

Introduction to PLUTO code

Miljenko Čemeljić

SHAO Visiting Scientist (PIFI), CAS



Nicolaus Copernicus Astronomical Center,
Warsaw, Poland



ASIAA, Taiwan, Visiting Scholar



General outline

- PART I: General introduction to PLUTO code and its numerical methods, installation of the code, testing of the installation, gnuplot visualization. Setup of HD accretion disk.
- PART II: Hands-on introduction to setup files of PLUTO. Visualization of HD results with Paraview & VisIt. Animation.
- PART III: Introduction to star-disk magnetospheric interaction simulations, with some examples. Setup of axisymmetric 2D setup with dipole and multipole magnetic field. Visualization of magnetic field lines with Paraview.
- PART IV: Full 3D magnetic setup. Restart and initialization from (modified) files. Visualization of 3D results. Various uses of the results: light-curves for comparison with observations, post-processing of the results (PLUTO+DUSTER): radiative effects in motion of dust grains. Concluding remarks.

Outline, Part I

- A very brief introduction to history of accretion disk numerics
- PLUTO equations, grids, numerical methods
- Test problems, templates.
- Practical part: installing the code, testing with the use of tests provided with the code.
- Initial and boundary conditions setup for purely HD disk

Required packages: C-compiler (MPI), Python (matplotlib, NumPy, SciPy), PLUTO source code, Gnuplot, Paraview, VisIt.

Introduction-history

- First numerical solution of (HD) accretion disk was by Prendergast & Burbidge (1968)
- Analytical solution was given by Shakura & Sunyaev (1973), with alpha-viscosity prescription.
- From that time to the 1990-ies many developments and models, with different approximations.
- In Kluźniak & Kita (2000) was given a solution of the HD disk in the full 3D. It was obtained by the method of asymptotic approximation.
- In 2009, numerical simulations of star-disk magnetospheric interaction were done in 2D-axisymmetric simulations, by Romanova et al. (2009, 2013, with non-public code), Zanni & Ferreira (2009, 2013, with modification of a publicly available code PLUTO v. 3.0).
- Development of disk simulations in the direction of MRI in the disk (Flock et al.), but nothing much in star-disk magnetospheric simulations for a decade.
- In Čemeljić et al. (2017) and Čemeljić (2019, “Atlas” paper), the first repeating of Zanni et al. (2009, 2013) axi-symmetric viscous & resistive MHD simulations in 2D with PLUTO code, v.4.1. The results are similar to Romanova et al., obtained with their (non-public) code with entropy conservation, not energy.
- In “Atlas” paper I performed a parameter study to investigate the influence of different parameters. This publication, especially its Appendix, is used here as a guide through setup and numerical methods.

Equations solved by the code

- Code I used in “Atlas” is PLUTO (v.4.1) by Mignone et al. (2007, 2012). I will show you an updated version of this code, v.4.4
- We will perform simulations of a thin accretion disk.
- If the disk is to reach the quasi-stationary state, the Ohmic and viscous heating in the energy equation should be balanced with the source term. In such a case, we effectively neglect the dissipative terms in the energy equation—this is equal to the assumption that all the heat is radiated away from the disk.
- Viscosity and resistivity are still affecting the solution, in the equation of motion and in the induction equation.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \left[\rho \mathbf{v} \mathbf{v} + \left(P + \frac{\mathbf{B} \cdot \mathbf{B}}{8\pi} \right) \mathbf{I} - \frac{\mathbf{B} \mathbf{B}}{4\pi} - \boldsymbol{\tau} \right] = \rho \mathbf{g}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{\mathbf{B} \cdot \mathbf{B}}{8\pi} \right) \mathbf{v} - \frac{(\mathbf{v} \cdot \mathbf{B}) \mathbf{B}}{4\pi} \right] + \nabla \cdot \left[\underbrace{\eta_m \mathbf{J} \times \mathbf{B} / 4\pi - \mathbf{v} \cdot \boldsymbol{\tau}}_{\text{heating terms}} \right] = \rho \mathbf{g} \cdot \mathbf{v} - \underbrace{\Lambda}_{\text{cooling}}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times (\mathbf{B} \times \mathbf{v} + \eta_m \mathbf{J}) = 0$$

the gravity acceleration is $\mathbf{g} = -\nabla \Phi_g$, where the gravitational potential of the star with mass M_\star is equal to $\Phi_g = -GM_\star/R$. Then $g_R = -1.0/R^2$ in the code units.

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{\mathbf{B} \cdot \mathbf{B}}{8\pi} \right) \mathbf{u} - \frac{(\mathbf{u} \cdot \mathbf{B}) \mathbf{B}}{4\pi} \right] + \nabla \cdot [\eta_m \mathbf{J} \times \mathbf{B} / 4\pi - \mathbf{u} \cdot \boldsymbol{\tau}] = \rho \mathbf{g} \cdot \mathbf{u} - \Lambda_{\text{cool}}$$

Numerical methods in PLUTO for star-disk interaction

- Simulations were performed using the second-order piecewise linear reconstruction.
- Van Leer limiter, which is more diffusive and enhances stability, is used in density and magnetic field and a minmod (monotonized central differences) limiter in pressure and velocity.
- An approximate Roe solver (hlld in the pluto options) was used, with a modification in the `flag_shock` subroutine: flags were set to switch to more diffusive hll solver when the internal energy was lower than 1% of the total energy, instead of switching in the presence of shocks.
- The second-order time-stepping (RK2) was employed.
- $\nabla \cdot \mathbf{B} = 0$ was maintained by the constrained transport method.
- The magnetic field was evolved with the split-field option, so that only changes from the initial stellar magnetic field were evolved in time. This means that in the final results, one has to add magnetic dipole (or other initial field) to the solution from the code.
- The power-law cooling is introduced to account for the disk dissipative heating. Physically it is good enough to represent bremsstrahlung. There are other cooling functions in PLUTO, one can also import a table, not necessarily a function.

Normalization in the code

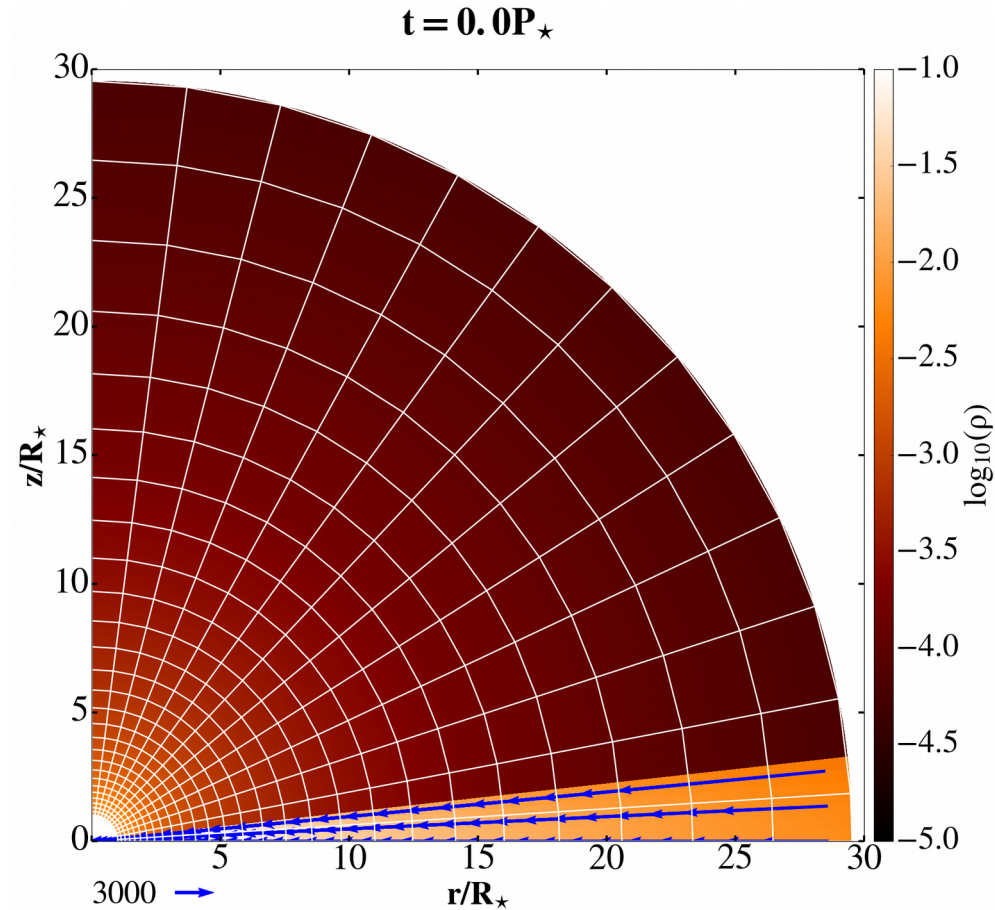
Code works in the normalized units. For translating into other units, one needs to multiply with the corresponding scaling factors. When using Cooling, one needs to define units in definitions.h subroutine.

Table A.1. Typical values and scaling for different central objects.

	YSOs	WDs	NSs
$M_{\star}(M_{\odot})$	0.5	1	1.4
R_{\star}	$2R_{\odot}$	5000 km	10 km
P_{\star}	4.6 d	6.1 s	0.46 ms
B_{\star} (G)	500	5×10^5	10^8
ρ_{d0} (g cm $^{-3}$)	1.2×10^{-10}	9.4×10^{-9}	4.6×10^{-6}
v_0 (km s $^{-1}$)	218	5150	136 000
$\dot{M}_0(M_{\odot}$ yr $^{-1}$)	5.7×10^{-7}	1.9×10^{-9}	10^{-9}
B_0 (G)	200	5×10^4	2.93×10^7

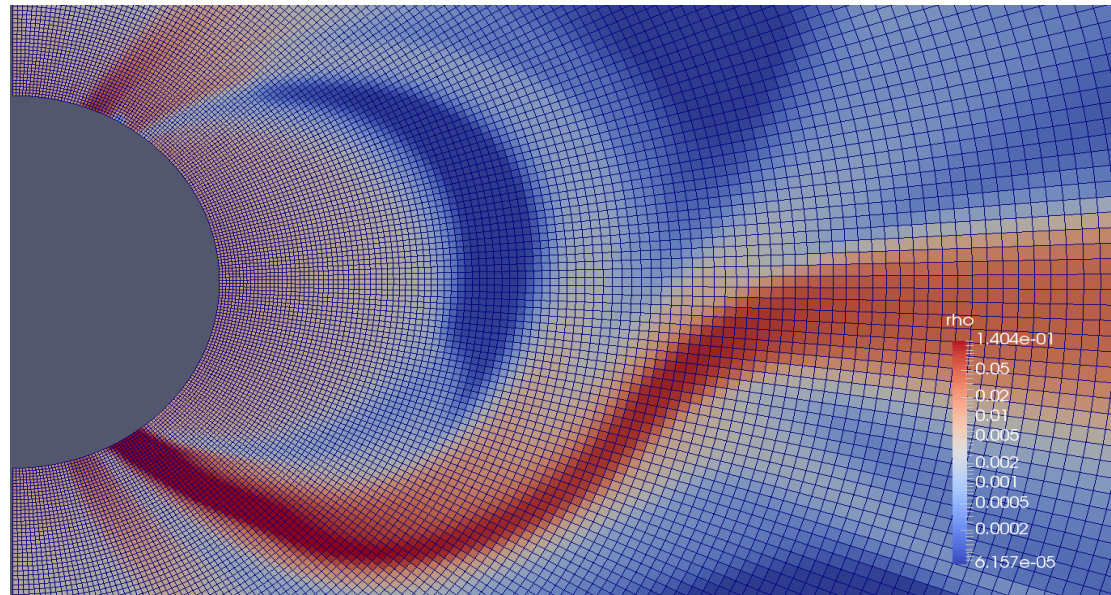
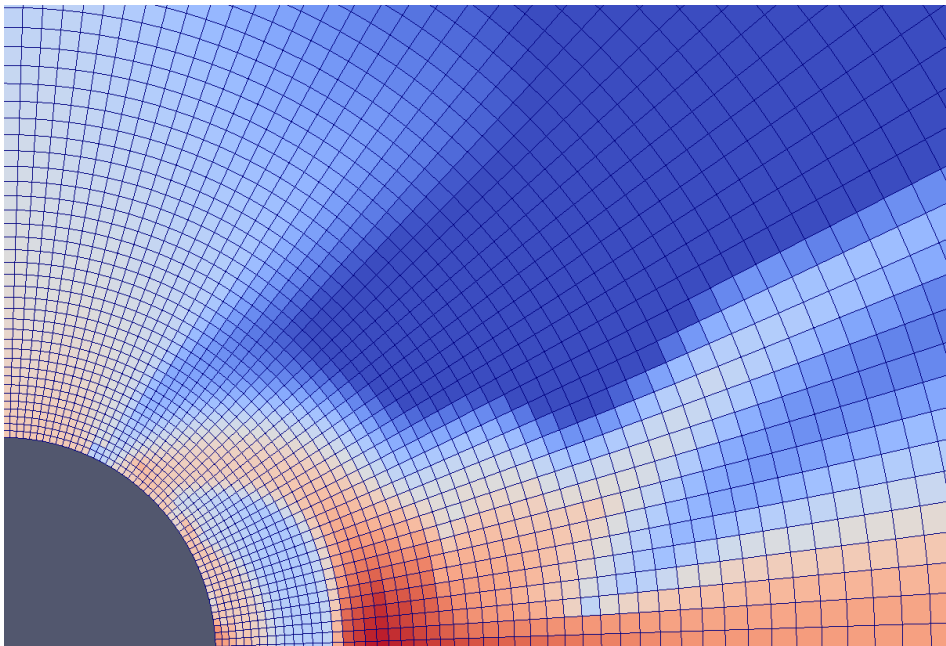
Star-disk simulations setup

- The Kluźniak-Kita (2000) analytical solution is purely HD.
- We will start with the HD disk, and then add the magnetic field.
- Stellar surface is a rotating boundary around the origin of the spherical computational domain. In the non-magnetic setup, it is just a simple setup with “absorption” of the flow atop the stellar surface—we assume the star to be a (differentially) rotating magnetized rotator.
- The initial corona is a non-rotating corona in a hydrostatic balance with the 100 times denser disk below it. After short relaxation, lasting for the few stellar rotations, corona starts rotating, following the disk.



Resolution needed in a setup

- Resolution in my production runs is $R \times \vartheta = [217 \times 100]$ grid cells in $\vartheta = [0, \pi/2]$, with a logarithmic grid spacing in the radial direction. For testing I also use $R \times \vartheta = [109 \times 50]$ grid cells, which gives qualitatively correct results.
- The accretion column is well resolved if a rule of thumb is satisfied that there is at least that many grid cells how many there is independent variables (5 in HD case, 8 in MHD case in our setups: density, pressure, 3 components of \mathbf{v} and \mathbf{B}).
- Star rotates at about 1/10 of the breakup rotational velocity.
- I did also $\vartheta = [0, \pi]$ cases in $R \times \vartheta = [217 \times 200]$ grid cells, as well as $R \times \vartheta = [109 \times 100]$ grid cells. Now there is no need to define the equatorial boundary condition, the simulation self-consistently computes across the domain. This case usually leads to an asymmetry with respect to the equatorial plane.



