BOLA: Boundary Layer for AGN

Tiago Costa (Newcastle University, UK)

November 8, 2023

1 Overview

To introduce a spherical boundary surface, two spherical layers of AREPO cells must be placed around the wind source. These two layers form a ball (BOLA, in Portuguese).

The innermost layer serves two purposes: (1) to fill up the volume enclosed by the spherical boundary (all volume must be associated to a cell in AREPO), and (2) to ensure a smooth spherical boundary instead of a rough, jagged surface. Conserved quantities within these cells are *not* used to compute fluxes and their values can be ignored when analysing the simulation output.

If a wind is injected across the boundary, the outermost layer is where the wind mass, momentum and energy are first deposited. Conserved quantities such as mass, momentum and energy are updated in these cells both as they receive wind material and as this is advected to the surrounding 'regular' AREPO cells.

The cells making up the two boundary layers form a HealPix tessellation, such that the solid angle is partitioned into curvilinear quadrilateral pixels of identical area. The number $n_{\rm pix}$ of pixels is given by $n_{\rm pix} = 12 \times n_{\rm side}^2$, for any given HealPix $n_{\rm side}$ parameter. In BOLA, the timestep length is set by the size of cells in the outer layer. Setting high values for $n_{\rm side}$ means the solid angle is more finely resolved, but will result in shorter timesteps and will increase the computational cost of the simulation.

The relative positions of boundary cells are kept constant in time. Similarly, boundary layer cells are not allowed to be refined or de-refined. Note, however, that the *volume* of cells in the outermost layer can change with time. This is because these cells are bounded at large radii by 'regular' AREPO cells, which are free to change shape as the mesh evolves, allowing the interface between 'regular' AREPO and cells in the outermost layer to be modified (see Figure 1).

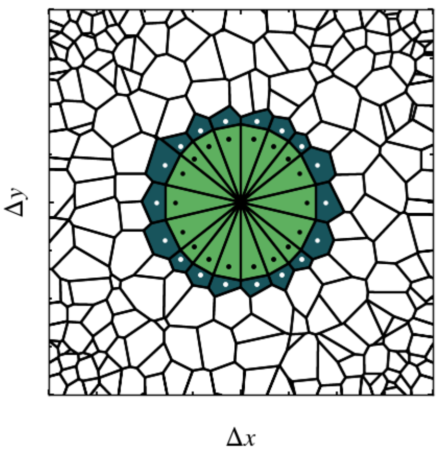

Figure 1: Cell configuration of BOLA, showing inner (light green) and outer layers (dark green) as well as their mesh-generating points (black and white, respectively). These two cell layers define an approximately spherical boundary.

The fact that the mesh-generating points constituting the boundary layers are not free to move breaks the quasi-Lagrangian character of AREPO, because the velocity of the mesh-generating points is now very different from the local gas velocity. This behaviour creates a difficulty since mesh-generating points of 'regular' cells located immediately outside the boundary surface by default move at the local gas velocity. If a fast wind flows through the boundary surface, these mesh-generating points will rapidly move away in the radial direction, in contrast to the mesh-generating points of boundary layer cells, which cannot move. As a result, cells around the boundary tend to become narrow and elongated along the wind flow direction. To avoid the formation of such poorly-shaped cells, we adopt a more aggressive form of the regularisation strategy of AREPO: a small drift speed is imparted to mesh-generating points, pushing them towards the centre of mass, thus ensuring that cells become approximately round. Instead of capping this drift speed to the local speed of sound, however, we allow the regularisation drift speed to exceed the speed of the injected wind by a small factor in any cell that has an interface to a boundary layer cell. This strategy introduces some level of dissipation due to larger advection errors, but these can be mitigated by refining the wind at its base.

Together with mass, momentum and energy fluxes, BOLA always injects a passive scalar across the boundary at a rate proportional to the wind mass flux. This passive scalar makes sure the code 'knows' where the injected wind is located after it is introduced. On the one hand, this option is useful in post-processing and can, for instance, be used to separate shocked ambient gas and shocked wind phases of an outflow, or to quantify the extent to which an AGN-driven wind has managed to percolate a porous interstellar medium.

Another important application of the passive scalar is that it can be used to prevent the wind fluid from de-refining. This option can be enabled with REFINE_AGNWIND in Config.sh.

