AHFinder

Miguel Alcubierre/Latex version by Peter Diener

Date: 2003/07/07 12:18:45

Abstract

Finding Apparent Horizons in a numerical spacetime

1 Purpose

Thorn `AHFinder` finds Apparent Horizons (AHs) in numerical spacetimes. It calulates various quantities like horizon area and its corresponding mass.

2 Using AHFinder

Thorn AHFinder can be used either with a minimization or a flow algorithm.

2.1 Parameters

General parameters

- `ahfinder::ahf_active` (steerable)
  To activate the thorn set ahf_active = ”yes”. This parameter is set by default to ”no”.

- `ahfinder::ahf_flow` (steerable)
  By default the minimization algorithm is used. To switch to the flow algorithm one has to set ahf_flow = ”yes”

Parameters used in evolutions

- `ahfinder::ahf_findevery` (steerable)
  Specifies how often the finder is called. The default is to find horizons at every iteration.
- `ahfinder::ahf_findafter` (steerable)
  The number of iterations after which the thorn is called the first time can be specified by this parameter. Here the default is 0, calling the thorn also on the initial time slice.
- `ahfinder::ahf_findaftertime`
  Instead of specifying the number of iterations, one can specify after how much coordinate time the thorn is called the first time. When different from zero, this parameter overrides the value of ahf_findafter. Default here is also 0.

Parameters specifying the expansion of the surface in sperical harmonics.

- `ahfinder::ahf_lmax`
  The maximal number of terms in the expansion in θ. The default value is 8. The maximal value is 19.

- `ahfinder::ahf_phi` (steerable)
  If axisymmetry is expected the surface does not need to be expanded in phi. This is the default. To look for non-axisymmetric surface use ahf_phi = ”yes”.

- `ahfinder::ahf_[xyz]c` (steerable)
  Sets the x-, y-, and z-coordinates of the center of the expansion. The default is the origin (ahf_xc = 0, ahf_yc = 0, ahf_zc = 0). The center of the expansion should be set inside the expected apparent horizon, otherwise the algorithm will fail.

- `ahfinder::ahf_wander`
  The center of the expansion can also be allowed to move. To do this use ahf_wander = ”yes”. However, this only works with the minimization algorithm. The default is for the center not to move.

- `ahfinder::ahf_r0` (steerable)
  Sets the radius of the initial sphere. The default is 0.0, forcing the largest sphere possible in the grid.

Looking for three horizons

The finder can also be used to look for three horizons each time. This is done by just running the algorithm three consecutive times with different initial guesses, and is useful for simulations of black hole collisions.

- `ahfinder::ahf_find3`
  Set ahf_find3 = ”yes” to search for three horizons. The default is to look for only one horizon.

- `ahfinder::ahf_[xyz]_[012]` (steerable)
  Sets the x-, y-, and z-coordinates of the center of the expansion for horizon 0, 1 and 2. The default in each case is the origin.

- `ahfinder::ahf_r0_[0-2]` (steerable)
  Sets the radius of the initial spheres for horizon 0, 1 and 2. The default in all cases is 0.0, forcing the largest sphere possible in the grid.

Further parameters for the initial guess

The initial guess can be furthermore controlled by some parameters which are set to ”no” by default.

- `ahfinder::ahf_guessold` (steerable)
  To use on old horizon as initial guess set ahf_guessold = ”yes”. However, if during the evolution the apparent horizon jumps discontinuously it might be lost by using this option.

- `ahfinder::ahf_nn0, ahfinder::ahf_nn2`
  If no old horizon is used the inital guess can be specified further for the minimization algorithm. This algorithm is sensitive to the initial guess, so this is important. The initial guess is set up by an expansion in spherical harmonics in the first two coefficients (l=0,l=2). The default for both these parameters are 10, in which case the algorithm tests 100 different combinations to find the best initial guess.

- `ahfinder::ahf_sloppyguess`
  It is also possible to use only a sphere as initial guess. This is much faster and is done by using ahf_sloppyguess = ”yes”. In this case a number of spheres (specified by ahf_nn0) with different radii are tested for the initial guess.

- `ahfinder::ahf_inner`
  If one wants to look for an inner horizon instead of an outer one, use ahfinder::ahf_inner = ”yes”. This only works with the minimization algorithm.

Parameters for surface intergrals

- `ahfinder::ahf_ntheta` (steerable)
  The number of subdivisions in θ. Default is 200.
- `ahfinder::ahf_nphi` (steerable)
  The number of subdivisions in ϕ. Default is 200.

Parameters indicating symmetries

- `ahfinder::ahf_ref[xyz]` (steerable)
  Specifies the existence of reflection symmetry on the yz-plane, xz-plane and xy-plane respectively. By default all are set to "no".
- `ahfinder::ahf_octant` (steerable)
  Octant symmetry is specified by using ahf_octant = "yes". This is set to "no" by default. Possible parmeter settings are "yes" for reflection symmetries on all three coordinate planes and "high" for an additional rotational symmetry of $\pi/2$ around the z axis.
- `ahfinder::ahf_cartoon` (steerable)
  A further symmetry can (must) be specified when running with the "axisymmetric" mode given by the Cartoon method by using ahf_cartoon = "yes". This is set to "no" by default.