PLUTO documentation, templates and test examples

- Source files of the code are downloadable from <http://plutocode.ph.unito.it/>
- After unpacking, the source code is in PLUTO directory. In PLUTO/Doc is the manual, userguide.pdf. Follow the “Quick start” at the beginning of the document to **install and test the code. Produce the gnuplot output specified in the manual, to verify if the setup works.**
- The code comes with templates of subroutines which are usually changed. They are given in the PLUTO/Src/Templates
- Test examples, under /PLUTO/Test_Problems, are basic versions of setups used in previous versions of the code, during the developments of the setups for simulations presented in publications by various groups. It is a good library of examples for faster start of one’s own project.
- I will provide the setup for HD and MHD accretion disk for this lectures.

Files for setup and their modifications

- The PLUTO original source is not changed, for our simulations we just append to it our version of some files in a pre-compiling step.
- The changed version of a file in the work directory has, by default in PLUTO, priority to the original version in PLUTO/Src directory.
- The files to be copied into the work directory from the working version into a new setup are **init.c**, **pluto.ini**, **definitions.h**, `res_eta.c`, `visc_nu.c`.
- The file `definitions.h` is the only *.h file changed in the work directory. All the other *.h files are to be changed directly in PLUTO/Src directory.
- In the file **pluto.ini** are defined the grid, solvers and run parameters.
- Most of the entries in **definitions.h** are done through the python environment during the setup of the run, but some entries are to be done by hand editing the `definitions.h` file, prior to compilation.
- In the file **init.c** is defined the physics setup.

Practical part 1: init.c file, Hydro-dynamical star-disk setup

The initial disk was set with the initial density ρ a self-similar profile and the aspect ratio ϵ :

$$\begin{aligned}\rho_d &= \rho_{d0} \left\{ \frac{\gamma - 1}{\gamma \epsilon^2} \left[\frac{R_\star}{R} - \left(1 - \frac{\gamma \epsilon^2}{\gamma - 1} \right) \frac{R_\star}{R \sin \theta} \right] \right\}^{1/(\gamma-1)} \\ &= \rho_{d0} \left\{ \frac{2}{5 \epsilon^2} \left[\frac{R_\star}{R} - \left(1 - \frac{5}{2} \epsilon^2 \right) \frac{R_\star}{R \sin \theta} \right] \right\}^{3/2}.\end{aligned}$$

The pressure was

$$\begin{aligned}P_d &= \epsilon^2 \rho_{d0} v_{K\star}^2 \left(\frac{\rho_d}{\rho_{d0}} \right)^\gamma = \\ &= \frac{\rho_{d0} v_{K\star}^2}{\epsilon^3} \left\{ \frac{2}{5} \left[\frac{R_\star}{R} - \left(1 - \frac{5 \epsilon^2}{2} \right) \frac{R_\star}{R \sin \theta} \right] \right\}^{5/2}\end{aligned}$$

The initial disk velocity profile is, following KK00:

$$\begin{aligned}v_{Rd} &= -\alpha_v \epsilon^2 \left[10 - \frac{32}{3} \Lambda \alpha_v^2 - \Lambda \left(5 - \frac{1}{\epsilon^2 \tan^2 \theta} \right) \right] \sqrt{\frac{GM_\star}{R \sin^3 \theta}}, \\ v_{R\phi} &= \left[\sqrt{1 - \frac{5 \epsilon^2}{2}} + \frac{2}{3} \epsilon^2 \alpha_v^2 \Lambda \left(1 - \frac{6}{5 \epsilon^2 \tan^2 \theta} \right) \right] \sqrt{\frac{GM_\star}{R \sin \theta}}\end{aligned}\quad (\text{A.10})$$

where

$$\Lambda = \frac{11}{5} / \left(1 + \frac{64}{25} \alpha_v^2 \right).\quad (\text{A.11})$$

In the corona, density and pressure are:

$$\begin{aligned}\rho_c &= \rho_{c0} (R_\star / R)^{1/(\gamma-1)}, \\ P_c &= \rho_{c0} \frac{\gamma - 1}{\gamma} \frac{GM_\star}{R_\star} \left(\frac{R_\star}{R} \right)^{\gamma/(\gamma-1)}\end{aligned}$$

Viscosity tensor is given by Landau-Lifshitz def:

$$\tau = \eta_v \left[(\nabla \mathbf{v}) + (\nabla \mathbf{v})^T - \frac{2}{3} (\nabla \cdot \mathbf{v}) \mathbf{I} \right],\quad (\text{A.7})$$

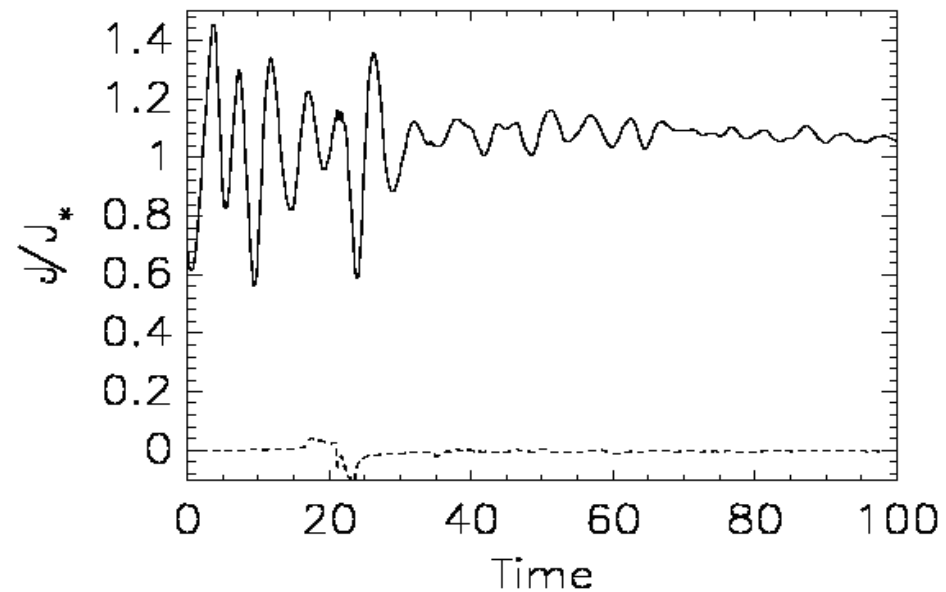
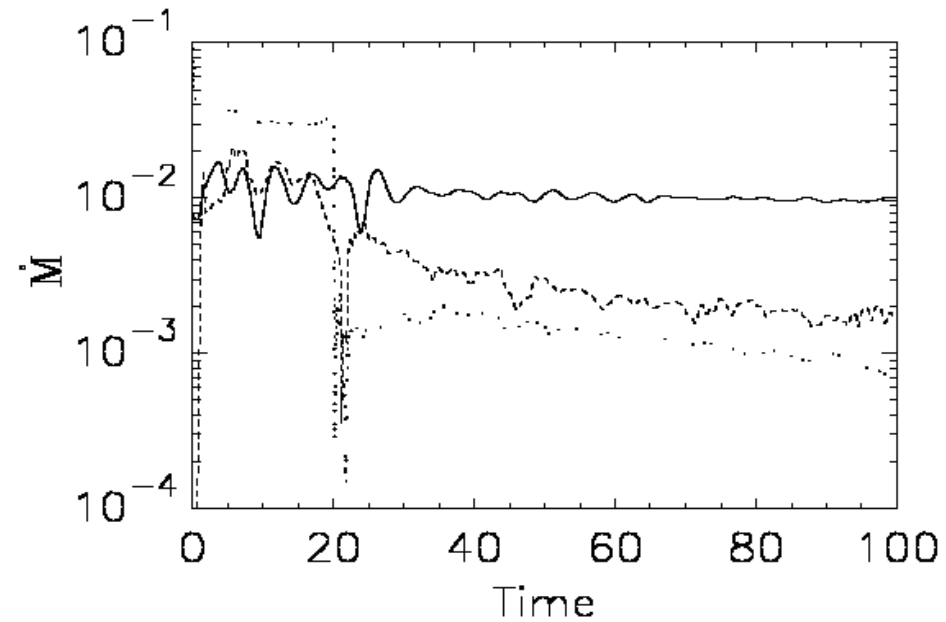
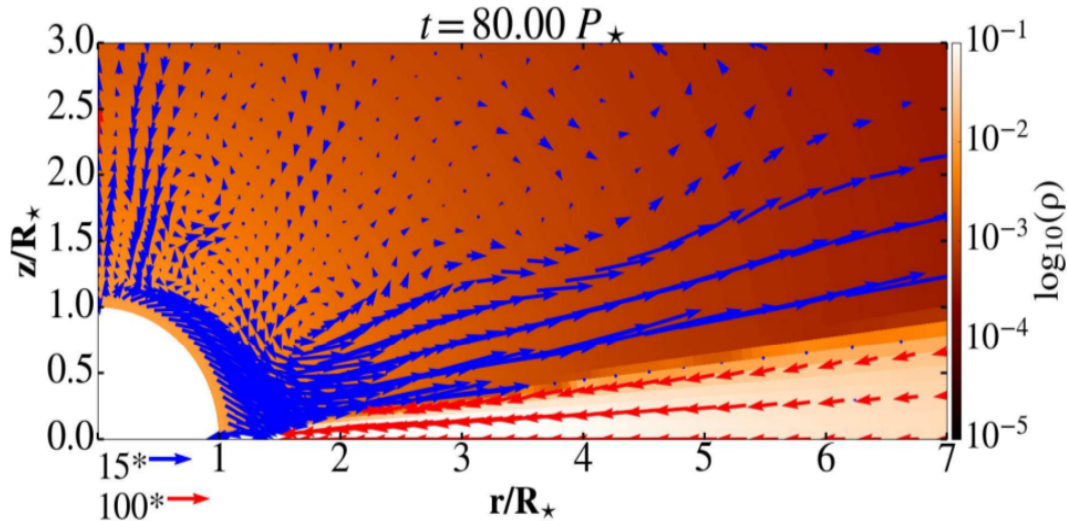
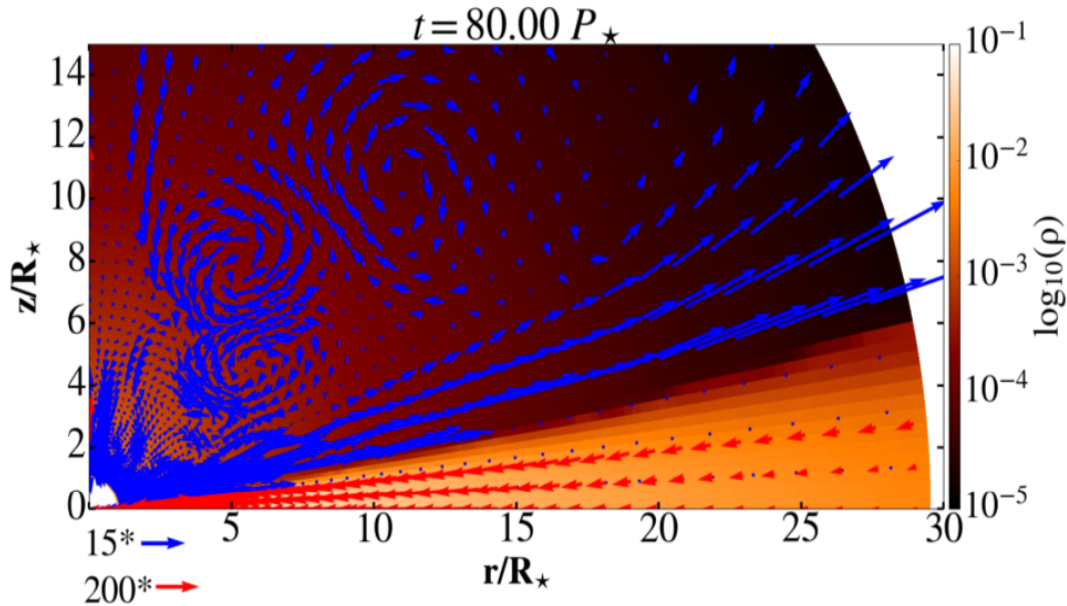
with the dynamic viscosity $\eta_v = \rho \nu_v$ given with

$$\eta_v = \frac{2}{3} \rho \alpha_v \left[c_s^2(r)|_{z=0} + \frac{2}{5} \left(\frac{GM_\star}{R} - \frac{GM_\star}{r} \right) \right] \sqrt{\frac{r^3}{GM_\star}},\quad (\text{A.8})$$

where ν_v is the kinematic viscosity. The magnetic diffusivity was assumed proportional to the viscosity, with the free parameter

$$\nu_m = \frac{3}{2} \alpha_m \frac{\nu_v}{\alpha_v},$$

Practical part 1: results in Hydro-dynamical star-disk simulations



Ideally, this is what we wish to obtain. Computational box and a zoom closer to the star after 80 stellar rotations. In color is shown the density, and vectors show velocity, with the different normalization in the disk and stellar wind. Next time we will learn to plot this in Paraview & VisIt.

Time dependence of the mass and angular momentum fluxes in the various components in our simulations.

Outline, Part II

- Hands-on introduction to setup files of PLUTO.
- Output formats & files to save.
- Visualization of HD results with Paraview.
- Visualization of HD results with VisIt.
- Animation, movies time!
- I show results with Python, some of it is included with the code.

Hands-on introduction to setup files of PLUTO

- Guided tour through the source files:
- Python setup.py
- pluto.ini-grid, solver, output steps, parameters
- Definition of output formats
- Files to save together with output files
- definitions.h-macros setting
- init.c-physical setup of initial and boundary conditions

Hands-on introduction: pluto.ini

```
I pluto.ini (ini) Row 22 Col 1 2:51 Ctrl-K H for help
[Grid]
X1-grid 1 1.0 109 l+ 30.
X2-grid 1 0.0 50 u 1.570796327
X3-grid 1 0.0 1 u 1.0

[Chombo Refinement]
Levels 4
Ref_ratio 2 2 2 2 2
Regrid_interval 2 2 2 2
Refine_thresh 0.3
Tag_buffer_size 3
Block_factor 8
Max_grid_size 64
Fill_ratio 0.75

[Time]
CFL 0.4
CFL_max_var 1.2
tstop 628.0
first_dt 1.e-6

[Solver]
Solver roe

[Boundary]
X1-beg userdef
X1-end userdef
X2-beg axisymmetric
** Doe's Own Editor v4.1 ** (utf-8) ** Copyright © 2015 **
```

```
I pluto.ini (ini) Row 57 Col 1 2:52 Ctrl-K H for help
X2-end eqtsymmetric
X3-beg outflow
X3-end outflow

[Static Grid Output]
uservar 3 nu num Te
dbl 6.279 -1 single_file
flt -1.0 -1 single_file
vtk 6.279 -1 single_file
tab -1.0 -1
dbl.h5 -6.279 -1
ppm -1.0 -1
png -1.0 -1
log 100
analysis -1.0 -1

[Chombo HDF5 output]
Checkpoint_interval -1
Plot_interval 1.0

[Parameters]
ALPHAM 0.4
MU 0.1
TEMPF 300.0
RHOC 0.01
RD 1.7
EPS 0.1
OMG 0.1
ALPHAV 1.0
DFLOOR 5.e-7
```

Output formats & files to save

- In pluto.ini we defined which output
- data.xxxx.dbl files – viewing, processing with Python, idl etc. packages.
-also, for restart, together with restart.out.
- Data.xxxxdbl.h5 - hdf5 files-need to set the flag in /PLUTO/Config/*.defs file
- data.xxxx.vtk files – for viewing with Paraview, VisIt.
- Together with data.xxxx.dbl and .vtk files, always save also, in the same directory with the results, pluto.ini, definitions.h. init.c, grid.out, dbl.out, restart.out, to know which setup produced the files, and also for some analysis and plotting packages, which might need them.
- grid.out files defines the geometry, dbl.out lists the output variables

