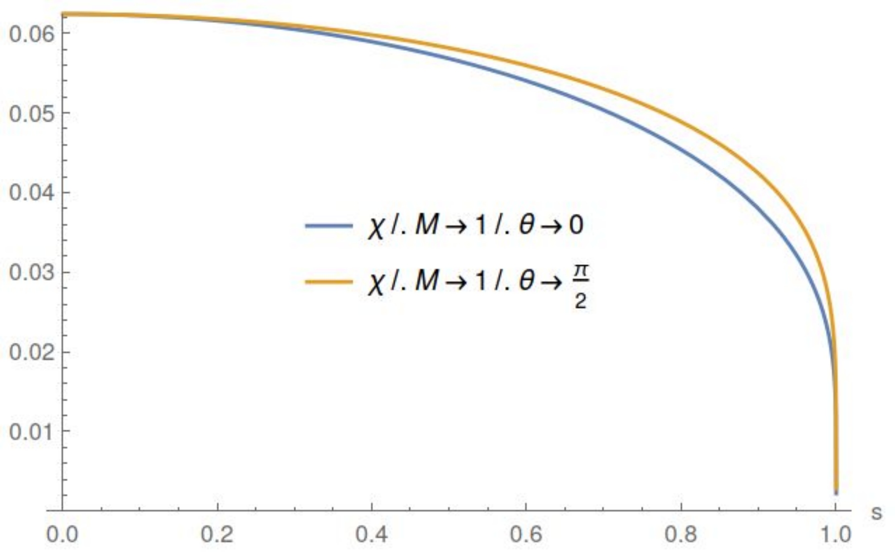
How does the AH behave in Kerr isotropic initial data?

For a BH of mass ‘M’ and dimensionless spin ‘s’ ranging from [-1,1] (s=a/M, ‘a’ being the spin of our current KerrBH initial data):

$$r_{AH} == \frac{1}{4} M \left(1 + \sqrt{1 - s^2}\right)$$

$$\chi_{AH,\theta=0} == \frac{\left(1 - s^2\right)^{1/6} \left(\frac{1+\sqrt{1-s^2}}{2}\right)^{2/3}}{16}$$

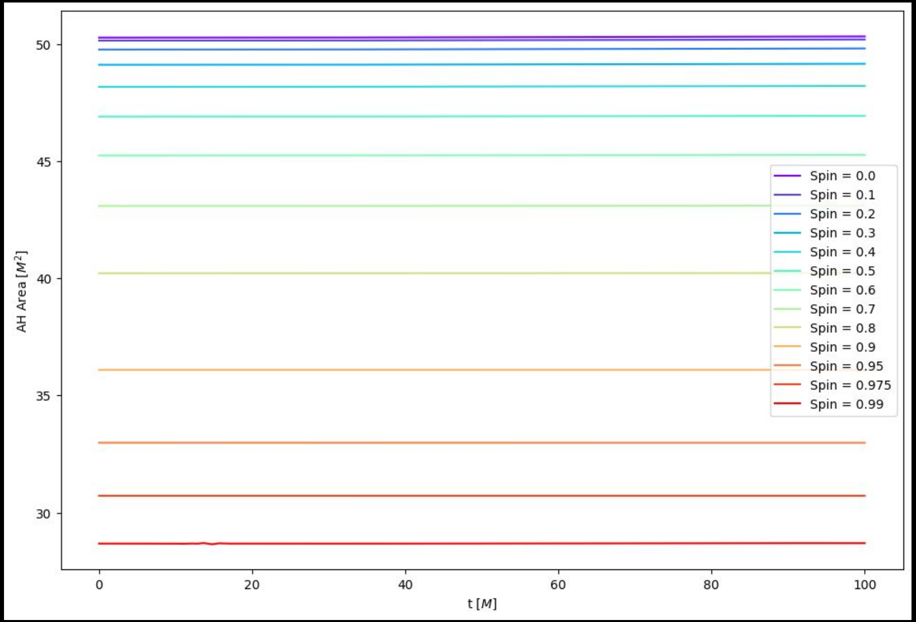
$$\chi_{AH,\theta=\pi/2} == \frac{\left(1 - s^2\right)^{1/6} \left(\frac{1+\sqrt{1-s^2}}{2}\right)^{1/3}}{16}$$

