/
****
***/

Code HARM-COOL version 485. (2019)
/
****
***/

Version prepared for the first public release.

This version, as of 2019, was
developed since 2012, by Agnieszka Janiuk, Ireneusz Janiuk, and Kostas Sapountzis.

Code is suppelemented with the EOS functions for the Fermi Gas, under arbitrary degeneracy
as by Ye-Fei Yuan (see Yuan 2005, Physical Review D, vol. 72, Issue 1, id. 013007),
and produce tables with the equation of state and neutrino cooling in the function of temperature and density.
The physical units adopted in the code are scaled with the black hole mass and density scale in the accretion disk.
Code unit conversion and interpolation over EOS tables is implemented within the conversion scheme between
conserved and primitive variables.

Code makes use of the tracer particles, for the subsequent post-processing of the r-process nucleosynthesis. Output variables on tracers are time, density, temperature, and electron fraction, and r-theta-phi coordinates.

This code version was developed upon the HARM by C. Gammie et al.
It has been extended to 3-Dimensions (phi-r-theta), and tested extensively in 2D and 3D configurations
with various initial conditions.
The output format can be chosen between Ascii standard and HDF5, accprdingly to its internal configuration.
The code is parallelized with MPI, and can be executed with mpirun command, or with proper batch scripts on supercomputers. It is up to the user to compile and run the code with proper libraries and compiling packages. Sample makefile is provided.
Sample initial condition is using the Fishbone-Moncrief torus solution.

Proper citation for this software includes the references given below by the authors of original code,
and in addition:

0. Janiuk Agnieszka; Yuan, Ye-Fei; Perna, Rosalba; Di Matteo, Tiziana, 2007, Astrophysical Journal, 664, 1011

1. Janiuk, Agnieszka; Mioduszewski, Patryk; Moscibrodzka, Monika; 2013, Astrophysical Journal, 776, 105

2. Janiuk, Agnieszka; 2017, Astrophysical Journal, 837, 39

3. Janiuk Agnieszka; Sapountzis Konstantinos; Mortier Jeremy; Janiuk Ireneusz; 2018, Supercomputing Frontiers and Innovations, 5, 86

4. Janiuk Agnieszka; 2019, ApJ, 882, 163

/
**/

Technical setup for compilation requires installing the MPI and hdf5 libraries.
Sample Makefile is provided.
Initial condition is provided with the standard Fishbone-Moncrief solution parameterized as in Gammie et al. (2003), and described in original README below.
To subsitute this initial condition with user-specified file, one must replace the file init.c, added as object in the Makefile.

Grid setups are chosen in decs.h and 2 options are available:
COORDS_GAMMIE
COORDS_CYLINDRIFY_GAMMIE
First are defined according to Gammie et al. (2003) and are the Modified Kerr-Shild coordinates. Second option are the cylindrified coordinates, where a few innermost cells around the pole become cylindrical See details in coord.c.

Resolution: setup in decs.h.
Default resolution is 288x256x256 in r-theta-phi directions.

Grid is distributed into threads in r and phi only. The User must take care for the proper slab division, when specifying the number of computational nodes and CPUs per node.
Standard command for running on 4 CPU on the desktop computer:
> mpirun -np 4 ./harm
For massive computations, it is recommended though to use computer clusters and own batch scripts for execution.

The initial analyze of MPI process execution is saved in stdout.txt.all for all the threads.

Output formats are either ASCII or hdf5 for dumping of the grid variables.
Defauls is HDF5, as specified in decs.h by option
HDFDUMP
Here, the user may choose which variables will be stored in the HDF files, by choosing (1) or (0) options. See dump.c for the definitions and details of the stored quantities.
Alternatively, specifically in the 2D (N3=1) simulations, the HDF output may be switched off, and then the code uses standard ASCII output dumps.

The equation of state is specified by the option
#define COOL_EOS
in decs.h
In default (0) case, the adiabatic gamma-law EOS is used, and parameterised by gamma in the init.c.
If the option
COOL_EOS
is turned on, the nuclear EOS of degenerate relativistic matter is computed, for the range of densities and temperatures.
See details in cooleos.c.
Once the code is executed, the tables of EOS for evey grid point are saved in the /cache sub-directory, for the future reading and interpolation.