Hands-on introduction: definitions.h

```
I A definitions.h (c) Row 3 Col 1 2:53 Ctrl-K H for help
#define PHYSICS HD
#define DIMENSIONS 2
#define COMPONENTS 3
#define GEOMETRY SPHERICAL
#define BODY_FORCE VECTOR
#define FORCED_TURB NO
#define COOLING NO
#define RECONSTRUCTION LINEAR
#define TIME_STEPPING RK2
#define DIMENSIONAL_SPLITTING NO
#define NTRACER 1
#define USER_DEF_PARAMETERS 9

/* -- physics dependent declarations -- */

#define DUST_FLUID NO
#define EOS IDEAL
#define ENTROPY_SWITCH NO
#define THERMAL_CONDUCTION NO
#define VISCOSITY EXPLICIT
#define ROTATING_FRAME NO

/* -- user-defined parameters (labels) -- */

#define ALPHAM 0
#define MU 1
#define TEMPF 2
#define RHOC 3
#define RD 4
#define EPS 5
#define OMG 6
#define ALPHAV 7
#define DFLOOR 8
** Joe's Own Editor v4.1 ** (utf-8) ** Copyright © 2015 **
```

```
I A definitions.h (c) Row 18 Col 1 2:53 Ctrl-K H for help
#define DUST_FLUID NO
#define EOS IDEAL
#define ENTROPY_SWITCH NO
#define THERMAL_CONDUCTION NO
#define VISCOSITY EXPLICIT
#define ROTATING_FRAME NO

/* -- user-defined parameters (labels) -- */

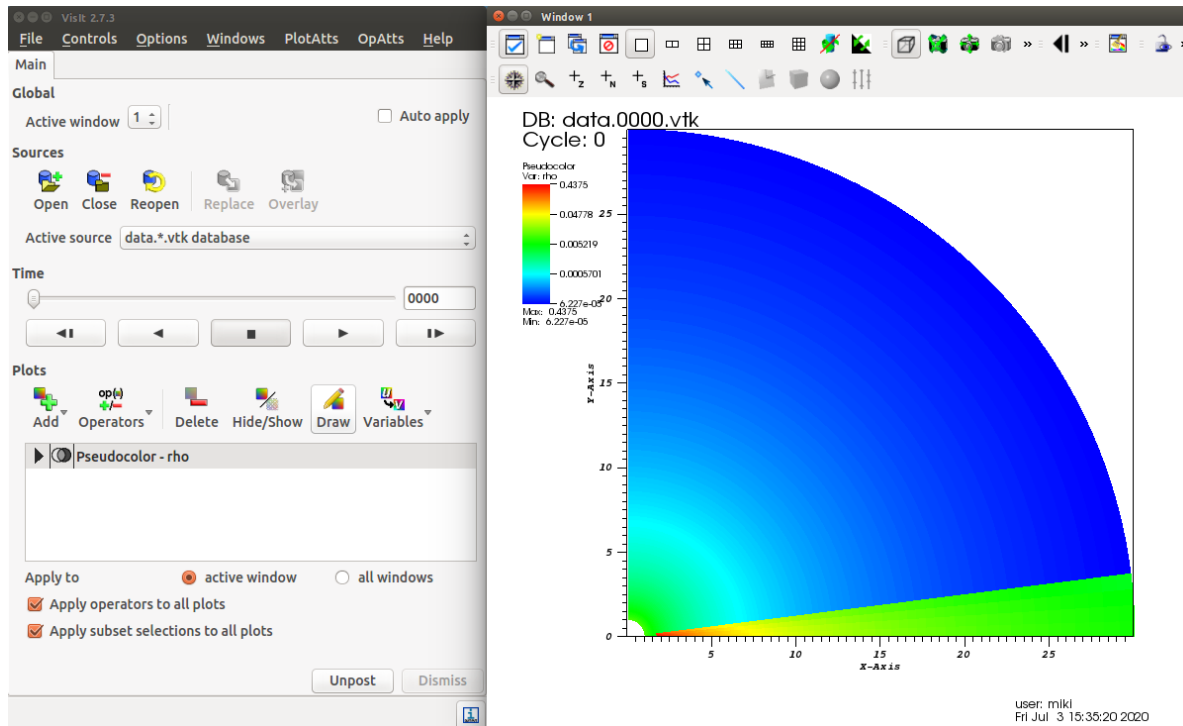
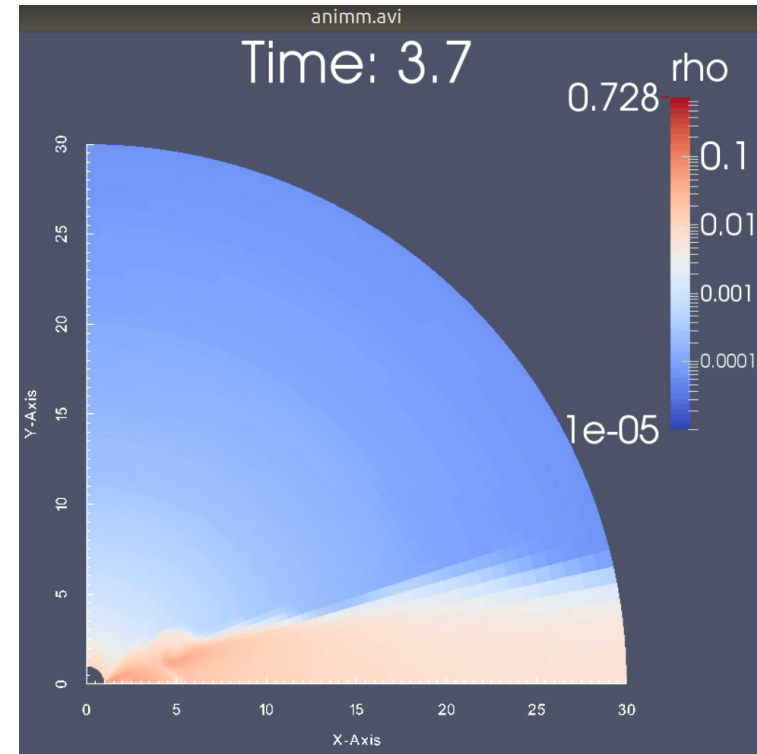
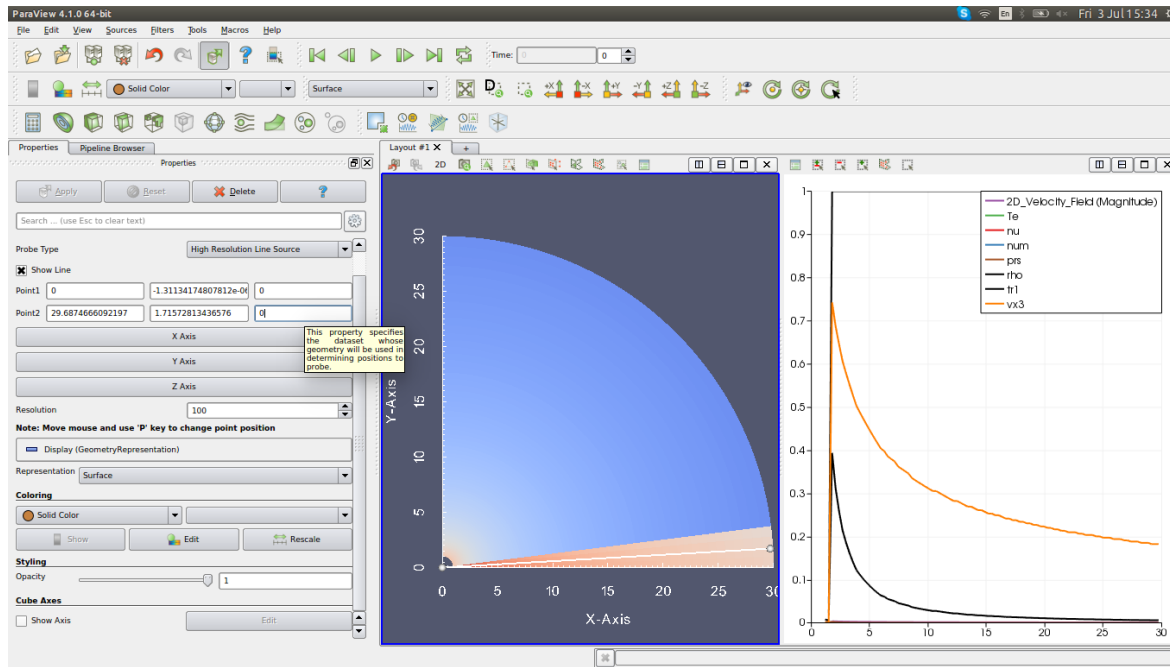
#define ALPHAM 0
#define MU 1
#define TEMPF 2
#define RHOC 3
#define RD 4
#define EPS 5
#define OMG 6
#define ALPHAV 7
#define DFLOOR 8

/* [Beg] user-defined constants (do not change this line) */

#define WARNING_MESSAGES NO
#define INTERNAL_BOUNDARY YES
#define SHOCK_FLATTENING MULTID
#define CT_EN_CORRECTION YES
#define UNIT_DENSITY 8.5e-11
#define UNIT_LENGTH 1.392e11
#define UNIT_VELOCITY 2.1839e7
#define VTK_VECTOR_DUMP YES
#define CT_EMF_AVERAGE ARITHMETIC
#define LIMITER VANLEER_LIM

/* [End] user-defined constants (do not change this line) */
```


Visualization packages: Paraview, VisIt, animation

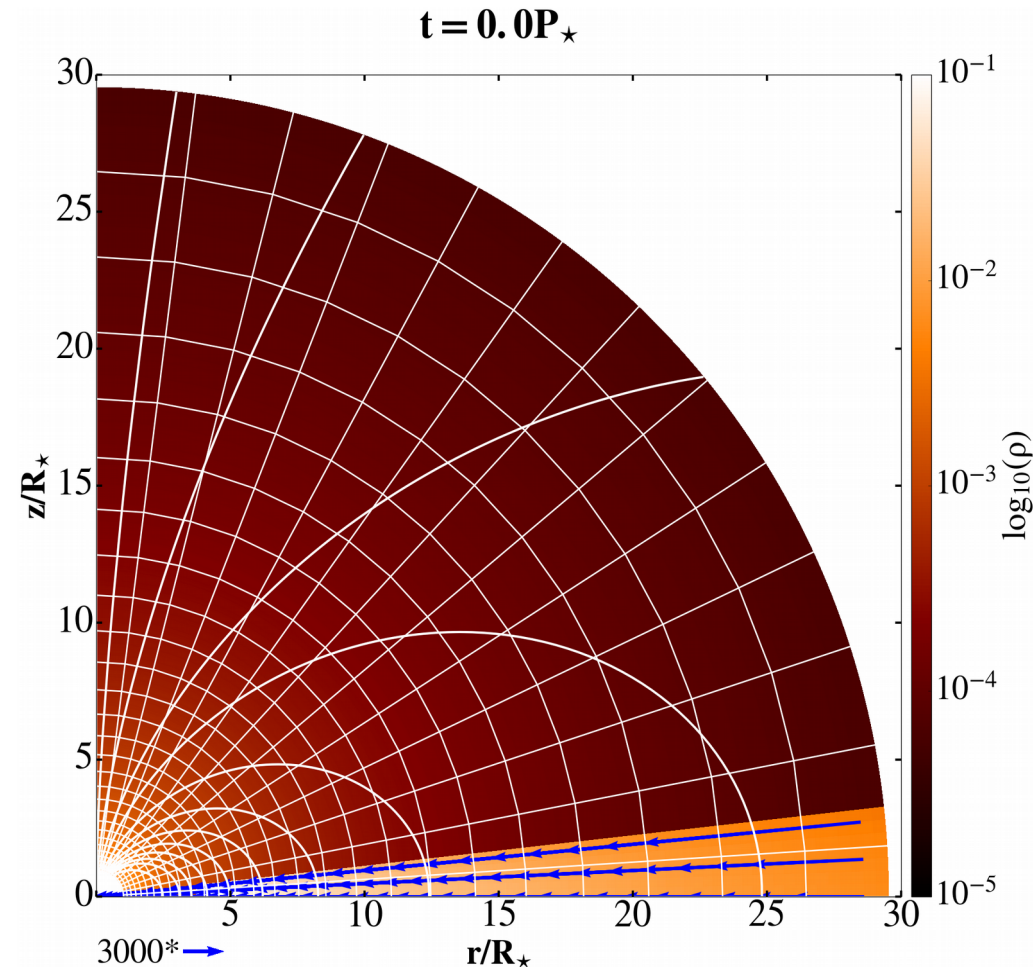


Outline, Part III

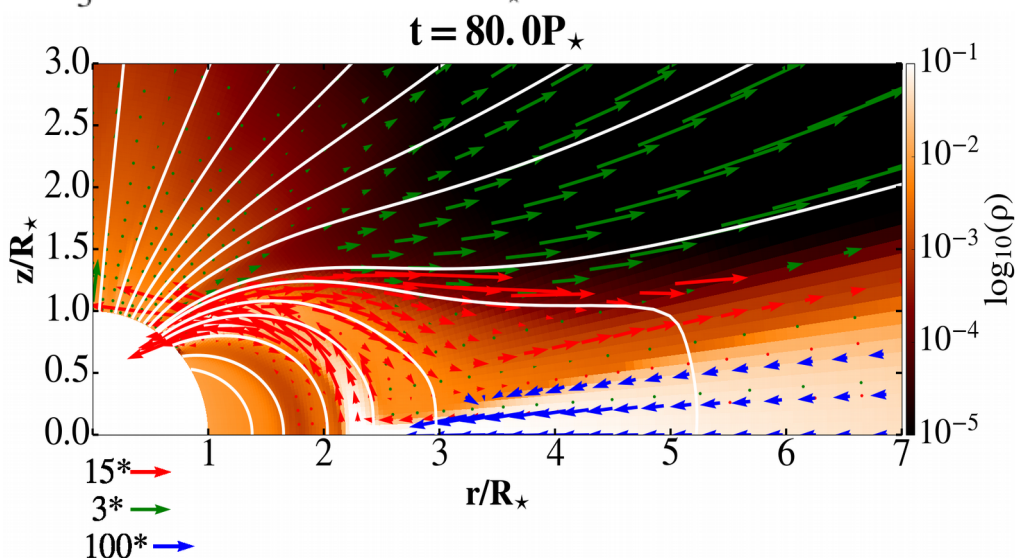
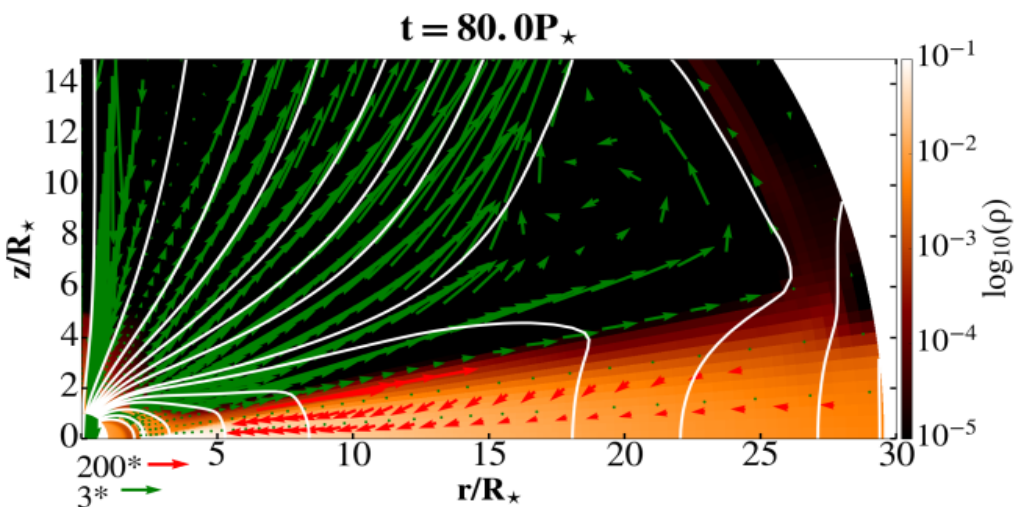
- Introduction to star-disk magnetospheric interaction simulations.
- Setup of axisymmetric 2D setup(=2.5D) with dipole magnetic field.
- Visualization of magnetic field lines with Paraview.

