Silesian University in Opava
Institute of Physics in Opava

Thin accretion disks

Part II: Numerical simulations

Dr. Miljenko Čemeljić

Opava, 2022

OPEN UNI - zlepšení otevřenosti a atraktivnosti studia na SU,
CZ.02.2.69/0.0/0.0/18_056/0013364

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Contents

Preface

Observational evidence in favor of accretion disks around newly born stars and many other objects accumulated during the second part of the XX ct. In many cases the disk radial extension is much larger than its height, so that the height to length ratio is at, or less, than 10%. Analytical solution for such disks in the purely hydrodynamic case was given by Kluźniak & Kita (2000). For the magnetic case, the general analytical solution is not possible because of the non-closure of the equations ([ČPK17]), as we explained in the first part of this course. For such cases, we rely on numerical simulations.

In this series of lectures, presented is a numerical simulations setup for a quasi-stationary thin magnetic accretion disk, based on the analytical solution. I detail the setup in both Newtonian and in General Relativistic codes.

Lectures about the Newtonian part of the setup were delivered at the Silesian University in Opava in October 2021 and briefly reviewed in the lectures of the first part of this series, the theoretical introduction during the February/March 2022. The General Relativistic part was reverse mentoring: with students we worked out a transfer of Newtonian setup into the general relativistic code KORAL. Author is grateful to students and the host, Institute of Physics in Opava, for a friendly and motivating environment. The work in Opava was supported by the Czech ESF projects No. CZ.02.2.69/0.0/0.0/18_054/0014696, and author was also funded by a Polish NCN grant No. 2019/33/B/ST9/01564.

PhD students, Debora Lančová at Silesian University in Opava and Fatemeh Kayanikhoo and Angelos Karakonstantakis in Copernicus Center (CAMK) in Warsaw, are thanked for reviewing the final draft of this text. A former PhD student in CAMK Warsaw, David Abarca, is thanked for his October 2021 lectures on KORAL code.

Chapter 1

Thin accretion disk in Newtonian gravity

While gravitational force was extensively studied in the celestial mechanics after Newton, and with the addition of thermodynamics became an essential ingredient for the "recipe" explaining the heat engine of stars, it was not until the second half of XX ct. that accretion was recognized as a most potent energetic process in the Universe, surpassing even the nuclear fission.

A collection of matter onto a rotating central body is a surprisingly complicated process, because without additional action of material forces in addition to gravity, it would produce a catastrophic implosion onto the central object. There would be no stars, galaxies and any sub-structure in the universe. The laws of conservation of momentum, energy and angular momentum themselves do not provide a way to dissipate or transport the angular momentum of the in-falling matter outwards, additional assumptions and mechanisms are needed.

The simplest model was with the dissipation of angular momentum outwards (while the material itself is moving closer towards the gravity center) thanks to the friction of the neighboring layers of material during its in-spiralling onto the central object [SS73; FKR02]. Such a simple model showed not to be efficient enough to prevent the catastrophic outcome, and additional action, usually of the magnetic field, was included.

With the inclusion of magnetic fields, Maxwell equations were added to the already complicated set of fluid equations, and the only way of solving them was to rely on severe approximations, or to (increasingly improving) numerical simulations. About approximations we learned in the first part of this lectures, here we proceed to the numerical simulations. In this chapter, we will show the setup and results with the Newtonian code PLUTO. In the next chapter we will show how we go from this to the general relativistic setup.

The setups and methods we study here are not a historical overview. They are the current state-of-the art material and ongoing research topic, and papers on them are not older then few years, or are still in writing. This means that methods presented here are the best we have up to now, and are subject to constant development.

1.1 Kluźniak-Kita analytical solution

In Kluźniak & Kita (2000) [KK00, hereafter KK00] paper[1], is presented a 3D, axisymmetric, purely hydrodynamic (HD) solution.

[1] It presents most of the PhD thesis of David Kita from 1995 at Madison University, USA [Kit95], which is not available online.

The initial disk was set with the initial density by KK00 with a self-similar profile and the aspect ratio ϵ:

$$\begin{aligned}\rho_{\mathrm{d}} = \rho_{\mathrm{d0}}\left\{\frac{\gamma-1}{\gamma\epsilon^2}\left[\frac{R_*}{R}-\left(1-\frac{\gamma\epsilon^2}{\gamma-1}\right)\frac{R_\star}{R\sin\theta}\right]\right\}^{1/(\gamma-1)} &= \\ = \rho_{\mathrm{d0}}\left\{\frac{2}{5\epsilon^2}\left[\frac{R_\star}{R}-\left(1-\frac{5}{2}\epsilon^2\right)\frac{R_\star}{R\sin\theta}\right]\right\}^{3/2}.\end{aligned} \tag{1.1}$$

The pressure was

$$\begin{aligned}P_{\mathrm{d}} = \epsilon^2\rho_{\mathrm{d0}}v_{\mathrm{K}\star}^2\left(\frac{\rho_{\mathrm{d}}}{\rho_{\mathrm{d0}}}\right)^{\gamma} &= \\ = \epsilon^2\left[\frac{R_\star}{R}-\left(1-\frac{\gamma\epsilon^2}{\gamma-1}\right)\frac{R_\star}{R\sin\theta}\right]^{5/2}.\end{aligned} \tag{1.2}$$

The capital R indicates the spherical radius, and $r = R\sin\theta$ is the cylindrical radius. The disk unit density ρ_{d0} and Keplerian speed $v_{\mathrm{K}\star} = \sqrt{GM_\star/R_\star}$ were both calculated in the disk midplane at $\mathrm{R}_\star$. The initial disk was truncated about the corotation radius.

The obtained disk is a polytropic hydrodynamical solution of the viscous accretion disk in full 3D, obtained by approximate expansion up to the second order in the terms of $\epsilon = c_{\mathrm{s}}/v_{\mathrm{K}}$, the disk aspect ratio measured on the midplane of the disk, where $c_{\mathrm{s}} = \sqrt{\mathrm{P_d}/\rho_{\mathrm{d}}}$ and v_{K} are the isothermal sound speed and Keplerian speed in the disk.

The viscosity was defined by the viscous stress tensor

$$\boldsymbol{\tau} = \eta_{\mathrm{v}}\left[(\nabla\mathbf{v})+(\nabla\mathbf{v})^T-\frac{2}{3}(\nabla\cdot\mathbf{v})\mathbf{I}\right], \tag{1.3}$$

with the dynamic viscosity $\eta_{\mathrm{v}} = \rho\nu_{\mathrm{v}}$ given with

$$\eta_{\mathrm{v}} = \frac{2}{3}\rho\alpha_{\mathrm{v}}\left[c_{\mathrm{s}}^2(r)|_{z=0}+\frac{2}{5}\left(\frac{GM_\star}{R}-\frac{GM_\star}{r}\right)\right]\sqrt{\frac{r^3}{GM_\star}}, \tag{1.4}$$

where ν_{v} is the kinematic viscosity. The magnetic diffusivity was assumed proportional to the viscosity, with the free parameter α_{m}:

$$\nu_{\mathrm{m}} = \frac{3}{2}\alpha_{\mathrm{m}}\frac{\nu_{\mathrm{v}}}{\alpha_{\mathrm{v}}}, \tag{1.5}$$

so that in the cgs system of units the resistivity is $\eta_{\mathrm{m}} = 4\pi\nu_{\mathrm{m}}$.