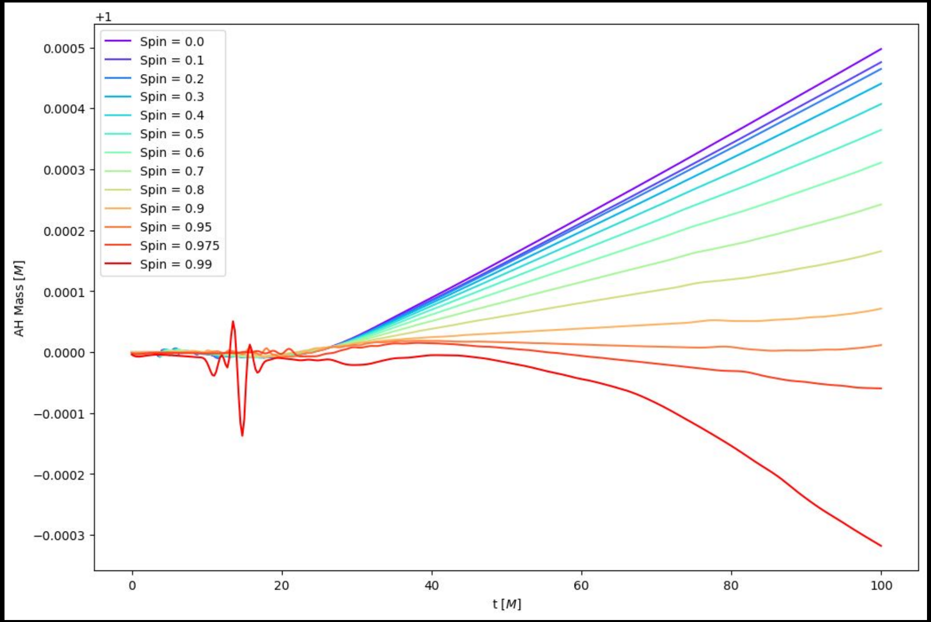


AH analysis

How does the AH behave in puncture gauge?

Kerr BH simulations with M=1 and spin from s=0 to s=0.99.
The area and mass:





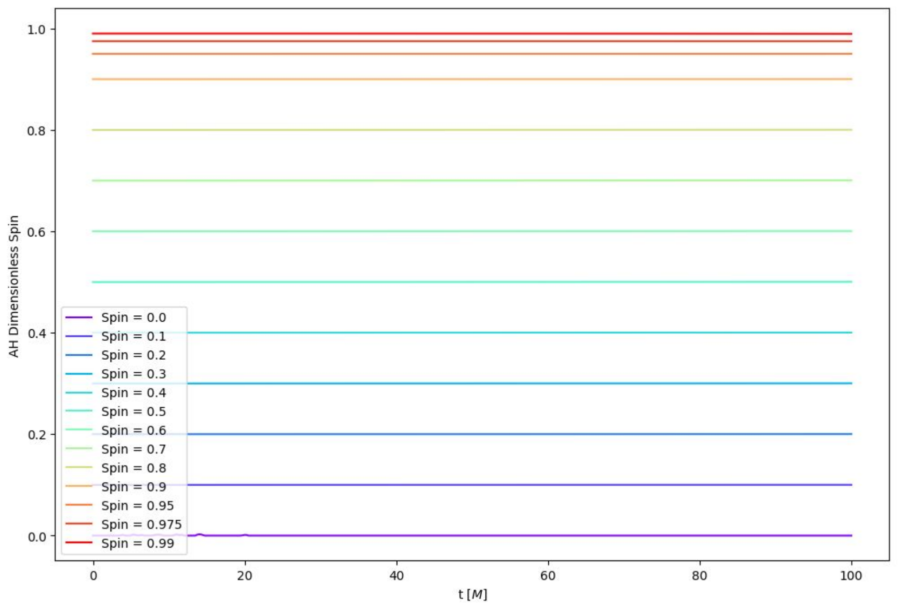
The mass slightly drifts over time. This is a known issue for when the resolution is not good enough OR when the finest level is too close to the AH.

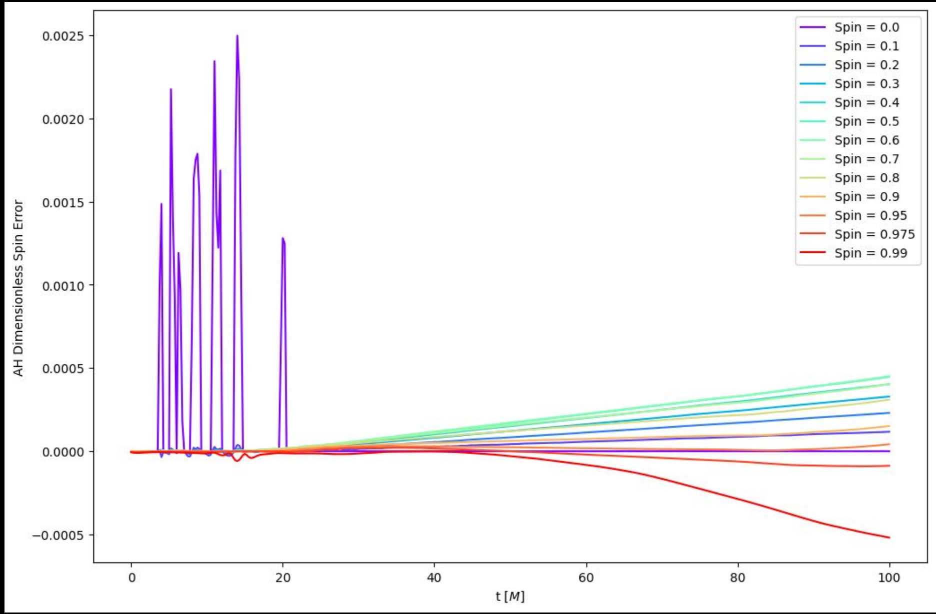
(KerrBH simulation with L=N=128, max_level=6, regrid_threshold=0.0065, regrid_interval=0, CCZ4 formulation)