Setup of magnetic star-disk interaction

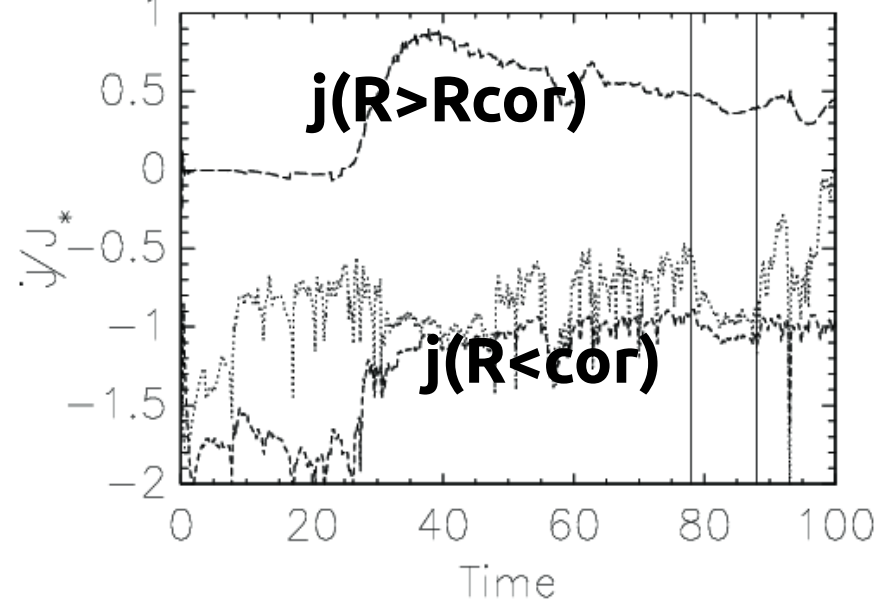
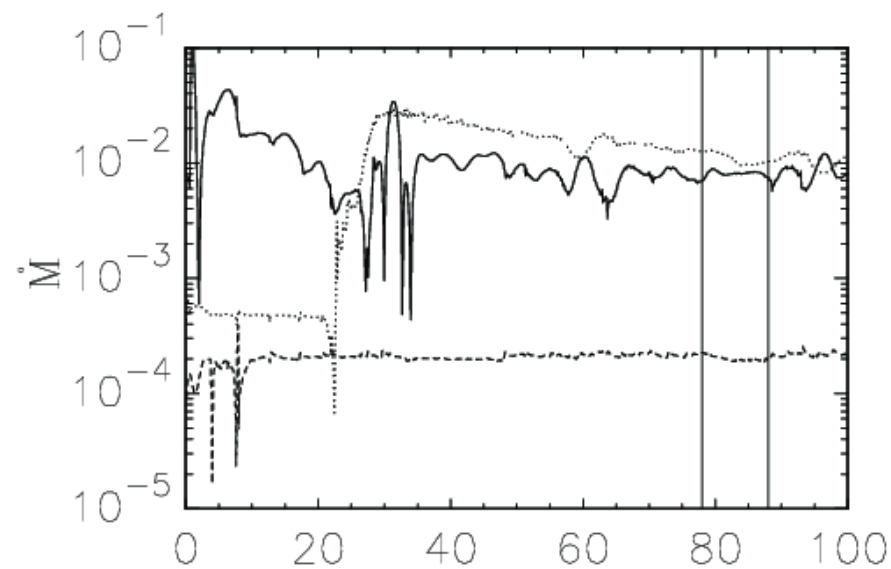
- The Kluźniak-Kita (2000) analytical solution is purely HD. We now add the stellar dipole magnetic field, with the axis matching the stellar rotation axis, so that the axi-symmetry would be preserved. Later we will set the simulations with quadrupole, octupole and mixed, multipole field.
- Stellar surface is a rotating boundary around the origin of the spherical computational domain. In the magnetic setup, it is **not** a simple setup with “absorption” of the flow atop the stellar surface—we assume the star to be a (differentially) rotating magnetized rotator. The initial corona is a non-rotating corona in a hydrostatic balance. After short relaxation, lasting for the few stellar rotations, corona starts rotating, following the disk.



Results in star-disk magnetospheric interaction simulations

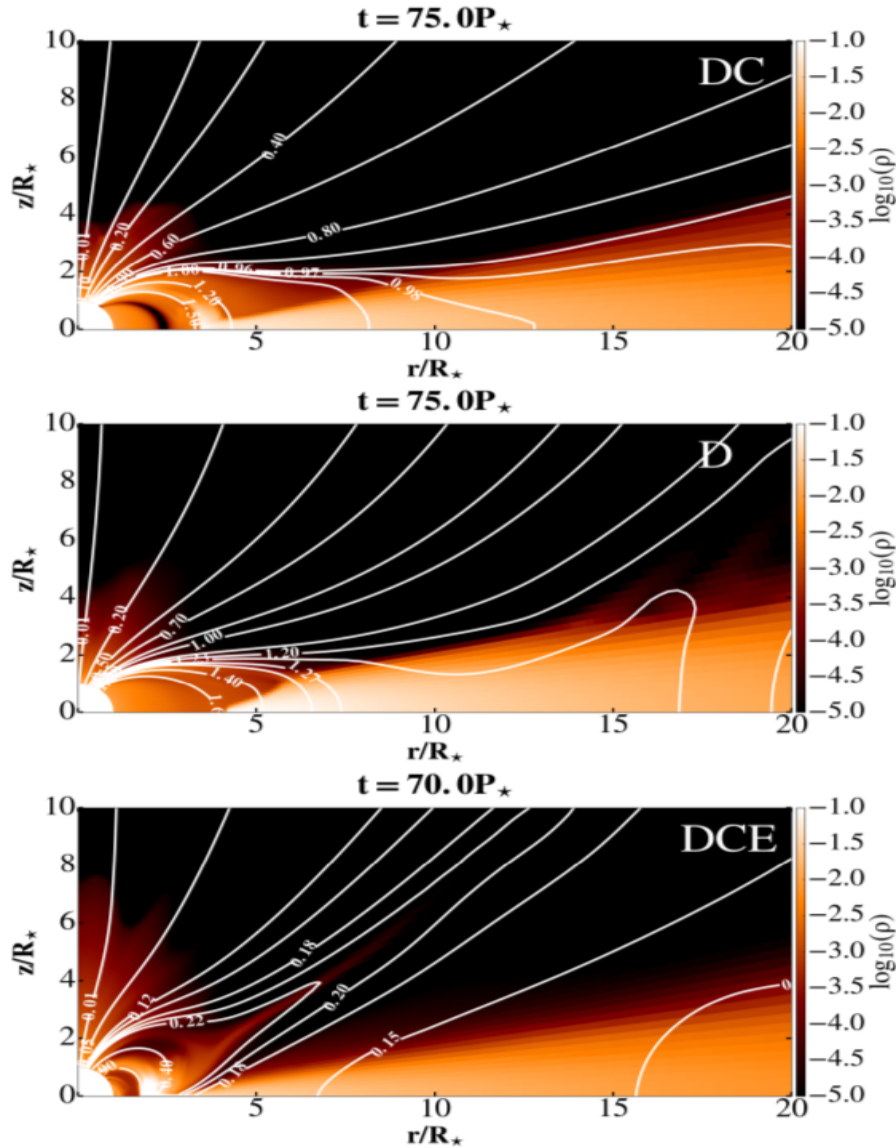


Computational box and a zoom closer to the star after 80 stellar rotations, to visualize the accretion column and the magnetic field lines (white solid lines) connected to the disk beyond the corotation radius $R_{\text{cor}}=2.92 R_{\text{s}}$. In color is shown the density, and vectors show velocity, with the different normalization in the disk, column and stellar wind.



Time dependence of the mass and angular momentum fluxes in the various components in our simulations with marked time interval in which the average value during the quasi-stationarity is computed.

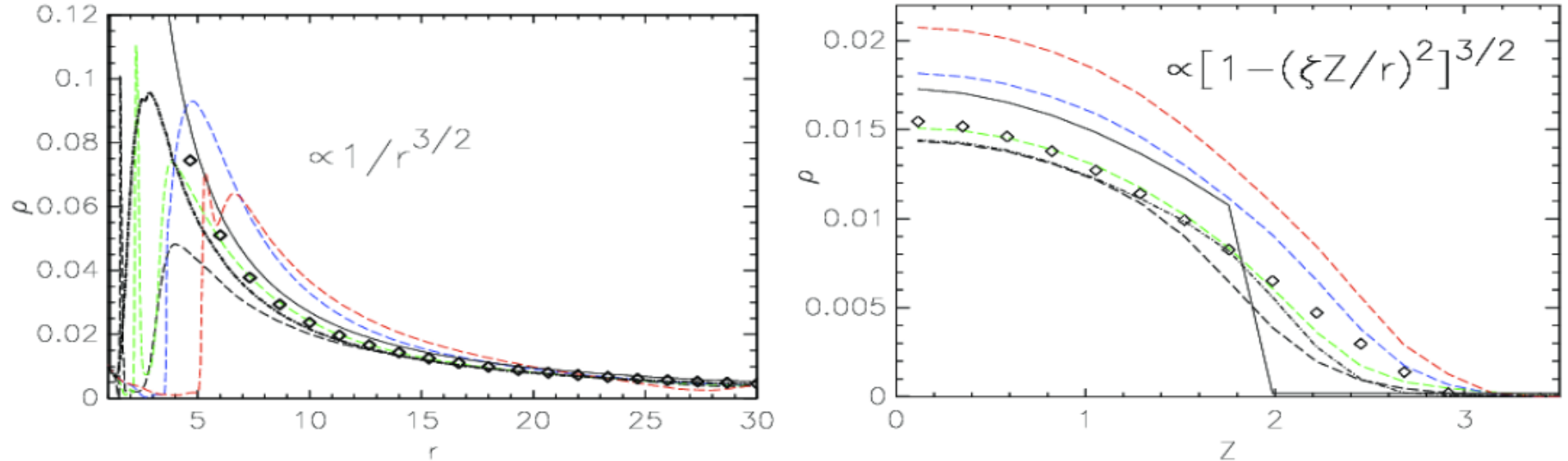
Types of solutions for slowly rotating stars in “Atlas”



Three different cases of geometry in the results. In the top and middle panels are shown $B=1$ kG and the resistivity $\alpha_m=1$, in the cases with $\Omega_s=0.1$ (top panel) and $\Omega_s=0.15$ (middle panel). Faster stellar rotation prevents the accretion column formation. In the bottom panel is shown the third case, with $B=0.5$ kG, resistivity $\alpha_m=0.1$ and $\Omega_s=0.1$, where a conical outflow is formed.

$\alpha_m =$	0.1	0.4	0.7	1
Ω_*/Ω_{br}				
$B_* = 250$ G				
0.05	DCE	DC	DC	DC
0.1	DCE	DC	DC	DC
0.15	DCE	DC	DC	DC
0.2	DCE	DC	DC	DC
$B_* = 500$ G				
0.05	DCE	DC	DC	DC
0.1	DCE	DC	DC	DC
0.15	DCE	DC	DC	DC
0.2	DCE	DC	DC	DC
$B_* = 750$ G				
0.05	DCE	DC	DC	DC
0.1	DCE	DC	DC	DC
0.15	DCE	DC	DC	DC
0.2	DCE	DC	DC	DC
$B_* = 1000$ G				
0.05	DCE	DC	DC	DC
0.1	DCE	DC	DC	DC
0.15	DCE	D	D	D
0.2	DCE	D	D	D

Comparison of results with increasing mag. Field strength



Matter density in the initial set-up (thin solid line) with the quasi-stationary solutions in numerical simulations in the HD (dot-dashed line) and the MHD (long-dashed line) cases. In black, green blue and red colors are the results in the MHD cases with the stellar magnetic field strength 0.25, 0.5, 0.75 and 1.0 kG, respectively. The closest match to the 0.5 kG case is depicted with the diamond symbols. In the left panel is shown the radial dependence nearby the disk equatorial plane, and in the right panel the profiles along the vertical line at $r=15$.

Similar plots for other quantities reveal trends on the solutions with the increasing magnetic field, stellar rotation rate or dissipation coefficients.

Trends in the “Atlas” results

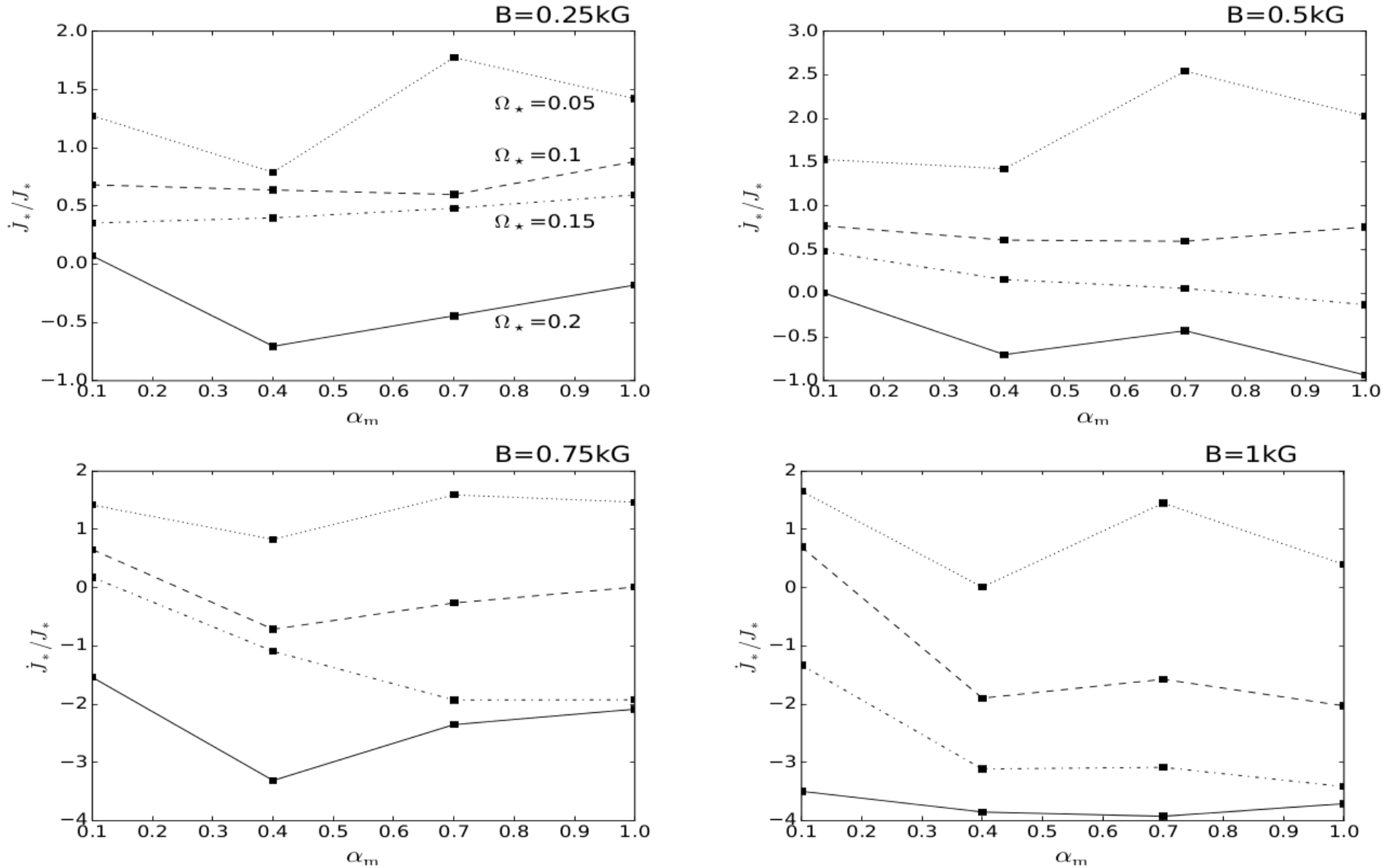


Fig. 4. Average angular momentum flux transported onto the stellar surface by the matter in-falling from the disk onto the star through the accretion column. Each panel shows a set of solutions with one stellar magnetic field strength and varying stellar rotation rate and resistivity. Results with $\Omega_*/\Omega_{\text{br}} = 0.05$ (dotted), 0.1 (dashed), 0.15 (dash-dot-dotted), and 0.2 (solid) are shown in units of stellar angular momentum $J_* = k^2 M_* R_*^2 \Omega_*$ (with $k^2 = 0.2$ for the typical normalized gyration radius of a fully convective star). A positive flux spins the star up, a negative flux slows it down. With the increase in stellar rotation rate, spin-up of the star by the infalling matter decreases and eventually switches to spin-down.