The diffusive parameters α_{v} and α_{m} were defined in separate subroutines `visc_nu` and `res_eta`. A condition for including the diffusive term was in both routines defined by the $\beta = P_{\mathrm{mag}}/P_{\mathrm{hyd}} > 0.5$, meaning that the magnetic pressure prevailed. In both subroutines, the diffusive term was taken into account only when the tracer value was unity, otherwise it was set to zero.

The initial disk velocity profile is, following KK00:

$$\begin{aligned}v_{Rd} &= -\alpha_{\mathrm{v}}\epsilon^2\left[10-\frac{32}{3}\Lambda\alpha_{\mathrm{v}}^2-\Lambda\left(5-\frac{1}{\epsilon^2\tan^2\theta}\right)\right]\sqrt{\frac{GM_*}{R\sin^3\theta}}, \\ v_{R\varphi} &= \left[\sqrt{1-\frac{5\epsilon^2}{2}}+\frac{2}{3}\epsilon^2\alpha_{\mathrm{v}}^2\Lambda\left(1-\frac{6}{5\epsilon^2\tan^2\theta}\right)\right]\sqrt{\frac{GM_*}{R\sin\theta}}\end{aligned} \tag{1.6}$$

where

$$\Lambda = \frac{11}{5}\Big/\left(1+\frac{64}{25}\alpha_{\mathrm{v}}^2\right). \tag{1.7}$$

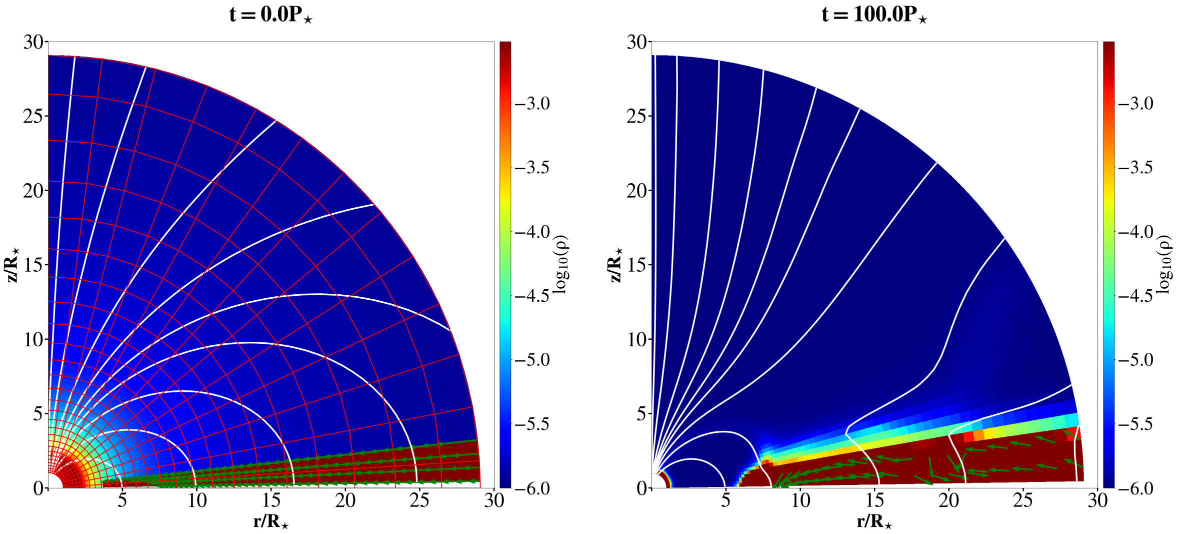

Figure 1.1: *Left panel:* A capture of our initial density distribution with a dipole initial magnetic field lines shown in white solid lines. A sample of initial velocity vectors is also shown. The computational grid is shown with red solid lines in 4 × 4 blocks of cells-in the radial direction the grid cells are logarithmically increasing, while in the polar direction they are uniform. *Right panel*: A snapshot in our simulation at $T = 100P_\star$ with the evolved magnetic field lines and a sample of disk poloidal velocity vectors (normalized to unity). Density is shown in a logarithmic color grading.

The initial corona is a non-rotating, polytropic corona with $\gamma = 5/3$ in hydrostatic balance. It is defined by the density and pressure, given by

$$\begin{aligned}\rho_\mathrm{c} &= \rho_{\mathrm{c}0}(R_\star/R)^{1/(\gamma-1)}, \\ P_\mathrm{c} &= \rho_{\mathrm{c}0}\frac{\gamma-1}{\gamma}\frac{GM_\star}{R_\star}\left(\frac{R_\star}{R}\right)^{\gamma/(\gamma-1)},\end{aligned} \tag{1.8}$$

where $\rho_{\mathrm{c}0} \ll \rho_{\mathrm{d}0}$ is the ratio between the initial coronal and disk density, set to 0.01.

1.2 Equations in the Newtonian PLUTO code

Extensive numerical simulations with a KK00 disk as an initial condition were performed in [Čem19] (see the Appendix there for the details), following [ZF09], using the Newtonian PLUTO code [Mig+07].

We use the KK00 HD solution from the previous section as an initial condition in our simulations, imposing an initial distribution of magnetic field onto it. A code is then solving the hydro-magnetic equations (written here in the cgs units):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1.9}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{1.10}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \left[\rho \mathbf{v}\mathbf{v} + \left(P + \frac{B^2}{8\pi} \right) \tilde{I} - \frac{\mathbf{B}\mathbf{B}}{4\pi} - \tilde{\tau} \right] = \rho \mathbf{g} \tag{1.11}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{\mathbf{B}\mathbf{B}}{8\pi} \right) \mathbf{v} + \underbrace{\eta_{\mathrm{m}} \mathbf{J} \times \mathbf{B}/4\pi - \mathbf{v} \cdot \tilde{\tau}}_{\text{heating terms}} \right] = \rho \mathbf{g} \cdot \mathbf{v} - \underbrace{\Lambda}_{\text{cooling}} \tag{1.12}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times (\mathbf{B} \times \mathbf{v} + \eta_{\mathrm{m}} \mathbf{J}) = 0, \tag{1.13}$$

with ρ, P, $\mathbf{v}$, $\mathbf{B}$ and η_{m} being the density, pressure, velocity, magnetic field and the Ohmic resistivity, respectively. The acceleration of gravity is $\mathbf{g} = -\nabla \Phi_{\mathrm{g}}$, and the gravitational potential of the star with mass $M_\star$ is $\Phi_{\mathrm{g}} = -GM_\star/R$.

The total energy density $E = P/(\gamma - 1) + \rho v^2/2$ and the electric current is given by Ampere's law $\mathbf{J} = \nabla \times \mathbf{B}/(4\pi)$.

