

## CSAM 2015, FZ Julich (CHARM school 14-18 september 2015)

### jetCloudInteractions

In this Euler (gas dynamical) test, we simulate the impact of a Mach 12 jet flow as it passes through lighter external medium (density contrast 3), and finally impacts on a spherical (density stratified) cloud which is denser than the jet. It is inspired from a 3D SPH study done by De Gouveia del Pino, E., ApJ 526, 862-873 (1999).

The user file can be used for either a 2D (purely planar, Cartesian) or a 3D setup. It is meant to demonstrate the use of having an additional tracer quantity added to the Euler equations.

The `amrvacusr.t` file shows how to set up the initial condition, and how to control the left ( $x = 0$ ) inlet boundary by fixing all (primitive) quantities.

The `parfiles` demonstrate the use of combining boundary prescriptions that are already available (free-outflow with ‘cont’ or ‘noinflow’ conditions) with a special (here, fixed) treatment of the inlet.

Use

```
diff amrvac2D.par amrvac3D.par
```

to spot simple differences between a 2D and a 3D variant. Check for `mxnest` to see how many grid levels you use, together with the (AMR level one) grid settings in `nxlone1`, `nxlone2`, `nxlone3`.

You can try to experiment with different resolutions, different numerical schemes (it now uses a strong stability preserving Runge-Kutta scheme, and a MP5 limiter in an HLLC discretization).

### dustyRM

This is a 2D hydro setup, meant to demonstrate the use of gas-dust evolutions. The setup can be done without any dust species (`parfile.nodust`), or with an arbitrary number of dust bins representing a dust grain size distribution. An example for 4 dust species (`parfile_4dust`) is given.

The setup studies a Mach 2 shock, impinging on an inclined density discontinuity (density contrast of 3). Dust is only present beyond the density discontinuity. The pure hydro setup will show you double mach reflection (twice, both bottom and top are handled reflective), along with Richtmyer-Meshkov (i.e. Kelvin-Helmholtz on the shocked contact) instability development. If dust is included, the dust grain size will determine in how far the dust is coupled to the gas, and possibly evacuated from the vortices.

The `amrvacusr.t` file shows you how to add additional output variables to the `vtu` files which are generated during runtime. In this example, the `vtu` files contain the primitive variables, together with a Schlieren plot of the density (i.e. the stretched density gradient).

Check out how we run at Courant parameter of 1.5 (thanks to the `ssprk54` time stepping).

Note: the `normvar(*)` array is used only in the conversion to `vtu` files, and here also in the drag force computation when dust is present. Note that Paraview may think that the ranges for (dust) densities are too small to render, but this is obviously not true (use the calculator to multiply to a range that paraview does recognize...).

Related research is in ‘Effect of dust on Kelvin-Helmholtz instabilities’, T. Hendrix & R. Keppens, 2014, Astron. & Astrophys. 526, A114, doi:10.1051/0004-6361/201322322

### orszagtang

This directory contains a setup for the Orszag-Tang problem, a typical MHD testcase. The code is set up to handle both 2D and 3D variants, under either an isothermal closure (no energy equation) or in fully adiabatic ideal MHD

(with energy evolution). This example serves to show the use of the `-eos=` flag at compile time. The test lets a Mach 1 vortical flow distort a series of magnetic islands.

Examples are given for 2D and 3D, for isothermal and full MHD (`amrvac.otmhd2D.par`, `amrvac.otmhd3D.par`, `amrvac.otmhdiso2D.par`, `amrvac.otmhdiso3D.par`). Use `diff` to spot obvious differences in these parfiles.

Boundaries are (double to triple) periodic. Note the settings for resolution (up to 5 grid levels in the 2D runs, domain decomposition at  $100^3$  in the 3D runs). When simulated far enough, and at high resolution, these ideal MHD 2D runs will show nice tearing instabilities on the developing current sheets. Note that there is no explicit resistivity here. The monopole condition is handled through a diffusive approach (`typedivbfix='linde'`).

## doubleGEM

This test is meant to illustrate the use of resistive to Hall-MHD evolutions. It simulates the double periodic, double GEM test (this test generalizes the standard Geospace Environment Modeling Challenge, a standard reconnection setup, to a setup where energy conservation can be verified easily, see ‘Resistive MHD reconnection: resolving long-term, chaotic dynamics’, R. Keppens, O. Porth, K. Galsgaard, J.T. Frederiksen, A.L. Restante, G. Lapenta, & C. Parnell, 2013, Phys. of Plasmas 20, 092109 (17pp). doi: 10.1063/1.4820946).