Trends in the “Atlas” results

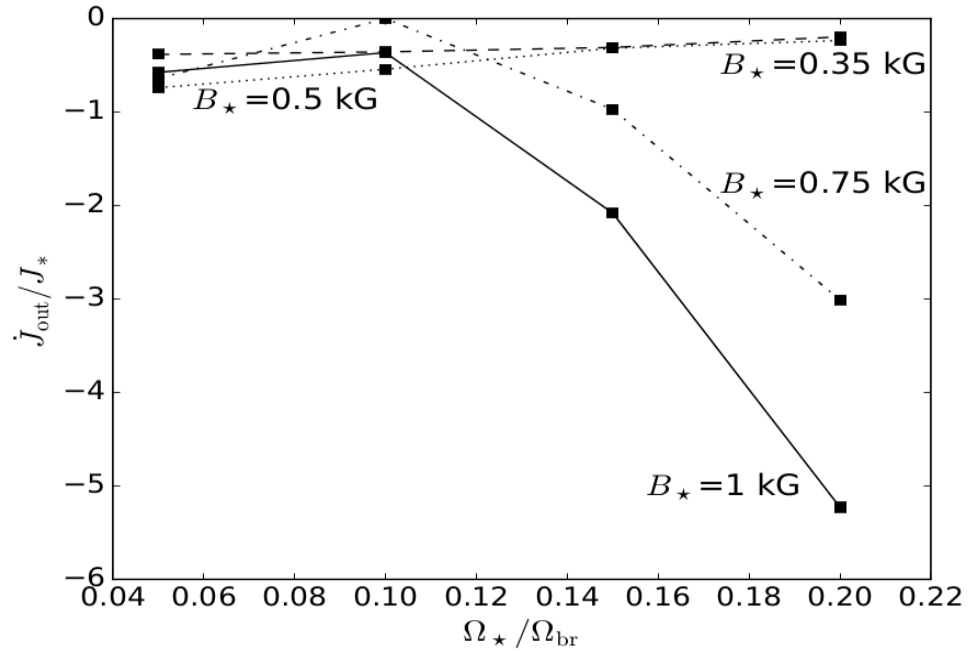
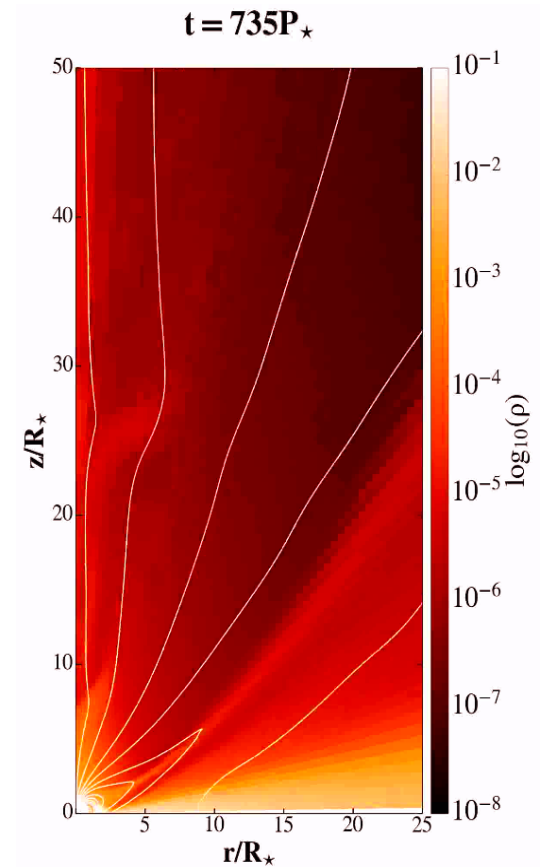
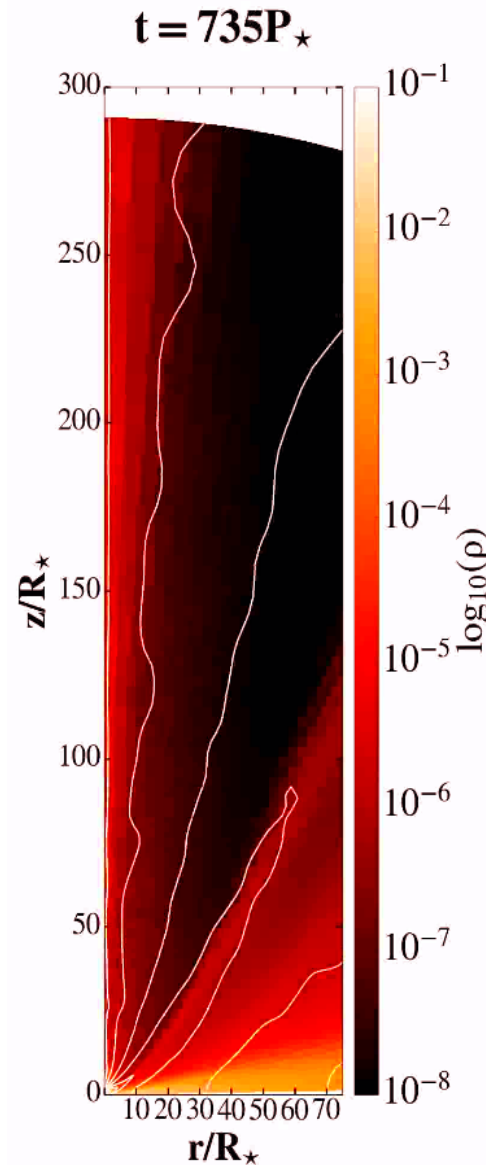
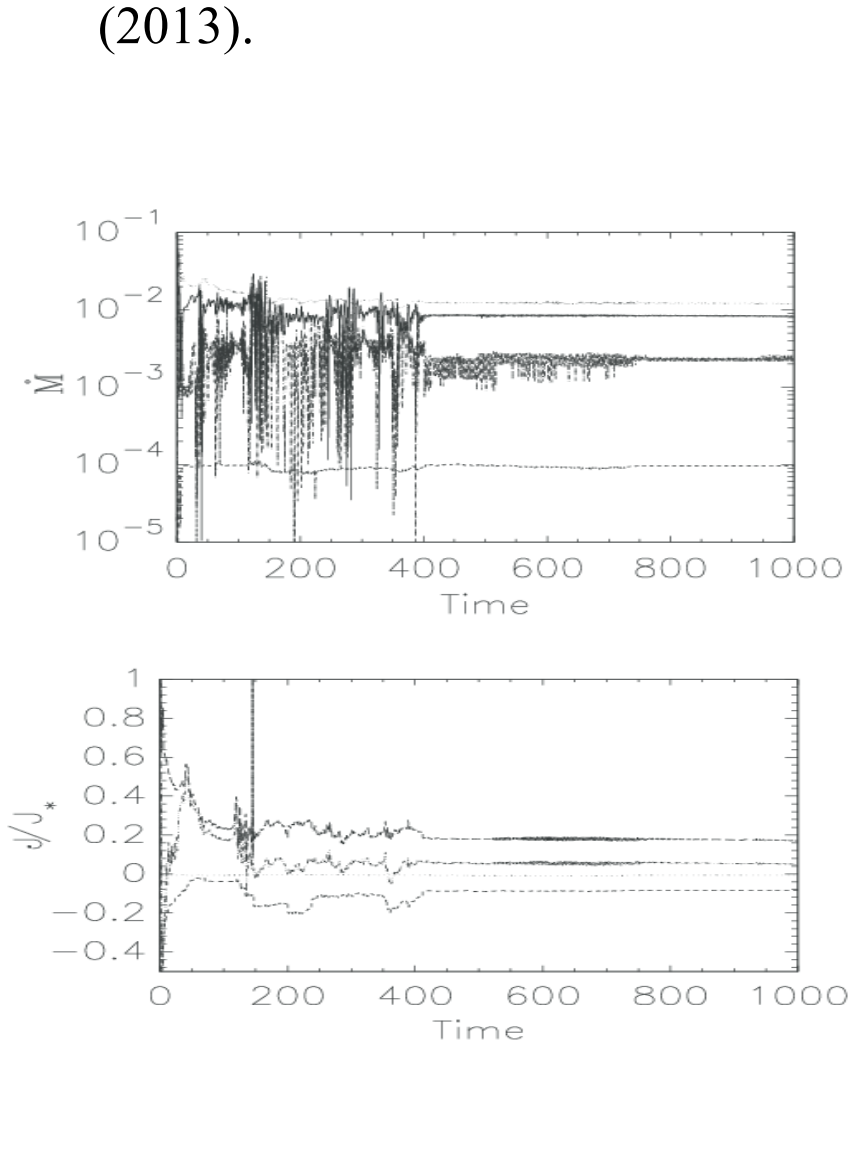


Fig. 5. Average angular momentum flux in the outflow that forms for $\alpha_m = 0.1$. It is computed at $R = 12R_*$ for different stellar rotation rates. Normalization is the same as in Fig. 4. Fluxes in $B_* = 0.25$ (dotted), 0.5 (dash-dotted), 0.75 (dashed) and 1 kG (solid) are shown.

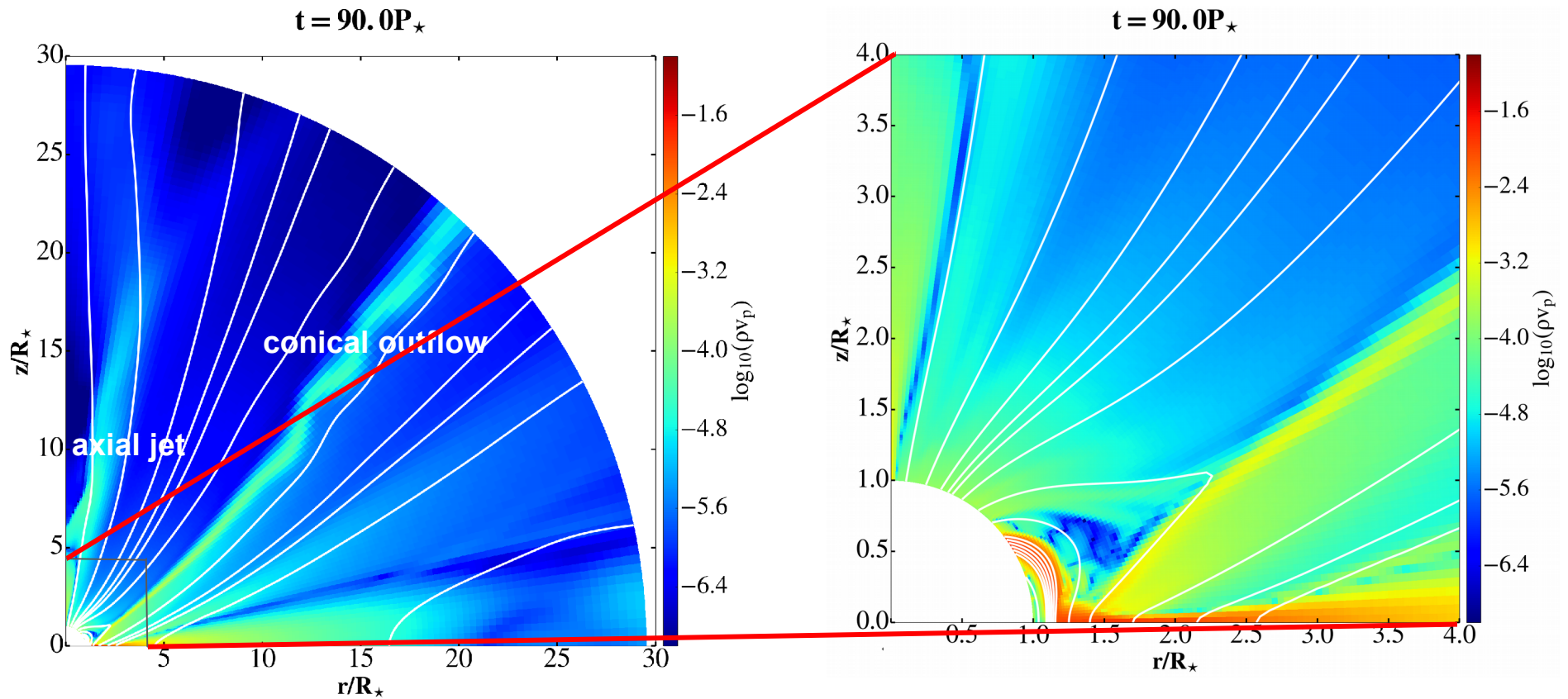
The jet launching

- In a part of the parameter space, there is a continuous launching of an axial jet from the star-disk magnetosphere. In my simulations, it showed that one has to wait until few hundreds of rotations of the star.
- The axial jet and the conical outflow are launched after the relaxation from the initial conditions. They are similar to the results in Romanova et al. (2009) and Zanni & Ferreira (2013).



Zoom into the launching region.