An ideal gas is assumed, with the plasma adiabatic index $\gamma = 5/3$, or a polytropic index $n = 3/2$. The unit tensor and the viscous stress tensor, respectively are represented by the terms $\tilde{I}$ and $\tilde{\tau}$. Above the disk surface, set is an initially non-rotating corona in a hydrostatic balance.

1.3 Setup of PLUTO simulations

We present the details in the setup of our non-ideal MHD numerical simulations of a Young Stellar Object, in the physical domain reaching 30 stellar radii, $R_{\mathrm{max}} = 30 R_\star$, with the anomalous[2] viscosity and resistivity parameters $\alpha_{\mathrm{v}} = 1$ and $\alpha_{\mathrm{m}} = 1$. The mass accretion rate in the disk $\dot{M}_0 = 6 \times 10^{-9}\ M_\odot/\mathrm{yr}$.

The stellar rotation rate is taken to be $\Omega_\star = 0.1\ \Omega_{\mathrm{br}}$, where Ω_{br} is the equatorial mass-shedding limit rotation rate, equal to the Keplerian angular velocity for the star $\Omega_{\mathrm{K}\star} = \sqrt{GM_\star/R_\star^3}$. Thus, the corotation radius is $R_{\mathrm{cor}} = (GM_\star/\Omega_\star^2)^{1/3} = (0.2)^{-2/3} R_\star \approx 4.6 R_\star$. In the Classical T-Tauri star case, the stellar mass is $M_\star = 0.5 M_\odot$, radius $R_\star = 2 R_\odot$, the Keplerian velocity at the stellar equator is $v_{\mathrm{K}\star} = 218\ km/s$ and the stellar rotation period is $P_\star = 2\pi/\Omega_\star = 4.64$ days. Then $\rho_{\mathrm{d}0} = 1.2 \times 10^{-10} g/cm^3$. In the magnetic case we add the stellar dipole field of $B_\star = 500\ G$, and the resistivity parameter $\alpha_{\mathrm{m}} = 0.4$, so that the magnetic Prandtl number $P_{\mathrm{m}} = 2\alpha_{\mathrm{v}}/(3\alpha_{\mathrm{m}}) = 0.27$.

The assumed values of anomalous coefficients $\alpha_{\mathrm{v}} = \alpha_{\mathrm{m}}$ are much larger than one would expect in an accretion disk. By such a choice we avoided changes in geometry of the flow: with $\alpha_{\mathrm{v}} < 0.685$ there is a backflow in the disk (see e.g. KK00 and [MČK20b] for a HD, and [MČK20a] for a MHD case), and with smaller values of α_{m} often a conical and/or axial outflows are launched from the magnetosphere [Kot+20]. A Table 1.1 gives rescaling to other types of objects. Since we are using the source term for the cooling function (we used a power law here, with the "cost" constant in the `cooling.c` routine set to a 0.1 of the original value to decrease the computation time. When using cooling source term, the results are NOT rescalable by default. Units in the `definitions.h` routine have to be set correctly in the cgs , to reflect the physical dimensions in the problem. It is done in the following lines in `definitions.h` routine:

```
#define UNIT DENSITY 8.5e-11
#define UNIT LENGTH 1.392e11
#define UNIT VELOCITY 2.1839e7
```

[2] Anomalous diffusive coefficients are much larger than their microscopic equivalent. They are usually given as free parameters in the simulations, assuming that dissipation is a result of turbulence.

Table 1.1: Typical values and scaling for different central objects. The mass M⋆, radius R⋆, period P⋆ and equatorial stellar magnetic field $B_\star$ are chosen to derive the rest of the quantities. The code units should be multiplied by the factors given in the table to apply them to different cases.

	YSOs	WDs	NSs
$M_\star(M_\odot)$	0.5	1	1.4
$R_\star$	$2R_\odot$	5000km	10km
$P_\star$	4.6d	6.1s	0.46ms
$B_\star$ (G)	500	5×10^5	10^8
$\rho_{d0}(\mathrm{g/cm^3})$	1.2×10^{-10}	9.4×10^{-9}	4.6×10^{-6}
v_0(km/s)	218	5150	136000
$\dot{M}_0(M_\odot/yr)$	5.7×10^{-7}	1.9×10^{-9}	10^{-9}
B_0(G)	200	5×10^4	2.93×10^7

The initial stellar magnetic dipole field was set with the field axis aligned with the stellar rotation axis. There is no resistivity in the magnetosphere outside of the disk. This means that the reconnection of the magnetic field is a numerical, not physical dissipation.

1.4 Modifications in PLUTO environment

The code we use is PLUTO v.4.4.2 ([Mig+07]). There are some modifications which need to be done in the public version of the code. Those changes are done by copying files mentioned below from the version in the `PLUTO/Src` subdirectory in the PLUTO main distribution version, into the work directory (the one from which the code is run, where `init.c` and `pluto.ini` files reside). The modified version of any `*.c` file in the work directory has, by default, priority to the original version in `PLUTO/Src` directory. The rule for `*.h` files is that they can be modified directly in `PLUTO/Src`, only `definitions.h` file can be modified in working directory.

The files to be copied into the work directory are, in alphabetic order: `boundary.c, cooling.c, ct_update.c, init.c, res_eta.c, visc_nu.c, write _vtk.c`.

The modifications are shortly described in commented sections in each of the files, below is given a short description of changes by positions in the subroutine, or, in the cases where the default subroutine in PLUTO is empty, as in the cases of `visc_nu.c` and `res_eta.c`, a complete subroutine text.

1.4.1 Modifications in `*.c` subroutines:

-In subroutine `init.c` the actual setup is given. We copy here the complete text of the subsequent part of this subroutine, with abbreviated comments:
The initial purely hydrodynamical solution for the disk by KK00 is set, with a non-rotating corona in hydrostatic equilibrium above it:
`void Init (double *v, double x1, double x2, double x3)` part:

```
    double coeff, eps2, pc, rcyl;
double br,bth;
double lambda;
double xhi2, Rco;

    rcyl=x1*sin(x2);
```