The storage of tracer particles is activated by option
TRACERSDUMP
in decs.h.
Default setup is no storage of tracers.
If the tracers are stored, the code after it is executed, makes 3 catalogues:
/tracers/in, /tracers/out, and
/tracers/jet,
for the test particles that leave the domain through the inner, or the outer boundary
(in the last case, the jet is defined as a part of outer boundary close to the poles). See details in tracers.c.

The ppm output files in /images directory are turned off (see image.c).

Any questions or bug reports can be sent to:
agnes@cft.edu.pl.

/
****
***/

Below, we provide the original README file from HARM-2D code, written by its Authors.

/
****
***

                        HARM version 1.0   (released May 1, 2006)

    This file is part of HARM. HARM is a program that solves hyperbolic
    partial differential equations in conservative form using high-resolution
    shock-capturing techniques. This version of HARM has been configured to
    solve the relativistic magnetohydrodynamic equations of motion on a
    stationary black hole spacetime in Kerr-Schild coordinates to evolve
    an accretion disk model.

    You are morally obligated to cite the following two papers in his/her
    scientific literature that results from use of any part of HARM:

    [1] Gammie, C. F., McKinney, J. C., \& Toth, G.\ 2003,
        Astrophysical Journal, 589, 444.

    [2] Noble, S. C., Gammie, C. F., McKinney, J. C., \& Del Zanna, L. \ 2006,
        Astrophysical Journal, 641, 626.

    Further, we strongly encourage you to obtain the latest version of
    HARM directly from our distribution website:
    http://rainman.astro.uiuc.edu/codelib/

****
***/

INTRODUCTION:

HARM is a conservative finite volume approach to solving the hyperbolic partial
differential equations (PDE). Even though it was written with general

relativistic
magnetohydrodynamics (GRMHD) specifically in mind, it can solve almost any
set of hyperbolic equations in conservation form:

dU / dt + dF(U) / dx^i = S(U)

where, if there is a set of equations, {U,F,S} are all state vectors and
represent the set of conserved variables, fluxes and sources, respectively.
There are also primitive variables, P, that are useful when solving hydrodynamic
equations. Often, one only know how to calculate F given P, requiring the user
to solve P(U) = U^{-1}(P). U(P) is a known algebraic set of equations in
GRMHD,
but P(U) is not so we are left to its calculation numerically. HARM performs
most
calculations with P and not with U, since one can derive all other MHD
quantities
from P efficiently (calculating P(U) is relatively slow compared to U(P)).
Hence,
the fluxes and sources are implemented here as functions of P.

The following are the files that comprise HARM. To the left are their names and
on the right are short descriptions.

HARM includes the following files:

README : This file, contains general documentation
lic.txt : Text file of copyright information
COPYING : GNU Public License (GPL)
makefile : The makefile
main.c : Primary routine, calls major components of the code;
bounds.c : Boundary conditions (problem dependent);
coord.c : Specify/handle the user's coordinate system and metric;
decs.h : Header file incl. glob. arrays/vars., macros, compile-
time parameters;
defs.h : Header file of definitions of those in decs.h
diag.c : Primary handler of all data output;
dump.c : Routine for writing grid functions at full precision
over whole grid;
fixup.c : Routines for handling unphysical/unstable values of P
and U ;
image.c : Routines for writing r8 or ppm raster images of grid
functions;
image_interp.c : Supplemental program to interpolate an r8 file to x,y
coordinates;
init.c : Procedures for calculating initial data;
interp.c : Slope-limiting/shock-capturing interpolation for Riemann
solution;
lu.c : LU-decomp. and back-subst. routines for metric calculations
metric.c : Routines for doing tensor operations (inv., dot prod., lower/
raising);
phys.c : For calculating U, F, S from P;
ranc.c : Random number generator;
restart.c : Checkpointing routines;
step_ch.c : Primary time-stepping routine and relatives;
u2p_defs.h : Inversion methods' main header file;
u2p_util.c : Misc. routines needed by inversion methods;
u2p_util.h : Header for u2p_util.h;
utoprim_1dfix1.c : P(U) assuming adiabatic or isothermal condition;
utoprim_1dvsq2fix1.c: Alternate version of utoprim_1dfix1.c
utoprim_2d.c : General GRMHD P(U) calculator with no assumptions;
map.ppm : PPM color map used to make images in PPM format;
maps : Directory containing other examples of color maps;
maps/blue-mono.ppm : Black-to-Blue map;
maps/bw.ppm : Greyscale (monochrome) map;

```
maps/color.ppm       :  256-color map;
maps/green-mono.ppm :  Black-to-Green map;
maps/red-mono.ppm   :  Black-to-Red map;
```

As in most finite volume programs, the numerical domain is discretized into parts called "cells." We assume that these cells are uniformly spaced with respect to code coordinates X1 and X2. One can, however, use a non-uniform set of coordinates (e.g. "r" and "theta") that are at least C-4 differentiable functions of X1,X2 (see bl_coord() in coord.c). Further, the user may specify any regular metric (arbitrary excision is not implemented) at compile-time in gcov_func() [coord.c].