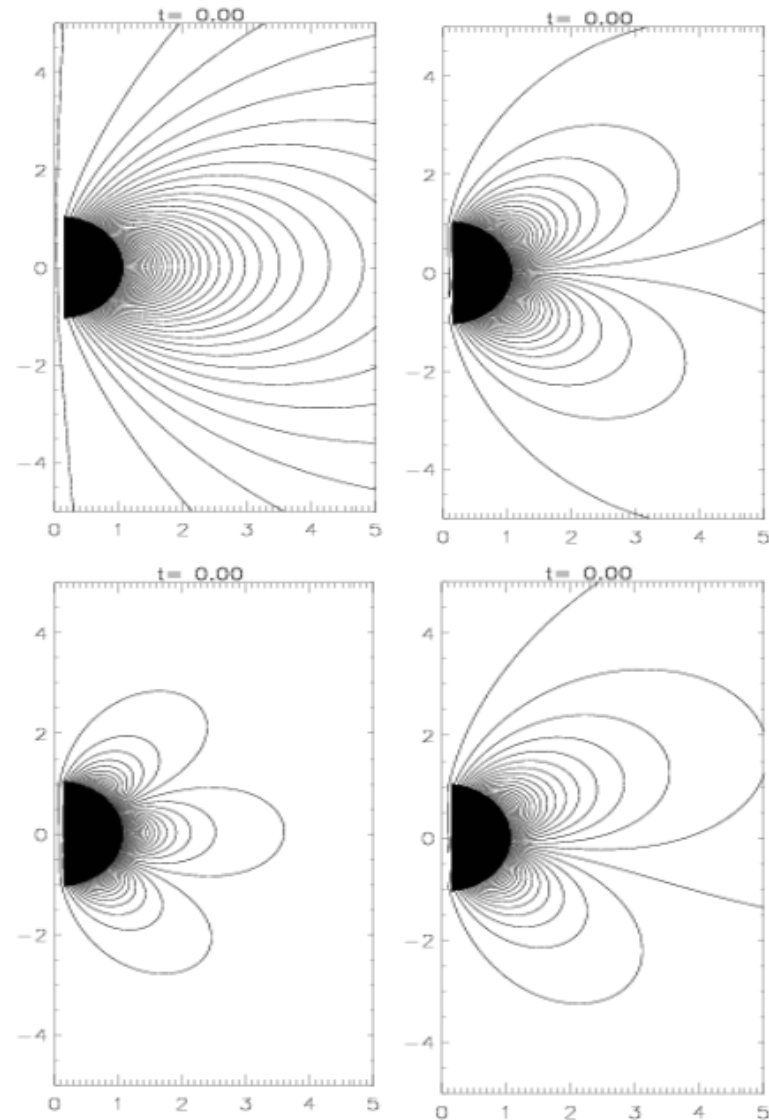
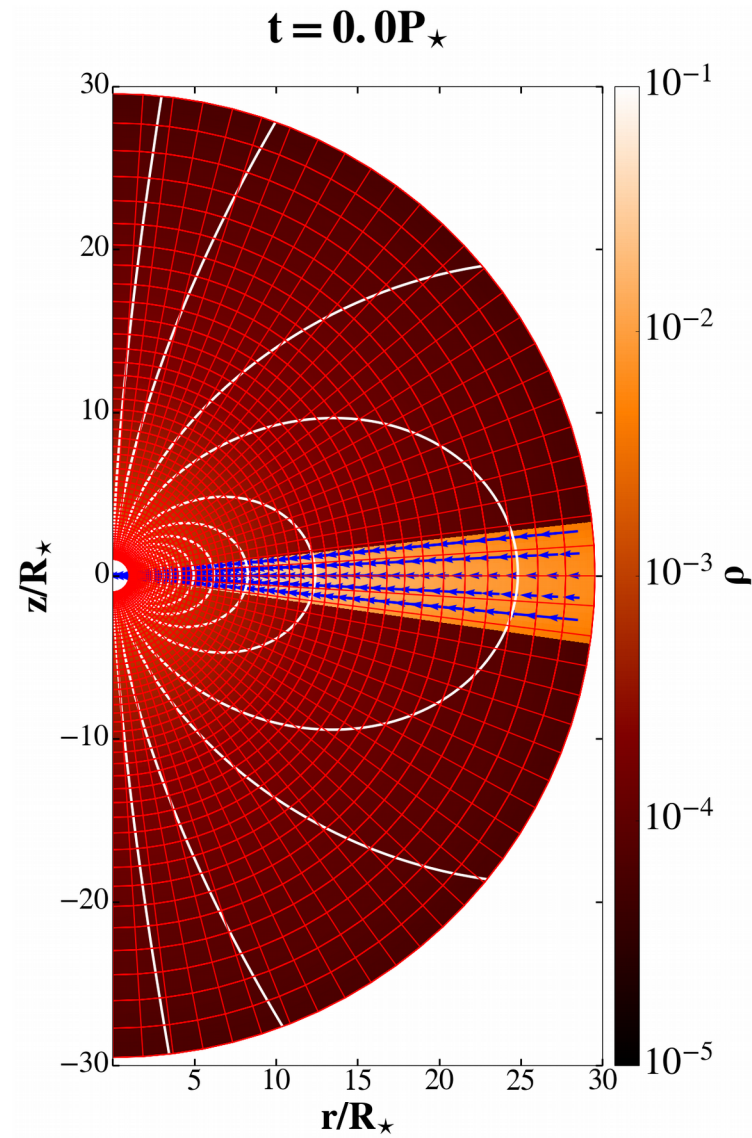
Magnetospheric launching of conical outflow and axial jet



Color shows **momentum** in the MHD simulation in the young stellar object with conical outflow and axial jet, lines are the magnetic field lines. Right panel is a zoom into the left panel close to the star, to show the closest vicinity of the star.

Setup with different geometries of initial magnetic field

- I did also $R \times \vartheta = [217 \times 200]$ grid cells in $\vartheta = [0, \pi]$, and also for different initial geometries of magnetic field: dipole, quadrupole, octupole and mixed multipole.



Results with different geometries of initial magnetic field

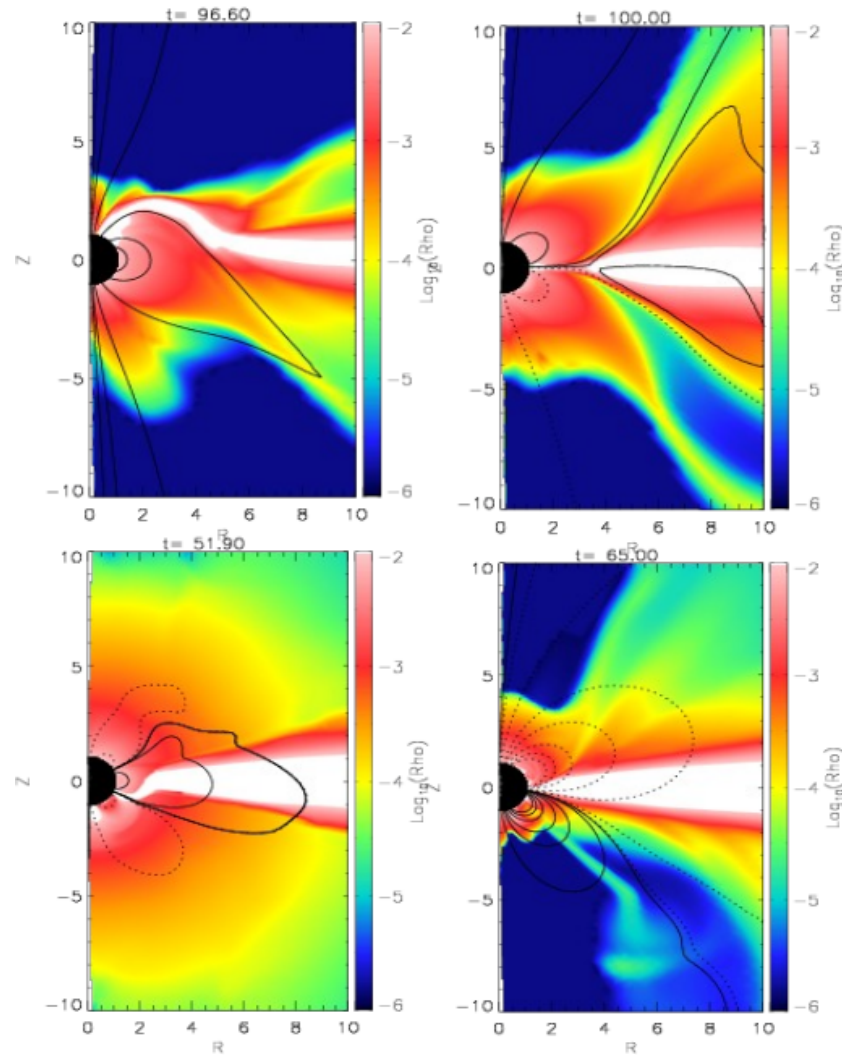


Figure 2: Stable solutions in our SDI simulations, for the different geometries of the magnetic field, with time t given in stellar rotations. The matter density distribution is shown in logarithmic color grading, with the poloidal magnetic field shown in solid and dashed lines, depending on the direction of the field. We sampled the magnetic field lines to leave the density distribution close to the star visible. The results for dipole, quadrupole, octupole and mixed multipole cases are shown from left to right, top to bottom, respectively.

Setup of magnetic runs-changes in pluto.ini and definitions.h

```
I pluto.ini (Modified)(ini) Row 27 Col 19 1:29 Ctrl-K H for help
Solver      hll

[Boundary]
X1-beg      userdef
X1-end      userdef
X2-beg      axisymmetric
X2-end      eqtsymmetric
X3-beg      outflow
X3-end      outflow

[Static Grid Output]
uservar     3 nu num Te
dbl         6.279 -1 single_file
flt         -1.0 -1 single_file
vtk         6.279 -1 single_file
tab         -1.0 -1
dbl.h5      -6.279 -1
ppm         -1.0 -1
png         -1.0 -1
log         100
analysis    -1.0 -1

[Chombo HDF5 output]
Checkpoint_interval -1
Plot_interval      1.0

[Parameters]
ALPHAM           0.4
MU               0.07
```

```
I A definitions.h (c) Row 1 Col 1 1:30 Ctrl-K H for help
#define PHYSICS MHD
#define DIMENSIONS 2
#define COMPONENTS 3
#define GEOMETRY SPHERICAL
#define BODY_FORCE VECTOR
#define FORCED_TURB NO
#define COOLING NO
#define RECONSTRUCTION LINEAR
#define TIME_STEPPING RK2
#define DIMENSIONAL_SPLITTING NO
#define NTRACER 1
#define USER_DEF_PARAMETERS 9

/* -- physics dependent declarations -- */
#define EOS IDEAL
#define ENTROPY_SWITCH NO
#define DIVB_CONTROL CONSTRAINED_TRANSPORT
#define BACKGROUND_FIELD YES
#define AMBIPOLAR_DIFFUSION NO
#define RESISTIVITY NO
#define HALL_MHD NO
#define THERMAL_CONDUCTION NO
#define VISCOSITY EXPLICIT
#define ROTATING_FRAME NO

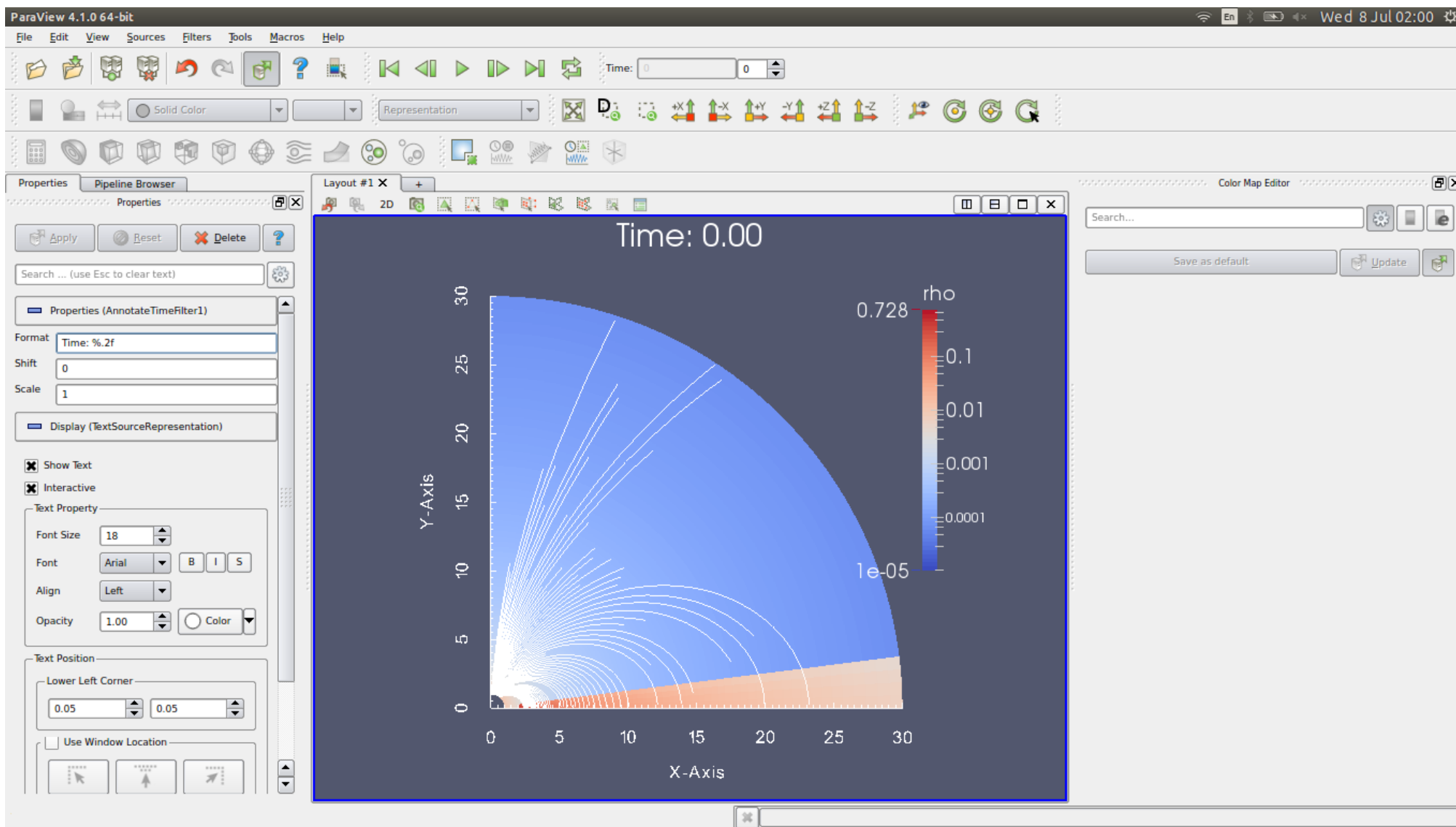
/* -- user-defined parameters (labels) -- */
#define ALPHAM 0
#define MU 1
#define TEMPF 2
#define RHOC 3
#define RD 4
#define EPS 5
```


Setup of magnetic dipole and multipoles in init.c

```
I A init.c (c) void UserDefBoundary (const Data *d, RBox *box, int side, Row 146 Col 1 4:22 Ctrl-K H for help)
#if PHYSICS == MHD
/* ***** */
void BackgroundField (double x1, double x2, double x3, double *B0)
/*
 *
 * PURPOSE
 *
 * Define the component of a static, curl-free background
 * magnetic field.
 *
 *
 * ARGUMENTS
 *
 * x1, x2, x3 (IN) coordinates
 *
 * B0 (OUT) vector component of the background field.
 *
 *
 * ***** */
{
/* dipole */
B0[0] = 2.*g_inputParam[MU]*cos(x2)/(x1*x1*x1);
B0[1] = g_inputParam[MU]*sin(x2)/(x1*x1*x1);
B0[2] = 0.0;
}
#endif
```

```
{
/* dipole */
B0[0] = 2.*g_inputParam[MU]*cos(x2)/(x1*x1*x1);
B0[1] = g_inputParam[MU]*sin(x2)/(x1*x1*x1);
B0[2] = 0.0;
/* */
/* quadrupole
B0[0] = 3.0/2.0*g_inputParam[MU]*(3.0*cos(x2)*cos(x2)-1.0)/(x1*x1*x1*x1);
B0[1] = 3.0*g_inputParam[MU]*cos(x2)*sin(x2)/(x1*x1*x1*x1);
B0[2] = 0.0;
*/
/* octupole
B0[0] = 2.0*g_inputParam[MU]*(5.0*cos(x2)*cos(x2)*cos(x2)-3.0*cos(x2))/(x1*x1*x1*x1*x1);
B0[1] = 0.5*g_inputParam[MU]*(15.0*cos(x2)*cos(x2)*sin(x2)-3.0*sin(x2))/(x1*x1*x1*x1*x1);
B0[2] = 0.0;
*/
}
```

Visualization of Bp lines with Paraview



Outline, Part IV

- Setup of full 3D simulations with dipole magnetic field.
- Restart and initialization from (modified) file.
- Visualization of 3D solutions with Paraview
- Examples of uses of the code:
 - comparison with observations: light curve
 - post-processing (PLUTO + DUSTER): influence of the radiative pressure on the motion of dust grain particles
- Concluding remarks

Restart of simulation from a (modified) file

- Setup of 3d run: just change to 3D in definitions.h (or through python setup) and change the pluto ini grid accordingly in pluto.ini file, also change there the boundary condition (usually “periodic”).
- Restart from a file is one of the options in PLUTO.
- One can start completely new simulation from a (modified) file. Previous result, with some modifications if needed, is now used as an initial condition for new simulation.
- From 2D simulation result, one can create a 3D initial condition, to start the 3D simulation.