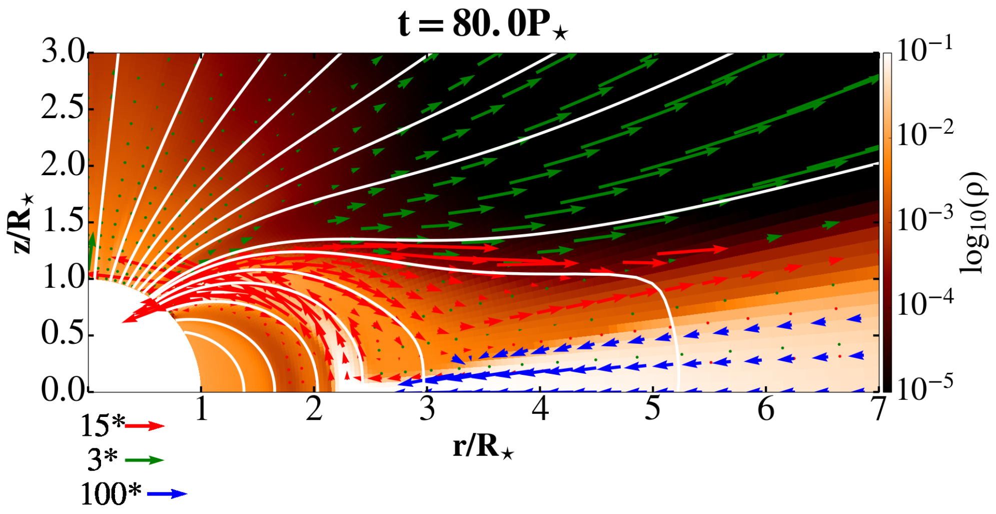

Figure 1.2: A zoom into our simulation result from [Čem19] after T=80 rotations of the underlying star. Shown is the density in logarithmic color grading in code units, with a sample of magnetic field lines, depicted with white solid lines. Velocities in the disk, column and stellar wind are shown with vectors, depicted in different normalizations with respect to the Keplerian velocity at the stellar equator.

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

    /* initial non-rotating adiabatic corona in hydrostatic equilibrium */
v[RHO] = g_inputParam[RHOC]*pow(x1,-3./2.);
v[PRS] = 2./5.*g_inputParam[RHOC]*pow(x1,-5./2.);

    pc=v[PRS];

    v[VX1] = 0.0;
v[VX2] = 0.0;
v[VX3] = 0.0;

    /* Keplerian adiabatic disk in vertical pressure equilibrium with the adiabatic corona, as given by Kluzniak & Kita (2000) */

    v[PRS]=eps2*pow(coeff,5./2.);

    if (v[PRS] >= pc && rcyl > g_inputParam[RD])
v[RHO] = pow(coeff,3./2.);
v[VX1] = -g_inputParam[ALPHAV]/sin(x2)*eps2*(10.-32./3.
*lambda*g_inputParam[ALPHAV]*g_inputParam[ALPHAV]
-lambda*(5.-1./(eps2*tan(x2)*tan(x2))))/sqrt(rcyl);
v[VX3] = (sqrt(1.-5./2.*eps2)+2./3.*eps2
*g_inputParam[ALPHAV]*g_inputParam[ALPHAV]
*lambda*(1.-6./(5.*eps2*tan(x2)*tan(x2))))/sqrt(rcyl);
```

```
v[TRC] = 1.0; /* Track the disk material */

else

v[PRS]=2./5.*g_inputParam[RHOC]*pow(x1,-5./2.);
v[TRC] = 0.0; /* Track the corona */

    #if PHYSICS == MHD
v[BX1] = 0.0;
v[BX2] = 0.0;
v[BX3] = 0.0;

    v[AX1] = 0.0;
v[AX2] = 0.0;
v[AX3] = 0.0;

    #endif
```

For the magnetic field, a split-field method is used, so that we evolve in time only changes from the initial stellar magnetic field [Tan94; Pow+99], with the constrained transport method used to maintain the $\nabla \cdot \mathbf{B} = 0$:
void BackgroundField (double x1, double x2, double x3, double *B0) part:

```
    /* 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;
```

In the **internal boundary** part (`side == 0`) loop at the beginning of the `UserDefBoundary` subroutine in `init.c` file), which enables change in the quantities inside the computational domain in PLUTO, the density in the grid cell just above the star was corrected when it falls below some limit value (we set it, at the end of the `pluto.ini` user defined parameters part, to 5×10^{-7}), to prevent the density near the star from becoming too low. The pressure was corrected in such a way to conserve the sound speed in the corona. To maintain the self-consistency, **velocities also need to be changed, to conserve the momentum**[3]. The energy is not conserved, but it is of minor concern here, since it is anyway dissipated.

We also ensured that the scalar tracer value was always set to zero in the corona. Around the reconnection sheet and outflows, the tracer scalar can obtain spurious values, so this is prevented here.

The numerical heating in the corona was prevented by enforcing the conservation of the entropy S, to keep the values close to the initial conditions. The maximum of the entropy is on top of the star, so it was kept throughout the computational box below the value $S_{max} = P_c/\rho_c^\gamma = 2\rho_{c0}^{-2/3}/(5R_*) = 8.61774$. For the minimum, the small number $S_{min} = 0.01$ was set. From this we obtained corrected values for the pressure interval in the computational box, by $P = \max[\min(P, S_{max}\rho^\gamma), S_{min}\rho^\gamma]$:

```
    void UserDefBoundary (const Data *d, RBox *box, int side, Grid *grid) part:
if (side == 0)
```

[3] We stress this part of the setup, as it is often overlooked, and introduces the physical inconsistency in the code. Whenever a floor value for some quantity is used, one should ensure the conservation laws are satisfied, if possible.

```
/* – Impose conditions for the solution inside domain near the star – */

    x1 = grid−>xgc[IDIR];
x2 = grid−>xgc[JDIR];
x3 = grid−>xgc[KDIR];

    /* Avoid too small density near the star at the beginning and correct
the pressure to conserve the same sound speed for material in corona */
DOM_LOOP(k,j,i)
//ccm−here set the rho(x1[i]) <= rho one cell above the star
if (x1[i] <= x1[IBEG+1] && d−>Vc[RHO][k][j][i]< g_inputParam[DFLOOR])
//ccm−save values of rho and cs2 before reset to dfloor
dden=d−>Vc[RHO][k][j][i];
cs2=g_gamma*d−>Vc[PRS][k][j][i]/d−>Vc[RHO][k][j][i];
d−>Vc[RHO][k][j][i]=g_inputParam[DFLOOR];
dfact=dden/d−>Vc[RHO][k][j][i];

    //ccm−modify prs so that we stay with the same cs
d−>Vc[PRS][k][j][i]=cs2*d−>Vc[RHO][k][j][i]/g_gamma;
//ccm−to conserve momentum & energy, modify velocities
d−>Vc[VX1][k][j][i]=dfact*d−>Vc[VX1][k][j][i];
d−>Vc[VX2][k][j][i]=dfact*d−>Vc[VX2][k][j][i];
d−>Vc[VX3][k][j][i]=dfact*d−>Vc[VX3][k][j][i];

//ccm−ensure that TRC=0.0 in corona. Around reconnection region and outflows
// sometimes TRC obtains spurious values, this is to prevent it.
// if(x2[j]<0.5*CONST_PI-atan(1.25*g_inputParam[EPS]) && // d−>Vc[RHO][k][j][i]<
if(x2[j]<0.5*CONST_PI-atan(3.*g_inputParam[EPS]) && d−>Vc[RHO][k][j][i]<
1.e3*g_inputParam[DFLOOR])d−>Vc[TRC][k][j][i]=0.0;