AH analysis

How does the AH behave in puncture gauge?

The spin:



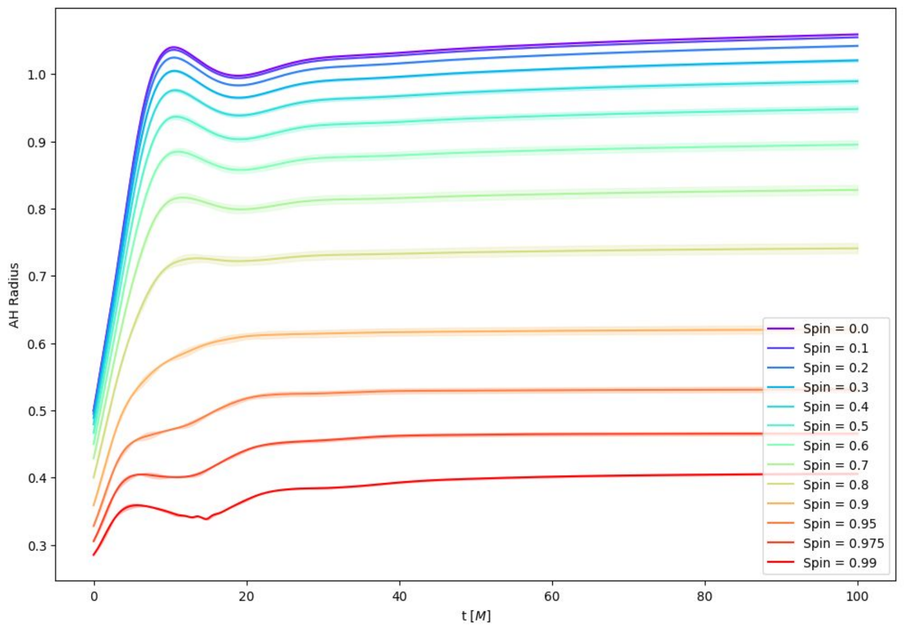


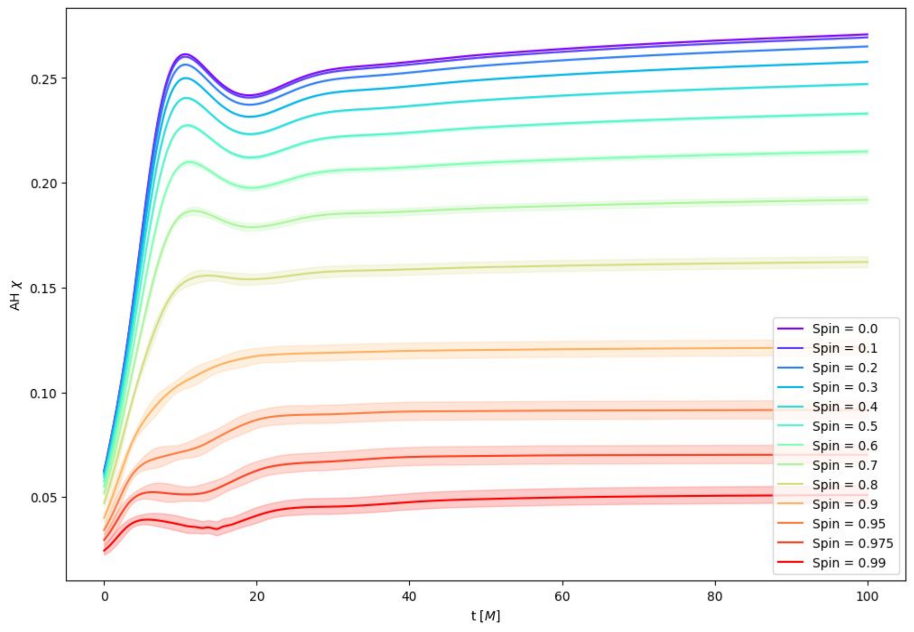
The spin also suffers from a drift, just like the mass.