Parameters for minimization algorithm

- `ahfinder::ahf_tol`
  Sets the tolerance for the minimization algorithm. The default value is 0.1.
- `ahfinder::ahf_maxiter`
  Sets the maximum number if iterations for each step of the POWELL algorithm. The default value is 10.
- `ahfinder::ahf_minarea`
  Usually the square of the expansion is minimized. To switch to minimization of the area one can use ahf_minarea = "yes" (default is "no"). Notice that only for time symmetric data the surface found by area minimization will correspond to an apparent horizon.

Parameters for the flow algorithm

- `ahfinder::ahf_flow` (steerable)
  The fow algotithm is used by setting ahf_flow = "yes". The default is "no"
- `ahfinder::ahf_flowiter` (steerable)
  Sets the maximum number of iterations for the flow algorithm. The default value is 200.
- `ahfinder::ahf_flowtol`
  Sets the tolerance for the flow. The default value is 0.0001.
- `ahfinder::ahf_flowa`
  Sets the α parameter for the flow. The default value is 0.01.
- `ahfinder::ahf_flowb`
  Sets the β parameter for the flow. The default value is 0.5.
- `ahfinder::ahf_flowh`
  Sets the weight of H flow. The default value is 0.0.
- `ahfinder::ahf_flowc`
  Sets the weight of C flow. The default value is 1.0.
- `ahfinder::ahf_flown`
  Sets the weight of N flow (not yet implemented). The default value is 0.0.

The character of the different flows and the α and β parameters are described in Carsten Gundlach's paper on his pseudo-spectral apparent horizon finder (gr-qc/9707050).

Parameters for output

- `ahfinder::ahf_logfile`
  By default no logfile for AHFinder is written. To obtain a log file one must set ahf_logfile = "yes".
- `ahfinder::ahf_verbose`
  if ahf_verbose = "yes" messages are printed to screen at the beginning and the end of the algorithm. This is the default.
- `ahfinder::ahf_veryverbose`
  if ahf_veryverbose = "yes" messages are also printed to screen during the iteration process. The default is "no".
- `ahfinder::ahf_2Doutput` (steerable)
  To get 2D output of grid functions use ahf_2Doutput = "yes" (default is "no"). The output is controlled be the thorn itself, not by cactus standard output.
- `ahfinder::ahf_3Doutput` (steerable)
  In future versions to get 3D output of grid functions use ahf_3Doutput = "yes" (default is "no").
- `ahfinder::ahf_areamap`
  To get an area map use ahf_areamap = "yes" (default is "no"). This is useful for looking at the behaviour of the area for surfaces close to the horizon.

Parameters for mask

- `ahfinder::ahf_mask` (steerable)
  The mask is 0 inside the horizon and 1 outside, and is used in black-hole excision techniques. By default the mask is off. It can be enabled by setting ahf_mask = "strong", which sets the mask only if the finder is sure that a horizon was found, or by setting ahf_mask = "weak", which makes the finder set the mask also if a horizon is probably there but either the resolution or lmax are to low to really resolve it.
- `ahfinder::ahf_masktype` (steerable)
  The mask can be of two types. Either ahf_masktype = "simple", which masks a cube contained inside the horizons, or ahf_masktype = "lego", which masks a lego-sphere.
- `ahfinder::ahf_maskshrink` (steerable)
  Sets a buffer zone between the region inside the horizon and the region where the mask is 0, by limiting the mask to a region smaller than ahf_maskshrink times the horizon radius. The default value is 0.8.
- `ahfinder::ahf_maskbuffer` (steerable)
  Sets a buffer zone between the region inside the horizon and the region where the mask is 0 with a width of at least ahf_maskbuffer grid points. The default value is 5.

2.2 Minimal parameter settings

Usually only a few of the parameters described above are needed in the parameter file. The simplest parameter settings for using the flow algorithm for a full 3D horizon with a large sphere as initial guess is

```
interpolation_order = 2     # Second order interpolation
ahf_active          = "yes"
ahf_flow            = "yes"
ahf_phi             = "yes"
```

This looks for a non-axisymmetric horizon around the origin with lmax = 8 and using the flow algorithm. It starts with the largest sphere that fits in the 3D grid and outputs 2D grid functions. The other parameters can be used if needed.

2.3 Hints for parameter settings

In full 3D the flow algorithm is faster than the minimization algorithm. However, in cases when there are very few terms in the expansion in spherical harmonics the minimization can be faster. In axisymmetry this typically happens for lmax$\leq$ 10.

While the default settings usually work fine, they can be changed to meet special purposes:

- If the horizon is expected to be far from spherical the parameter ahf_lmax can be set to a higher value. 12 should be high enough. However, only values up to 20 are supported.
- If the latter parameter is set to a value higher than 8 then the parameter ahf_maxiter can be raised to, e.g., 14. This can be useful since more iterations can be necessary for higher coefficients of the expansion.