    /* Avoid numerical heating in the corona */
//ccm−smax is entropy at the stellar surface, smin is just a small number
smax=8.61774;
smin=0.01;
d−>Vc[PRS][k][j][i]=MAX(MIN(d−>Vc[PRS][k][j][i],smax*pow(d−>Vc[RHO][k][j][i],g_gamma))
,smin*pow(d−>Vc[RHO][k][j][i],g_gamma));
```

In the **inner boundary conditions** (`side == X1_BEG`.) loop, which are atop the stellar surface in our setup, the density, pressure and toroidal components of the velocity and magnetic field were prescribed from the active zones into the boundary. The linear extrapolation was used, with Van Leer limiter in the density and the magnetic field, and `minmod` limiter in the pressure and velocity.

For the numerical stability in the corona, in the cases with $v_R > 0$, we introduced a correction of the pressure by a free parameter T_f in the inner radial boundary condition. It should be set to the number of order a few hundred. It adjusts the ram pressure ρv^2 on top of the star (stellar wind) from the default $T_f = 2v_R^2/5$ to $T_f = (2/5 - T_f)v_R^2$ so that the pressure in the corona is given by

$$P = (2/5 - T_f v_R^2)\rho_c R_i^{-5/2}. \tag{1.14}$$

The first term here is the coronal initial pressure $P_c = 2\rho_c R_i^{-5/2}/5$. Because $P \sim \rho T$ in the ideal gas law, we can think of 2/5 as an effective temperature, with T_f an ad hoc correction.

In the axisymmetric 2D setup, since we use the constrained transport method in which a staggered grid is used, the toroidal component of the magnetic field was set in the staggered grid.

We used a custom boundary condition for B_φ, in which $\Omega_{\rm eff} = \Omega - \rm v_p B_\varphi/(rB_p)$ is derived from the condition for the stellar surface as a rotating perfect conductor[4]. The magnetic torque to drive the plasma rotation on top of the star was set, with the matching measured by the comparison of the stellar angular velocity and the effective rotation rate of the field lines by the ratio $\Omega_{\rm eff}/\Omega_*$. This prescribes rotation of the matter on top of the star and the effective rotation rate of the field lines to $\Omega_{\rm eff}$, with v_p and B_p standing for the poloidal velocity and magnetic field, respectively. In addition, in the constrained transport method, in the subroutine `ct_update`, we set the toroidal component of the EMF to zero. Then in the stellar reference frame, the electric field is zero, with the flow speed parallel to the magnetic field. The stellar rotation rate was set with those corrections.

We have further:

```
    if (side == X1_BEG) /* – X1_BEG boundary – */
r = grid−>x[IDIR];
rr= grid−>xr[IDIR];
rl= grid−>xl[IDIR];
thr=grid−>xr[JDIR];
thl=grid−>xl[JDIR];
x1 = grid−>x[IDIR];
x2 = grid−>x[JDIR];
x3 = grid−>x[KDIR];

    /* ccm1214−in use of limiters I follow ZF09, using VanLeer
for rho and B, and MinMod for pressure and v */
if (box−>vpos == CENTER)
//ccmt−reverting to old loop structure, as new one misses a cell at jmin
//ccm BOX_LOOP(box,k,j,i)
X1_BEG_LOOP(k,j,i)
/*Linear extrapolation of rho */

    if (d−>Vc[TRC][k][j][IBEG]> 0.01 && d−>Vc[VX1][k][j][IBEG]<0.0)
dvar1dr=(d−>Vc[RHO][k][j][IBEG+2]-d−>Vc[RHO][k][j][IBEG+1])/(r[IBEG+2]-r[IBEG+1]);
dvar2dr=(d−>Vc[RHO][k][j][IBEG+1]-d−>Vc[RHO][k][j][IBEG])/(r[IBEG+1]-r[IBEG]);
dvardr=VANLEER_LIMITER(dvar1dr,dvar2dr);
// dvardr=MINMOD(dvar1dr,dvar2dr);
dvardr=MIN(dvardr,0.0);
d−>Vc[RHO][k][j][i] = d−>Vc[RHO][k][j][i+1]-dvardr*(r[i+1]-r[i]);
d−>Vc[PRS][k][j][i] = d−>Vc[PRS][k][j][i+1]
*pow(d−>Vc[RHO][k][j][i]/d−>Vc[RHO][k][j][i+1],g_gamma);
d−>Vc[TRC][k][j][i] =d−>Vc[TRC][k][j][IBEG];

else
```

[4] This is a novel feature, introduced in [ZF09] (see the Appendix there for the details on the effect of this correction), which enables additional stability of the solution near the surface of the central object.

```
d−>Vc[RHO][k][j][i] = g_inputParam[RHOC]*pow(r[i],-3./2.);
//ccm−put tempf in pluto.ini about few hundreds. It is so
// called effective temperature from polytropic law theory.
temp=2./5.;
if(d−>Vc[VX1][k][j][IBEG]>0.0) temp -= g_inputParam[TEMPF]
*d−>Vc[VX1][k][j][IBEG]*d−>Vc[VX1][k][j][IBEG];
d−>Vc[PRS][k][j][i] = temp * g_inputParam[RHOC]*pow(r[i],-5./2.);
d−>Vc[TRC][k][j][i] = 0.0;


    #if PHYSICS == MHD
/* Toroidal magnetic field BC */
d−>Vc[BX3][k][j][i] =d−>Vc[BX3][k][j][i+1]*x1[i+1]/x1[i]; //ala Romanova //
// d−>Vc[BX3][k][j][i] = 0.; // steady-state BC

    d−>Vc[VX1][k][j][i]=d−>Vc[VX1][k][j][IBEG];
d−>Vc[VX2][k][j][i]=d−>Vc[VX2][k][j][IBEG];

    //ccmt−at the end, set the stellar rotation, including E_phi=0 condition.
//Do not forget to set E_phi=0 in ct.c routine, too.
/* BC for Vphi */
BackgroundField (x1[i],x2[j],x3[k],Bgji);
d−>Vc[VX3][k][j][i] = g_inputParam[OMG]*x1[i]*sin(x2[j])
+(d−>Vc[VX1][k][j][i]*(Bgji[0]+d−>Vc[BX1][k][j][i])
+d−>Vc[VX2][k][j][i]*(Bgji[1]+d−>Vc[BX2][k][j][i]))
*(d−>Vc[BX3][k][j][i])/((Bgji[0]+d−>Vc[BX1][k][j][i])
*(Bgji[0]+d−>Vc[BX1][k][j][i])
+(Bgji[1]+d−>Vc[BX2][k][j][i])*(Bgji[1]+d−>Vc[BX2][k][j][i]));

    #endif

else if (box−>vpos == X2FACE)
#ifdef STAGGERED_MHD
BOX_LOOP(box,k,j,i) d−>Vs[BX2s][k][j][i] = d−>Vs[BX2s][k][j][IBEG];
#endif
else if (box−>vpos == X3FACE)
#ifdef STAGGERED_MHD
//ccm-not called in 2D, but yes in 3D, then def. smthing as e.g. Romanova
//ccm or the more complicated bphi condition. You have to include it here,
//not in cell centered part.
// BOX_LOOP(box,k,j,i) d−>Vs[BX3s][k][j][i] = d−>Vs[BX3s][k][j][IBEG];
#endif
```

In the **outer boundary conditions** in the (`side == X1_END`) part in the further loop in the coronal part of the domain we set the logarithmic extrapolation in the density and pressure. In the radial and meridional components of the velocity an outflow was set, and a linear extrapolation with the `minmod` limiter for the azimuthal velocity component and Van Leer limiter in the toroidal magnetic field component.

At the disk outer radial boundary, the initial hydrodynamical values are inserted, anticipating the thickening of the disk for 25%. Because the velocity might roll back above the thickened disk, inflow to twice the initial disk height in the corona is prevented by setting the velocities to zero. The toroidal magnetic field is linearly extrapolated with a Van Leer limiter.