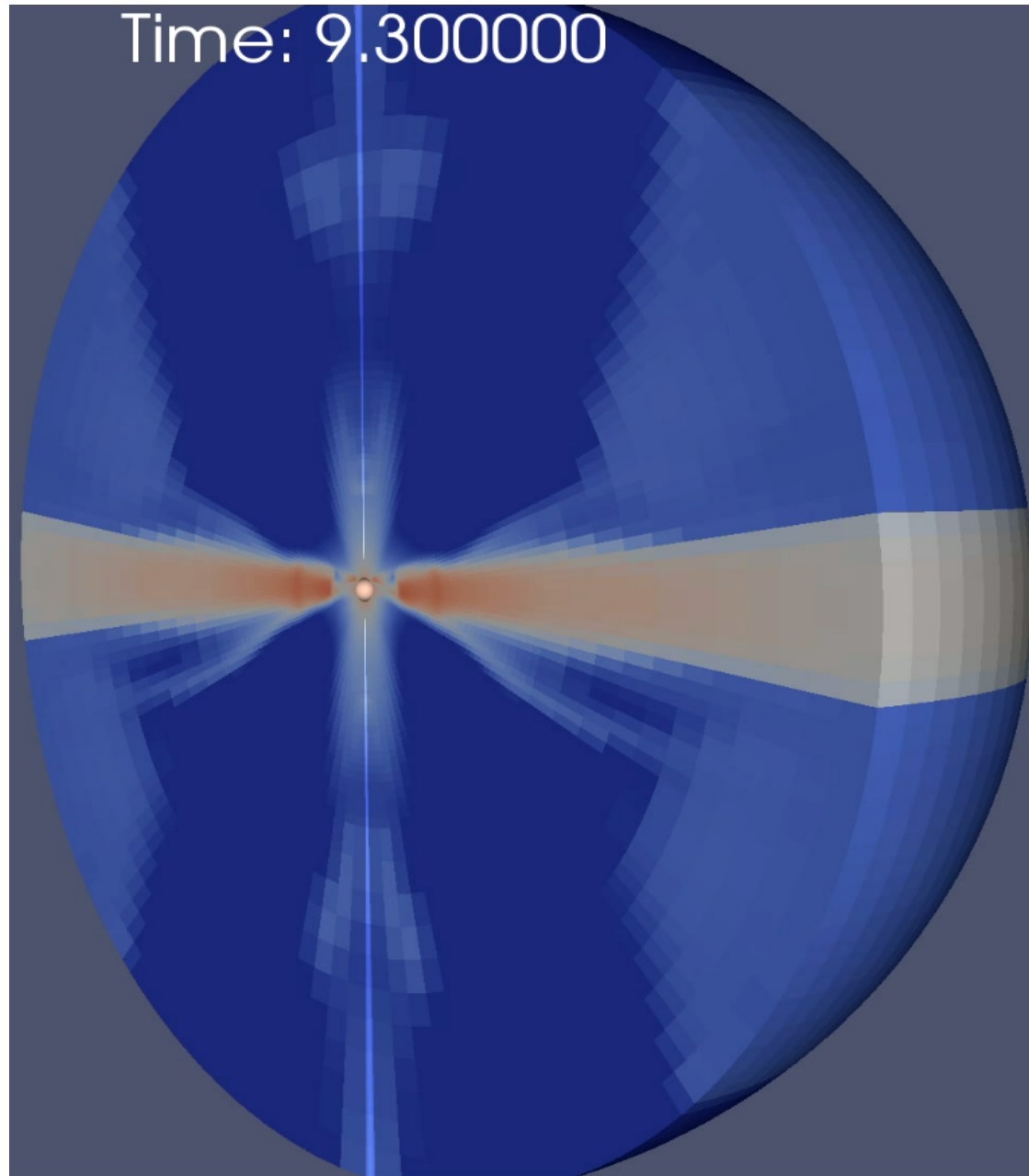
```
I A init.c (c) Row 93 Col 1 2:04 Ctrl-K H for help
/* ***** */
void InitDomain (Data *d, Grid *grid)
/*!
 * Assign initial condition by looping over the computational domain.
 * Called after the usual Init() function to assign initial conditions
 * on primitive variables.
 * Value assigned here will overwrite those prescribed during Init().
 *
 * ***** */
int i,j,k,id;
double *x1 = grid->x[IDIR];
double *x2 = grid->x[JDIR];
double *x3 = grid->x[KDIR];
double r;
double coeff, eps2, pc, rcyl;
double lambda;
double xhi2, Rco;

rcyl=x1[i]*sin(x2[j]);
eps2=g_inputParam[EPS]*g_inputParam[EPS];
coeff=2./5./eps2*(1./x1[i]-(1.-5./2.*eps2)/rcyl);
lambda=11./5./(1.+64./25.*g_inputParam[ALPHAV])*g_inputParam[ALPHAV];

id = InputDataOpen ("./Adata.0045.dbl","grid.out", " ", 0, CENTER);
TOT_LOOP(k,j,i){
d->Vc[RHO][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[PRS][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX1][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX2][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX3][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[BX1][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
}
InputDataClose(id);
/**
TOT_LOOP(k,j,i){
// pc=2./5.*g_inputParam[RHOC]*pow(x1[i],-3./2.);
if (d->Vc[RHO][k][j][i] >= pc) {
// if (d->Vc[TRC][k][j][i] >= 0.5) {
else
d->Vc[RHO][k][j][i] = g_inputParam[RHOC]*pow(x1[i],-3./2.);
d->Vc[PRS][k][j][i] = 2./5.*g_inputParam[RHOC]*pow(x1[i],-5./2.);
d->Vc[VX1][k][j][i] = 0.0;
if (x2[j] < CONST_PI/2./20.) d->Vc[VX1][k][j][i] = 0.001;
d->Vc[VX2][k][j][i] = 0.0;
d->Vc[VX3][k][j][i] = 0.0;
d->Vc[TRC][k][j][i] = 0.0; // Track the corona
}
}
/**
/* ***** */
void Analysis (const Data *d, Grid *grid)
/*
```

```
I A init.c (c) void InitDomain (Data *d, Grid *grid) Row 121 Col 1 2:04 Ctrl-K H for help
d->Vc[RHO][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[PRS][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX1][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX2][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[VX3][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[BX1][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[BX2][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[BX3][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
d->Vc[TRC][k][j][i] = InputDataInterpolate(id,x1[i],x2[j],x3[k]);
}
InputDataClose(id);
/**
TOT_LOOP(k,j,i){
// pc=2./5.*g_inputParam[RHOC]*pow(x1[i],-3./2.);
if (d->Vc[RHO][k][j][i] >= pc) {
// if (d->Vc[TRC][k][j][i] >= 0.5) {
else
d->Vc[RHO][k][j][i] = g_inputParam[RHOC]*pow(x1[i],-3./2.);
d->Vc[PRS][k][j][i] = 2./5.*g_inputParam[RHOC]*pow(x1[i],-5./2.);
d->Vc[VX1][k][j][i] = 0.0;
if (x2[j] < CONST_PI/2./20.) d->Vc[VX1][k][j][i] = 0.001;
d->Vc[VX2][k][j][i] = 0.0;
d->Vc[VX3][k][j][i] = 0.0;
d->Vc[TRC][k][j][i] = 0.0; // Track the corona
}
}
/**
/* ***** */
void Analysis (const Data *d, Grid *grid)
/*
```

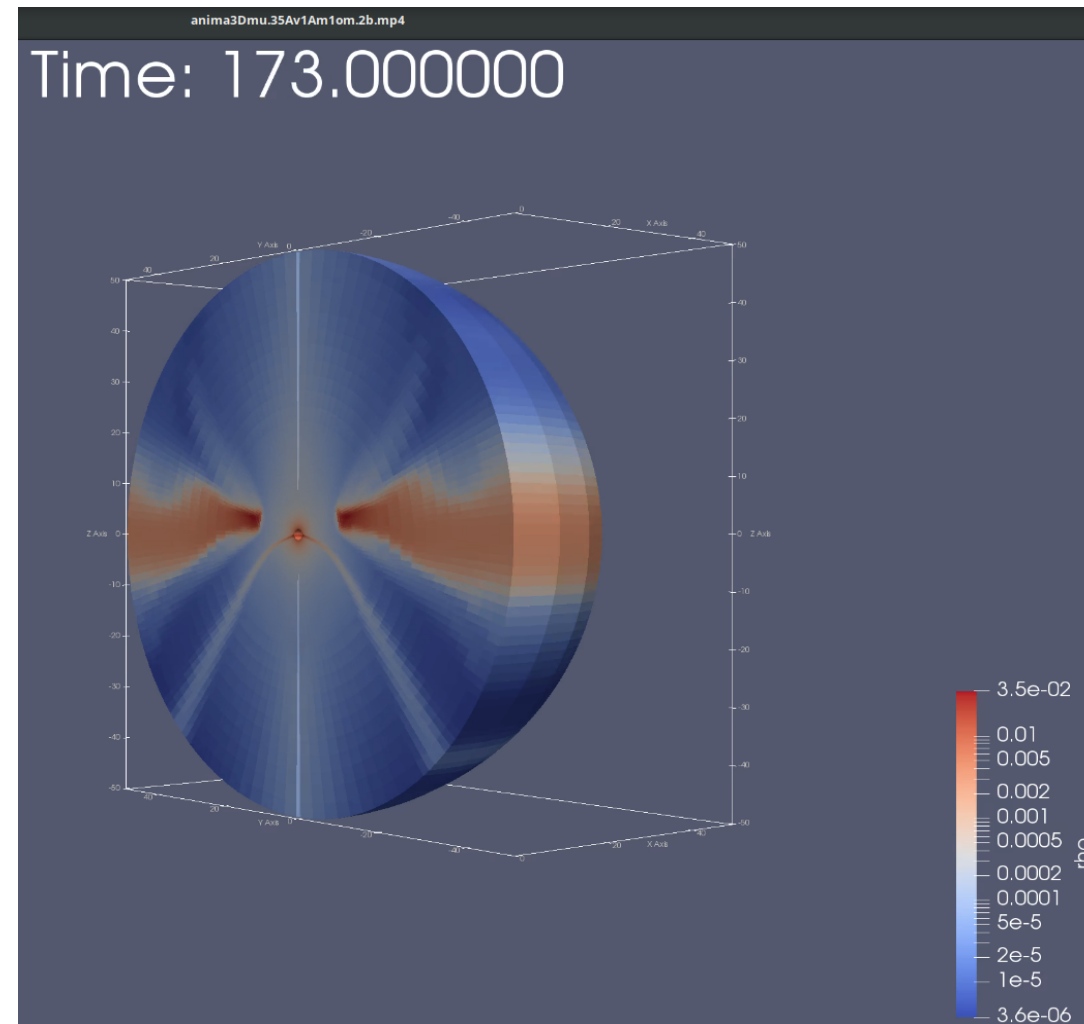
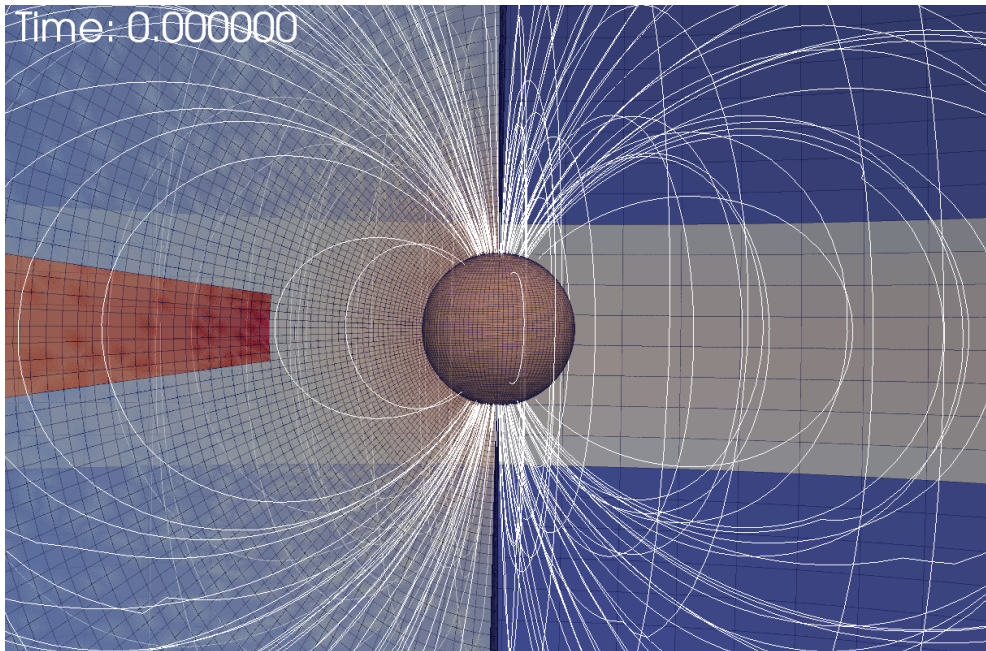
The asymmetric jet launching in 3D simulations



- In the full 3D simulations I obtain asymmetric jets.