2 Compiling AREPO to run BOLA

To inject a wind across the boundary surface, the first step is to checkout the branch bola on the AREPO repository and make sure the following compilation option is enabled in Config.sh:

- AGNWIND_FLAG

Various free parameters will have to be set in the AREPO parameter file in order to run BOLA. These parameters set both the location of the source(s), the properties of the injected wind and the accretion flow.

2.1 Wind injection

To inject a wind, the required parameters are:

AGNPosX Spatial x coordinates of source in code units,

AGNPosY Spatial y coordinates of source in code units,

AGNPosZ Spatial z coordinates of source in code units,

InnerLayerID Integer flag that identifies cells making up the inner layer [recommended: 1],

OuterLayerID Integer flag that identifies cells making up the outer layer. This value should be set to InnerLayerID + 1 [recommended: 2],

SphereRadAGNWind Radius of the boundary surface in code units.

If BOLA moves (e.g. a black hole moving relative to a gaseous medium), the grid closely around it has to move along with the black hole. Switching on BOLA_FREEZEMESH in Config.sh ensures that the velocity of mesh-generating points converge to that of BOLA as they approach its centre. Note, however, that, by 'freezing' the mesh, this feature results in larger advection errors. If BOLA_FREEZEMESH is enabled, then one additional parameter must be defined in the parameter file:

FreezeMeshRadius The mesh is gradually made static (in the central source's frame), starting at a radial distance of FreezeMeshRadius x SphereRadAGNWind from the wind source.

The injected wind properties are governed by the parameters:

LuminosityAGN Bolometric luminosity of the central AGN in $[\mathrm{erg\,s^{-1}}]$,

MomentumAGNWind Factor by which the wind momentum flux is boosted above L_{AGN}/c, where L_{AGN} is LuminosityAGN and c is the speed of light in vacuum. This parameter is typically ~ 1,

TemperatureAGNWind Temperature of the wind at injection in [K]. The wind temperature does not play a role in outflow dynamics if the flow is highly supersonic. This parameter can be set safely to any value $< 5 \times 10^6\,\mathrm{K}$,

VelocityAGNWind Wind radial speed across the boundary surface in $[\mathrm{km\,s^{-1}}]$. Typical AGN wind speeds range from $1,000\,\mathrm{km\,s^{-1}}$ to $30,000\,\mathrm{km\,s^{-1}}$.

BoostMeshDriftinWind Factor by which the regularisation speed of mesh-generating points in cells with an interface to BOLA cells exceeds VelocityAGNWind [recommended: 3].

If wind refinement is enabled, there is one additional parameter:

ResolutionBoostAGNWind Factor by which the target mass of wind cells is reduced compared to the typical target cell mass in the simulation. This option requires enabling REFINE_AGNWIND.

2.2 Flickering AGN

It is also possible to inject an intermittent wind. The wind power is assumed to follow a square-wave 'light-curve'. This option can be enabled by switching-on `AGN_DUTYCYCLE`. The user will have to define the following parameters in the parameter file:

`TimeFirstBurst` The time at which the first wind injection episode takes place [code units],

`TimeLastBurst` The time at which the last wind injection episode starts [code units],

`NumBursts` The desired number of wind injection events,

`BurstDuration` The duration of a single wind injection episode [Myr].

Wind injection episodes occur at regular time intervals $\Delta t = (\texttt{TimeLastBurst} - \texttt{TimeFirstBurst})/(\texttt{NumBursts} - 1)$. The simulation will be interrupted and the code will output an error if $\Delta t \leq \texttt{BurstDuration}$, i.e. the nonsensical scenario that the wind episode lifetime is longer than the time interval between different episodes.

3 Acknowledging `BOLA`

Code development constitutes a very significant time investment on the part of developers. The majority of scientific numerical projects is only possible because of such painstaking code development work. For this reason, if you present results obtained using `BOLA` in published articles or oral presentations, we ask you to please credit **Costa, T., Pakmor, R & Springel, V.:** *Powering galactic superwinds with small-scale AGN winds* (MNRAS, 2020).

Since BOLA is in development, we further ask you to please

1. not conduct development work on the `BOLA` numerical infrastructure without our prior consent,

2. keep me informed about ongoing projects using `BOLA`.