```
    if (side == X1_END) /* – X1_END boundary – */

    r = grid−>x[IDIR];
x1 = grid−>x[IDIR];
x2 = grid−>x[JDIR];
x3 = grid−>x[KDIR];

    if (box−>vpos == CENTER)
BOX_LOOP(box, k, j, i)
d−>Vc[TRC][k][j][i] =d−>Vc[TRC][k][j][IEND];

    /* Logarithmic extrapolation of rho, prs */
a1=log10(d−>Vc[RHO][k][j][IEND]/d−>Vc[RHO][k][j][IEND-1])/log10(r[IEND]/r[IEND-1]);
a2=log10(d−>Vc[RHO][k][j][IEND-1]/d−>Vc[RHO][k][j][IEND-2])/log10(r[IEND-1]/r[IEND-2]);
a=VANLEER_LIMITER(a1,a2);
// a=MINMOD(a1,a2);
a=MIN(a,0.0);
d−>Vc[RHO][k][j][i] = d−>Vc[RHO][k][j][i-1]*pow(r[i]/r[i-1],a);
d−>Vc[PRS][k][j][i] = d−>Vc[PRS][k][j][IEND]
*pow(d−>Vc[RHO][k][j][i]/d−>Vc[RHO][k][j][IEND],g_gamma);

    /* Ouflow BC for poloidal velocities */
d−>Vc[VX1][k][j][i] = d−>Vc[VX1][k][j][IEND];
d−>Vc[VX2][k][j][i] = d−>Vc[VX2][k][j][IEND];

    #if PHYSICS == MHD
dvar1dr=(d−>Vc[BX3][k][j][IEND]-d−>Vc[BX3][k][j][IEND-1])/(r[IEND]-r[IEND-1]);
dvar2dr=(d−>Vc[BX3][k][j][IEND-1]-d−>Vc[BX3][k][j][IEND-2])/(r[IEND-1]-r[IEND-2]);
dvardr=VANLEER_LIMITER(dvar1dr,dvar2dr);
// dvardr=MINMOD(dvar1dr,dvar2dr);
d−>Vc[BX3][k][j][i] = d−>Vc[BX3][k][j][i-1]+dvardr*(r[i]-r[i-1]);

    #endif
/* Linear extrapolation of Vphi */
dvar1dr=(d−>Vc[VX3][k][j][IEND]-d−>Vc[VX3][k][j][IEND-1])/(r[IEND]-r[IEND-1]);
dvar2dr=(d−>Vc[VX3][k][j][IEND-1]-d−>Vc[VX3][k][j][IEND-2])/(r[IEND-1]-r[IEND-2]);
dvardr=MINMOD_LIMITER(dvar1dr,dvar2dr);
d−>Vc[VX3][k][j][i] = d−>Vc[VX3][k][j][i-1]+dvardr*(r[i]-r[i-1]);

    //ccm−at the disk outer boundary in R, set the initial HD values
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);
```

```
coeff = MAX(coeff,0.0);
lambda=11./5./(1.+64./25.*g_inputParam[ALPHAV]*g_inputParam[ALPHAV]);

    //ccm−here we assume disk will puff-up for 25%, so we include this at b.c.
//ccm−for [0,pi/2] case use only the line below:
if (x2[j] >= 0.5*CONST_PI-atan(1.25*g_inputParam[EPS]))
d−>Vc[RHO][k][j][i] = pow(coeff,3./2.);
if(d−>Vc[RHO][k][j][i] == 0.0) d−>Vc[RHO][k][j][i] = d−>Vc[RHO][k][j][IEND];
d−>Vc[PRS][k][j][i] = eps2*pow(coeff,5./2.);

d−>Vc[VX1][k][j][i] = -g_inputParam[ALPHAV]/sin(x2[j])
*eps2*(10.-32./3.*lambda*g_inputParam[ALPHAV]*g_inputParam[ALPHAV]
-lambda*(5.-1./(eps2*tan(x2[j])*tan(x2[j]))))/sqrt(rcyl);
d−>Vc[VX3][k][j][i] = (sqrt(1.-5./2.*eps2)+2./3.*eps2
*g_inputParam[ALPHAV]*g_inputParam[ALPHAV]
*lambda*(1.-6./(5.*eps2*tan(x2[j])*tan(x2[j]))))/sqrt(rcyl);

//ccm−velocity could roll back even higher than the disk will pile-up,
// so we prevent inflow in the corona from even higher region
if (x2[j] <= 0.5*CONST_PI-atan(3.*g_inputParam[EPS]))
if(d−>Vc[VX1][k][j][i]<0.0)
d−>Vc[VX1][k][j][i] = 0.0;
d−>Vc[VX2][k][j][i] = 0.0;

else if (box−>vpos == X2FACE)
#ifdef STAGGERED_MHD
BOX_LOOP(box,k,j,i) d−>Vs[BX2s][k][j][i] = d−>Vs[BX2s][k][j][IEND];
#endif
else if (box−>vpos == X3FACE)
#ifdef STAGGERED_MHD
//ccm not used in 2D, in 3D yes but then do better
// BOX_LOOP(box,k,j,i) d−>Vs[BX3s][k][j][i] = d−>Vs[BX3s][k][j][IEND];
#endif

if (side == X2_BEG) /* – X2_BEG boundary – */
if (box−>vpos == CENTER)
else if (box−>vpos == X1FACE)
#ifdef STAGGERED_MHD
BOX_LOOP(box, k, j, i) d−>Vs[BX1s][k][j][i] = 0.0;
#endif
else if (box−>vpos == X3FACE)
#ifdef STAGGERED_MHD
#if INCLUDE_KDIR
BOX_LOOP(box, k, j, i) d−>Vs[BX3s][k][j][i] = 0.0;
#endif
#endif

if (side == X2_END) /* – X2_END boundary – */
if (box->vpos == CENTER)
else if (box->vpos == X1FACE)
#ifdef STAGGERED_MHD
BOX_LOOP(box, k, j, i) d->Vs[BX1s][k][j][i] = 0.0;
#endif
else if (box->vpos == X3FACE)
#ifdef STAGGERED_MHD
#if INCLUDE_KDIR
BOX_LOOP(box, k, j, i) d->Vs[BX3s][k][j][i] = 0.0;
#endif
#endif