The setup here illustrates the use of Hall MHD, where one additionally needs to activate Hall-physics in the `definitions.h` file. Also, it shows how to use the GLM approach to monopole control, which also requires defining in `definitions.h`. Look for

```
#define GLM
#define HALL
```

and spot the corresponding code parts in the user file (`amrvacusr.t`) or the source code (once compiled, these are in `F90sources/`, where you can inspect the `amrvacphys.f` file, compared to the `amrvacphys.t` file from the `$AMRVAC_DIR/src/mhd` directory).

Note that our (explicit) treatment of the Hall term makes the Hall run (`doublegemmhdrunA`) much more challenging than the resistive MHD run (`doublegemmhdrunB`).

See how we use the `iprob` problem switch in the `amrvacusr.t` and the parfiles, to use the same compiled code for running different problems.

We set up the problems to use fourth order finite differences here (using MP5 limiting) and strong stability preserving Runge-Kutta schemes. Note the use of extra ghost cells (`dixB`).

The test is done in 2.5D (invariance in the ignored  $z$ -direction), note that the resistive run keeps  $B_z$  and  $v_z$  zero at all times, while Hall physics causes the generation of out-of-plane vector components.

A visco-resistive variant of this test was discussed in ‘MPI-AMRVAC for solar and astrophysics’, O. Porth, C. Xia, T. Hendrix, S.P. Moschou, & R. Keppens, 2014, ApJS 214, 4 (26pp) doi:10.1088/0067-0049/214/1/4 .

## lfff

This is a test to demonstrate how to generate a potential or linear force-free field extrapolation from a given (HMI) magnetogram. One can download such magnetograms from the SDO/HMI website, you then get these in FITS format. An IDL script `...pro` is provided, that assumes you have an IDL license and access to the library solar software (SSW) for SDO data handling. This routine just converts the FITS file into a (binary, uniform Cartesian grid) dat file, containing the vertical magnetic field component only. For convenience, an already converted magnetogram (in `*dat`) format is provided as well.

The user file shows how we use the LFFF module (see the `INCLUDE::amrvacmodules/lfff.t` statement) to then generate a potential or linear force-free (look for the `alpha` parameter) magnetic field extrapolation. You should set

the code to a 3D isothermal MHD run (use the `-eos=iso` switch), and the parfile is set up to stop when the initial condition for a possible dynamical evolution is generated (i.e., the only thing that is computed here is the LFFF extrapolation, you should use Paraview to visualize the magnetogram and some selected fieldlines).

Note: to see fieldlines properly in Paraview, we need corner-valued data. This can be done directly on converting to vtU, by selecting e.g. `convert_type='vtuBmpi'`. You can also let the code dump the actual cell-centered values `convert_type='vtuBCCmpi'`, but then need to use a CelldataToPointdata filter, before you can start drawing fieldlines. Another quirk of paraview is that it needs the Interpolator type to be set to **Interpolator with Cell locator**, before the Streamline tracer will be able to handle field lines across resolution changes (which we have in this 3-level AMR extrapolated field).

A description of this functionality and how the LFFF extrapolation works technically is found in (the references of) ‘MPI-AMRVAC for solar and astrophysics’, O. Porth, C. Xia, T. Hendrix, S.P. Moschou, & R. Keppens, 2014, ApJS 214, 4 (26pp) doi:10.1088/0067-0049/214/1/4 .

## pfss

This is a very similar exercise as the one above, but this time generating a Potential Field Source Surface model from a global magnetogram. It also has an IDL routine provided for converting downloadable synoptic maps to more easily handled maps of the radial field component, on a spherical grid, uniform in both spherical angles. This file is read in (and an example \*dat for a specific date is given) by the user file (again with isothermal MHD).

Check the user file to see how we use the PFSS module (see the `INCLUDE::amrvacmodules/pfss.t` statement). The par file is set to stop after the field is generated. You can visualize again with Paraview. Note how we can use AMR to zoom in on specific active regions.

A description of this functionality and how the PFSS extrapolations compares to local cartesian extrapolations, can be found in ‘MPI-AMRVAC for solar and astrophysics’, O. Porth, C. Xia, T. Hendrix, S.P. Moschou, & R. Keppens, 2014, ApJS 214, 4 (26pp) doi:10.1088/0067-0049/214/1/4 .