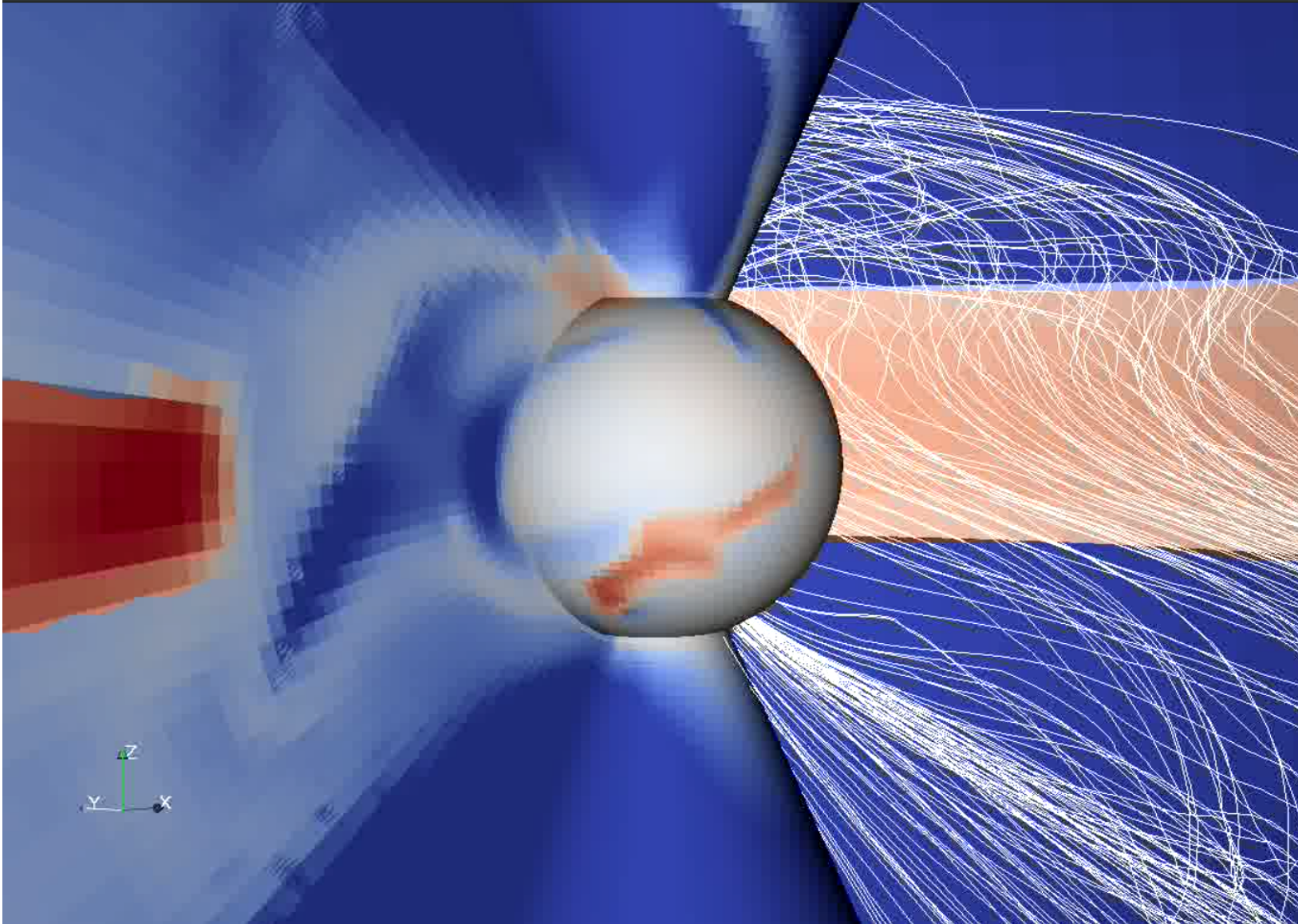
3D simulations, case with magnetic field aligned with rotation

- The first step here is to perform the axisymmetric 3D simulation. I use the spherical grid. The magnetic field in this case is aligned with the rotation axis.
- Zoom into the vicinity of the star=inner boundary condition at $T=0$ and at $T=173$



3D simulations, tilted magnetic field case

- The magnetic field in this case is not aligned with the rotation axis.



Comparison with observations, “hiccups” in light curves

- A curious case with switching of the hemisphere to which the column is attached (Čemeljić & Siwak, 2020).

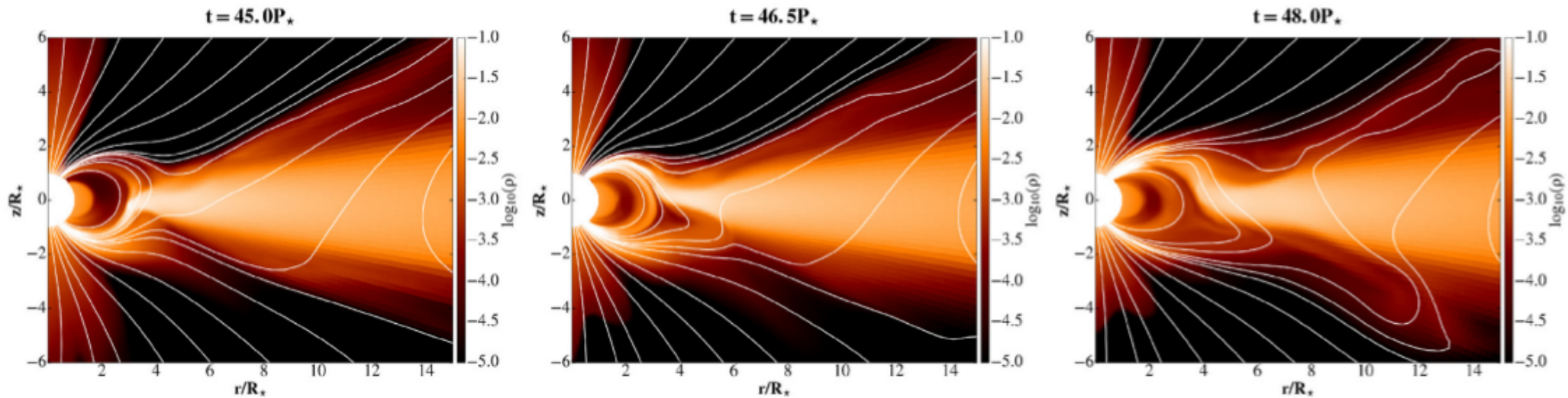


Figure 4. A sequence of snapshots from the results in the interval when switching of the accretion column from the Southern to the Northern hemisphere occurs.

Comparison with observations, “hiccups” in light curves

- Such switching could explain some of the cases of aperiodicity in the Young Stellar Objects.

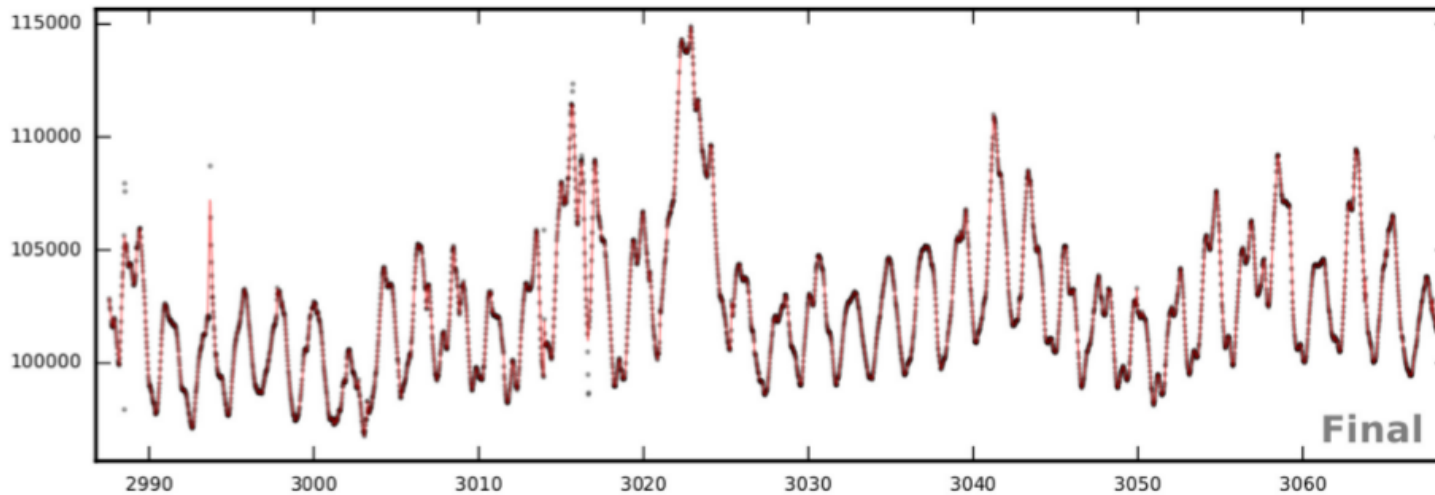
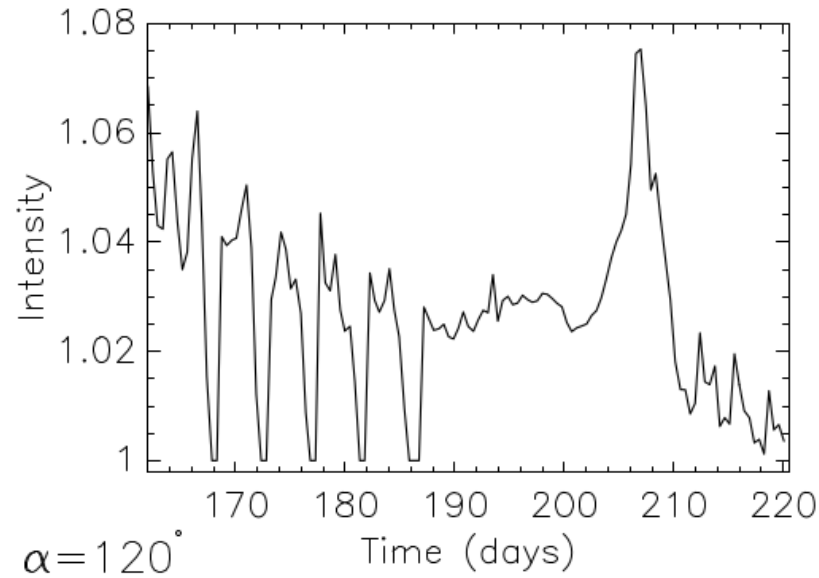
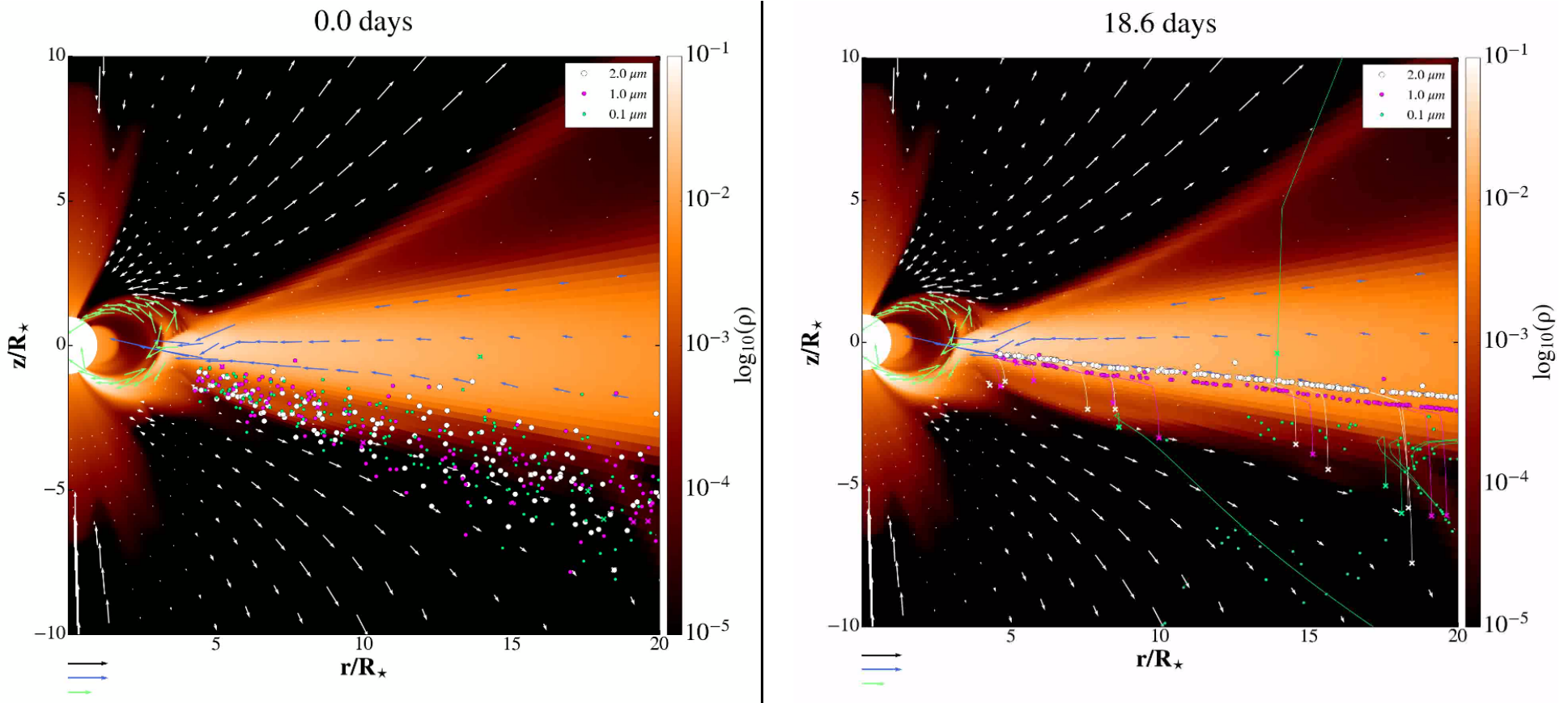


Figure 8. Top panel: intensity in a 3D model made from the longer time sequence than shown in Fig. 7. Shifts in the phase of the observed light (‘hiccups’) occur during the switch of the accretion column from the southern to northern stellar hemisphere between the 190 and 210 d. Bottom panel: light curve from the *Kepler* observation of V1000 Tau in which part of the contribution could occur because of the column switching. Time in the abscissa is annotated in Julian days.

“STARDUST”: Dusty disk in Young Stellar Objects

During a Summer Student Program, C. Turski wrote the Python script “DUSTER” for post-processing of the quasi-stationary results in my star-disk solutions. He added the dust particles and computed their movement in the disk-corona solution as a background. The results are used to model dust distribution in the disk.



$$\ddot{\vec{r}} = -G \frac{M_\star}{r^3} \vec{r} - \frac{\rho_{\text{gas}}}{\rho_{\text{gr}}} \frac{c_s}{a} (\dot{\vec{r}} - \vec{v}_{\text{gas}}) + \beta G \frac{M_\star}{r^2}$$

$$\beta = 0.4 \frac{L_\star}{L_\odot} \frac{M_\odot}{M_\star} \frac{3000 \frac{\text{kg}}{\text{m}^3}}{\rho_{\text{gr}}} \frac{\mu\text{m}}{a}, \quad R_{\text{in}} = 0.0344 \Psi \left(\frac{1500\text{K}}{T_{\text{gr}}} \right)^2 \sqrt{\frac{L_{\text{tot}}}{L_\odot}} [\text{AU}],$$

Summary

-In 4 lectures, I presented the PLUTO code, in hands-on approach.

- We started with general description and thin disk setup in HD, and continued towards MHD solutions for accretion disk.
- I show the setup for viscous and resistive MHD star-disk magnetospheric interaction simulations for a thin accretion disk.
- We learned different ways of visualization, Paraview, VisIt, results with Python
- I detailed the 3D setup of magnetic star-disk simulations.
- Restarting/initializing simulations from a (modified) previous result file was shown.
- We also saw use of the code in post-processing, to create a light curve, compute the dust settling or radiative pressure effects on the dust.