HARM was written in a modular way so that methods can be easily interchanged. For instance, different slope limiters can be inserted into slope_lim() [interp.c], though HARM is currently hard-coded to have only two ghost zones per boundary. Currently only Lax-Friedrichs-type and HLL algorithms are implemented to calculate the numerical flux function, but other could be installed by altering fluxcalc() [step_ch.c]. A different set of PDE's can be solved by replacing primtoU(), primtoflux(), source(), vchar()
[phys.c] and utoprim_2d() [utoprim_2d.c]. If the number of PDE's changed, though,
one would have to change "NPR" [decs.h] and instances of arrays accessing indexes
beyond current array allocations.

If you wish to use our algorithms and PDE's but want to change the initial conditions, then you need only change init.c and any compile-time parameters in decs.h (see below).

```
##########################################################################################
#######
##########################################################################################
#######
##########################################################################################
#######
##########################################################################################
#######
```

STARTING A RUN:

To get started, first make sure the settings in "makefile" are valid for your system. Once that is done, type the following commands:

```
prompt>  make
prompt>  ./harm
```

to run the simulation. The current setting (128x64 cells) should be an acceptable size for any modern workstation. The code is configured to evolve the following initial conditions:

- Kerr hole using Kerr-Schild metric,
  with a=0.9375 (BH normalized spin/M)

- Fishbone-Moncrief torus w/ inner edge rin=6M,
  pressure max. at rmax=12M

- Azimuthal vector potential component follows contours of constant
  density;

The following files will be generated by HARM (using the default settings):

-- ./ener.out: accretion rates of rest-mass, total energy and
angular momentum onto the black hole
(plus other quantities, see diag.c for details)

-- ./dumps/dump[000-???] : double-precision, plain text output of
grid functions (see dump.c for details)

-- ./images/im_*_[0000-????].ppm : dynamically-scaled snapshots of rho,
bsq, u, the lorentz factor,
and their logarithms at higher
time resolution than the "dumps".
Use ImageMagick's "display" command
to view, and ppm2fli (see below) to
animate.

Parameters that you may want to immediately adjust are:

decs.h: N1, N2 = number of cells in the X1 and X2 directions, respectively.
(X1 is the radial-going direction, X2 is the poloidal-going one).

init.c: a = spin of the black hole (spatial dimensions scale w/ black hole mass,
i.e. M_BH = 1);

gam = adiabatic index for fluid;

rin = radial position of inner edge of torus;
rmax = radial position of pressure maximum;
beta = initial value of 2 P_max / b_max^2
Rout = radial coordinate of outer numerical boundary;
tf = end time of solution in units of black hole mass;
DTd = frequency of writing dump files in units of M
DTi = frequency of writing image files in units of M

##
##
##
##

OUTPUT FORMATS:

The user has a choice of the types of images HARM makes: PPM or R8.
The former is a standard image format, see "man ppm" or the web for more details. R8 is an unmapped format of 1-bit in "color" depth, see image_r8() of image.c for details. R8 images are smaller in size since they do not

provide the color scheme. Their downside is that after generating the R8
images, one must convert them to PPM or some other format with a
user-supplied parsing program. We have left this R8 conversion utility
as an exercise for the user, but have provided the PPM format for those
of you who are lazy and want to see pictures immediately.

ppm:
      -- animate ppm files with either "convert im_lrho*.ppm im_lrho.gif" or use
            ppm2fli (http://vento.pi.tu-berlin.de/ppm2fli/main.html)

r8:
      -- raw format that can be easily converted to a ppm file.

The format of the dump files can be easily gleaned from dump() in dump.c .

##
#######
##
#######
##
#######
##
#######

BUG REPORTS:

-- please send any bug reports to:

       harm@astro.uiuc.edu

##
#######
##
#######
##
#######
##
#######

HISTORY:

-- HARM version 1.0 released to the general public under the GPL on May 1st,
2006;