2.4 Output to Files

The output of the thorn consists of two gridfunctions and several one dimensional output files.

- To depict the position of the horizon, the most important files are ahfgrid_2d_...ieee. These files contain a 2D gridfunction whose zero level locates the horizon.
- The files ahf_exp_2d_...ieee show the expansion of outgoing photons on the level set of the gridfunction ahfgrid. The horizon coincides with zeros of the expansion.
- The surface area of the horizon is given in ahf_area.tl
- The mass of the horizon is given in ahf_mass.tl
- The coefficients of the expansion in spherical harmonics are given in ahf_coeff.alm.
- The files ahf_circ_eq.tl, ahf_meri_p1.tl and ahf_meri_p2.tl contain the equatorial circumference of the surface, the length of the meridian at $\phi = 0$, and the length of the meridian at $\phi = \pi/2$ of the surface, respectively.
- If an output of a logfile is set in the parameters, the log file for the last time the horizon was called is ahf_logfile.

2.5 Some results with the finder

The finder has been examined with puncture initial data for single and binary-black hole scenarios.

Calculations with different grid spacings but constant grid size show convergence of the horizon area.

This has been checked with different linear momenta in the z direction $P_z = (0M, 2M, 5M)$ and vanishing spin. Also for $P_z = 2M$ and a spin of $5M$ in the x direction the horizon converges. Figure 1 shows the case with $P_z = 2M$ and vanishing spin.

Further, not only the area converges but also the shape of the horizon. For both the minimization and the flow algorithm the horizon converges to the same shape, as can be seen from the coefficients fo the expansion. The order of convergence for the coefficients is between 1.4 and 1.7.

By using the parameters ahf_xc, ahf_yc, ahf_zc it can also be shown that the finder also locates horizons which are not centered. This works in general as long as the surface can be expanded in spherical harmonics around this point, but the error increases with the off-centering.

The parameter ahf_r0 can be used e.g. when dealing with two black holes. If one searches for separate horizons one can center the finder on one of the locations of the holes and use an initial radius ahf_r0 smaller than the coordinate distance of the holes. With this parameter settings the single horizon can be found faster. But also a setup with an initial sphere of maximum radius should work at least for

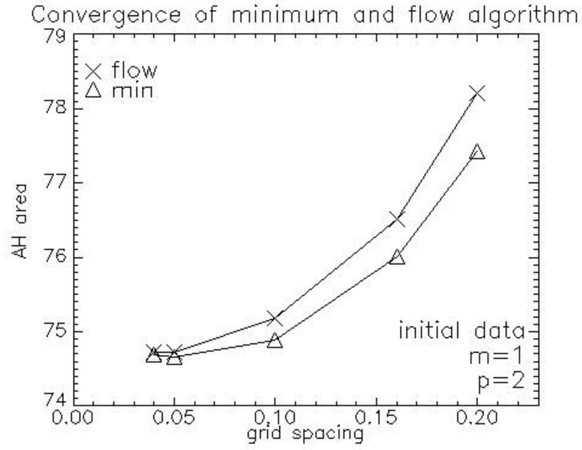

Figure 1: Convergence of the horizon area for $P_z = 2M$

the flow algorithm. This has been checked with puncture data for two holes with vanishing linear and angular momentum for each hole (equivalent to Brill-Lindquist data) and is shown in Figure 2. Here for a coordinate distance of the holes of 1.6M the separated horizons for the holes are found but no common horizon. For a coordinate distance of 1.5M a common horizon is found and also single ones, which are inner surfaces in this case. This coincides with other work where the critical coordinate distance for a single horizon is between 1.53M and 1.56M (gr-qc/9809004).

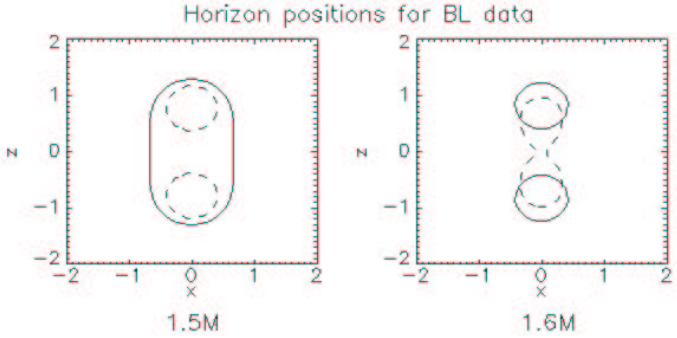

Figure 2: Horizon positions for BL data

The dashed lines show inner trapped surfaces in the left figure and the surface where the algorithm stopped without finding a horizon in the right figure.

Also the Misner case was checked. Here for $\mu = 1.35$ a common horizon is found. For $\mu = 1.37$ separated horizons are found. From the literature we know that (e.g. gr-qc/9809004) the critical value of μ is 1.36. This is confirmed by the horizon finder.

The information of when a horizon was found can be seen in the cactus-logfile. There will be output from the thorn even if no horizon was found.

References