AH analysis

How does the AH behave in puncture gauge?

The AH radius and value of χ on the AH:



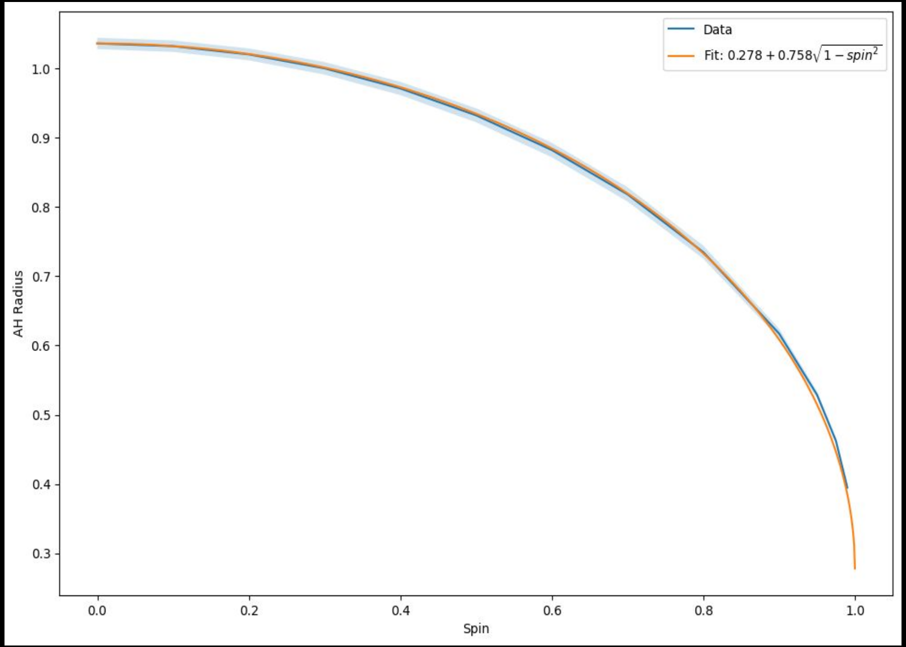


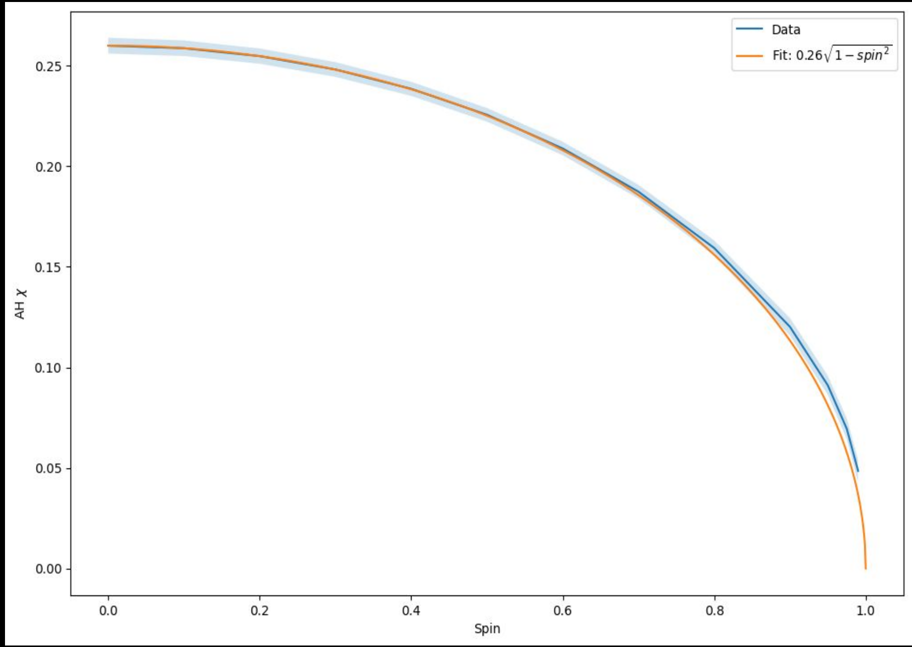
Both wobble slightly until some asymptotic value. In the plots above we already see some drift at later times, just like for the area/mass/spin. For spin=0, we can see that asymptotic behavior settles at χ ~ 0.26 and r ~ M.

AH analysis

How does the AH behave in puncture gauge?

Asymptotic values of AH radius and value of χ on the AH vs spin:





The asymptotic values was extracted for each spin and a fit with an appropriate expression can be seen in the legend. This is just an estimate, but often a useful one. The blue region around the blue line is an estimate of the error from the numerical simulations.