if (side == X3_BEG) /* – X3_BEG boundary – */
X3_BEG_LOOP(k,j,i)

    if (side == X3_END)  /* – X3_END boundary – */
X3_END_LOOP(k,j,i)

void BodyForceVector(double *v, double *g, double x1, double x2, double x3) part:
g[IDIR] = -1.0/x1/x1;
g[JDIR] = 0.0;
g[KDIR] = 0.0;
```

In the other `*.c` files, the further modifications are needed:

-in the 80th line of `ct_update.c` subroutine for the Constrained Transport method, the `E_phi=0` condition should be set at the inner, stellar surface boundary condition:
`if (grid[IDIR].lbound != 0) JTOT_LOOP(j) Ex3e[0][j][IBEG-1]=0.0;`

-in the `boundary.c` subroutine, the line `UserDefBoundary (d, &center_box, side[is], grid);` is to be copied from its original position at the beginning of §6f about line 415 to the end of this subroutine, 15 lines below.

-if the power law cooling is used, at the beginning of `cooling.c` subroutine, about line 65, the original `cost` value can be divided by 10, 100 or so, to speed-up the simulation. The corectness of such a modification in the given physical case is to be verified (it amounts to smaller "cost" in the bremsstrahlung radiation).

-in the `visc_nu.c` subroutine, the following text is to be inserted:

```
double coeff, cs, eps2, rcyl, beta, Bg[3], Bpol;
rcyl=x1*sin(x2);
eps2=g_inputParam[EPS]*g_inputParam[EPS];
coeff=2./5./eps2*(1./x1-(1.-5./2.*eps2)/rcyl);
coeff = MAX(coeff,0.0);
cs=eps2*coeff; /* initial sound speed*/

BackgroundField (x1,x2,x3,Bg);
Bpol=sqrt((v[BX1]+Bg[0])*(v[BX1]+Bg[0])+(v[BX2]+Bg[1])*(v[BX2]+Bg[1]));
beta=2.*v[PRS]/(Bpol*Bpol);
```

```
if (beta>0.5)// && x2>0.1 && x2<CONST_PI-0.1)
*nu1=2./3.*v[RHO]*g_inputParam[ALPHAV]
*cs*sqrt(rcyl*rcyl*rcyl)*v[TRC];
else
*nu1 = 0.0;

*nu2 = 0.0;
```

The viscosity and resistivity are parameterized by the [SS73] α-prescription as proportional to $c_{\rm s}^2/\Omega_{\rm K}$: -in the `res_eta.c` subroutine, the following text is to be inserted:

```
    double cs, eps2, rcyl, coeff, beta, Bg[3], Bpol;
rcyl=x1*sin(x2);
eps2=g_inputParam[EPS]*g_inputParam[EPS];
coeff=2./5./eps2*(1./x1-(1.-5./2.*eps2)/rcyl); /* initial profile */
coeff = MAX(coeff,0.0);
cs=eps2*coeff; /* initial sound speed2̂ */

    BackgroundField (x1,x2,x3,Bg);
Bpol=sqrt((v[BX1]+Bg[0])*(v[BX1]+Bg[0])+(v[BX2]+Bg[1])*(v[BX2]+Bg[1]));
beta=2.*v[PRS]/(Bpol*Bpol);

    if (beta > 0.5)// && x2>0.1 && x2<CONST_PI-0.1)
eta[IDIR] = g_inputParam[ALPHAM]*cs
*sqrt(rcyl*rcyl*rcyl)*v[TRC];
eta[JDIR] = g_inputParam[ALPHAM]*cs
*sqrt(rcyl*rcyl*rcyl)*v[TRC];
eta[KDIR] = g_inputParam[ALPHAM]*cs
*sqrt(rcyl*rcyl*rcyl)*v[TRC];
else
eta[IDIR] = 0.0;
eta[JDIR] = 0.0;
eta[KDIR] = 0.0;
```

-in the “Write VTK vector fields” part of the subroutine `write_vtk.c` about line 327, changes for mag. field, to output total B0+B1, not only the varying part B1, are to be introduced:

```
if (vel_field)
DIM_EXPAND(v[0] = V[0][k][j][i]; x1 = grid−>x[IDIR][i]; ,
v[1] = V[1][k][j][i]; x2 = grid−>x[JDIR][j]; ,
v[2] = V[2][k][j][i]; x3 = grid−>x[KDIR][k];)
else
BackgroundField (x1, x2, x3, Bg);
DIM_EXPAND(v[0] = V[0][k][j][i]+Bg[0]; x1 = grid−>x[IDIR][i]; ,
v[1] = V[1][k][j][i]+Bg[1]; x2 = grid−>x[JDIR][j]; ,
v[2] = V[2][k][j][i]+Bg[2]; x3 = grid−>x[KDIR][k];)
```

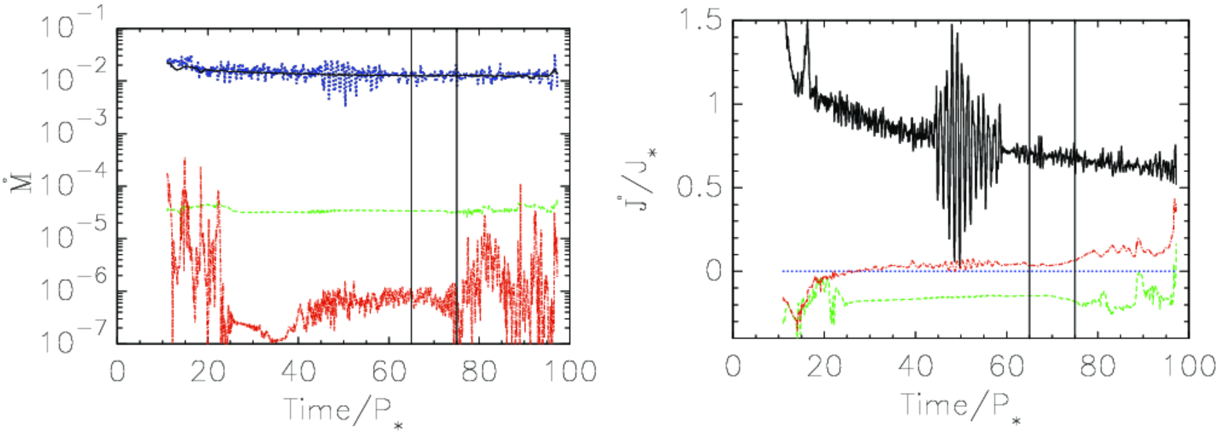

Figure 1.3: Mass and angular momentum fluxes in various flow components in the simulation with the $\Omega_\star = 0.1\Omega_{\rm br}$, $B_\star = 500$G and $\alpha_{\rm v} = \alpha_{\rm m} = 1$. With vertical solid lines is indicated the time interval in which we average the fluxes in each of the flow components. *Left panel*: Mass fluxes in code units $\dot{M}_0 = \rho_{\rm d0}\sqrt{GM_\star R_\star^3}$. The solid (black) line shows the mass flux through the disk at R=12R$_\star$ and the dotted (blue) line shows the mass flux loaded onto the star through the accretion column. Those two fluxes are much larger than the fluxes in the other components of the flow, the mass flux through the magnetospheric ejection at the radius R=12R$_\star$ (dot-dashed red line) and into the stellar (magnetospheric) wind (long-dashed green line). *Right panel*: Torques on the star in the units of $J_\star = M_\star R_\star^2\Omega_\star$, which in the case of YSO's, correspond to the stellar spin-up/down in Myrs. Positive torque spins the star up, and negative slows down its rotation. With the dashed (green) line is shown the torque by the stellar (magnetospheric) wind, the torques by the matter flowing onto the star through the accretion column from the distance beyond and below the corotation radius R_{cor} are shown with the dotted (blue) and solid (black) lines. With the dot-dashed (red) line is shown the torque exerted on the star by the magnetospheric ejection.

1.4.2 Modifications in the `pluto.ini` subroutine:

The file `pluto.ini` with parameters of the simulation should also be modified. We work in the spherical grid. The resolution we are using in the example setup here is $R \times \theta = (109 \times 50)$ grid cells, in a logarithmic stretched radial grid and in a half of the meridional half-plane in a uniform grid $\theta = [0, \pi/2]$. This resolution showed to be sufficient for qualitative capturing of the various features in 2D simulations. For the publication precision, at least $R \times \theta = (217 \times 100)$ grid cells is needed in the same physical domain of $R \times \theta = ([1, 30] \times [0, \pi/2])$. In the `GRID` definition section:

```
    [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
```

In the `TIME` section:

```
[Time]
CFL 0.4
CFL_max_var 1.2
tstop 627.9
first_dt 1.e-6
```

-in the `SOLVER` section:

```
[Solver]
Solver hll
```

-in the `BOUNDARY`section:

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

-in the `Static Grid Output` section:

```
[Static Grid Output]
uservar 3 nu num Te
dbl 6.279 -1 single_file
vtk 6.279 -1 single_file
log 100
```

-in the `PARAMETERS` section:

```
[Parameters]
ALPHAM 0.4
MU 0.7
TEMPF 10.0
RHOC 0.01
RD 3.7
EPS 0.1
OMG 0.1
ALPHAV 1.0
DFLOOR 5.e-7
```

1.5 Settings choice in `definitions.h` subroutine:

Here the modules used in the codes are defined. We copy the complete text:

```
#define PHYSICS MHD
#define DIMENSIONS 2
#define GEOMETRY SPHERICAL
#define BODY_FORCE VECTOR
#define COOLING POWER_LAW
#define RECONSTRUCTION LINEAR
#define TIME_STEPPING RK2
#define NTRACER 1
#define PARTICLES NO
#define USER_DEF_PARAMETERS 9

    /* – physics dependent declarations – */
```

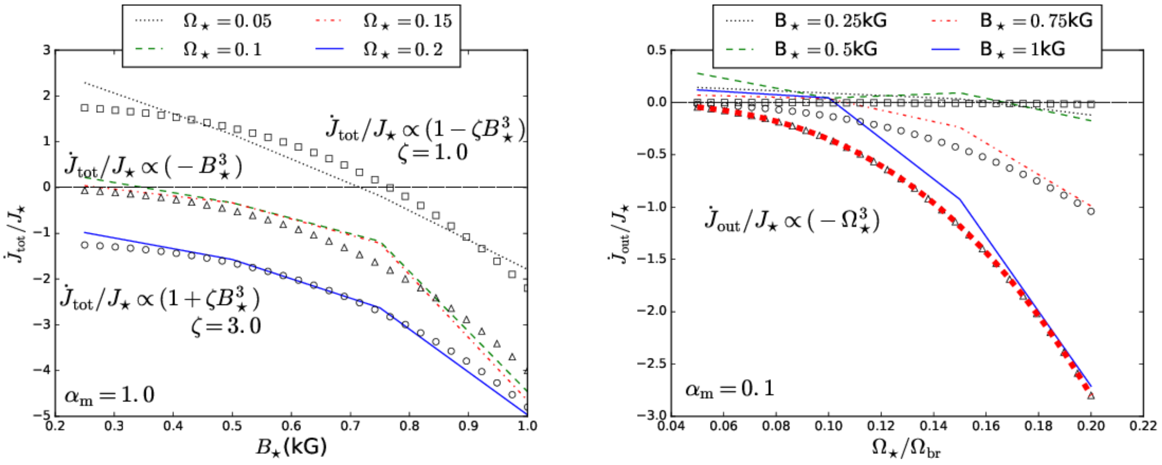

Figure 1.4: Torques on the star *left panel* and the torque exerted by the magnetospheric ejection *right panel*, expressed in units of $\dot{J}_\star$, in the cases with $\alpha_{\rm v} = 1.0$ and different $B_\star$ and $\Omega_\star$, respectively, for slowly rotating stars, in the cases with $\alpha_{\rm m} = 1.0$ and $\alpha_{\rm m} = 0.1$.

```
#define EOS IDEAL
#define ENTROPY_SWITCH NO
#define DIVB_CONTROL CONSTRAINED_TRANSPORT
#define BACKGROUND_FIELD YES
#define AMBIPOLAR_DIFFUSION NO
#define RESISTIVITY EXPLICIT
#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
#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) */
```

In the code, normalized equations are solved. The unit length, velocity, and mass are chosen with the stellar radius $R_\star$, the Keplerian speed at the stellar surface $v_{\mathrm{K}\star}$, and mass $M_\star$. The time unit is then $t_0 = R_\star/v_{\mathrm{K}\star}$. Time t in the results is measured in the number of stellar rotation periods P$_\star$. The mass flux rate is measured in $\dot{M}_0 = \rho_{\mathrm{d}0}R_\star^2 v_{\mathrm{K}\star}$, which is the free parameter in simulation, from which $\rho_{\mathrm{d}0}$ is determined, the disk initial density. The initial coronal density is defined as a free parameter in the code, $\rho_{\mathrm{c}0} = 0.01\rho_{\mathrm{d}0}$. The magnetic field unit is defined by $B_0 = v_{\mathrm{K}\star}\sqrt{\rho_{\mathrm{d}0}}$. The torque in the simulations is measured in units of $\dot{J}_0 = \rho_{\mathrm{d}0}R_\star^3 v_{\mathrm{K}\star}^2$.

The simulations can be rescaled to different objects using the scaling coefficients from Table 1.1. In the case of compact objects, when the radial extension of the domain, measured from the axis of rotation, reaches the light cylinder, $R_{\ell c}\Omega_\star = c$, where the azimuthal velocity equals the speed of light, one should consider the validity of the results. After this surface, there should not be any back-reaction towards the star, since it would be acausal.

1.6 Results in PLUTO setup

In [Čem19] we performed a parameter study with the similar set-up as here, only that we did not use the cooling source term, but excluded the dissipation fluxes of viscosity and resistivity in the energy equation. We also used a modified Roe solver, as described in the setup in the Appendix there. We varied the stellar rotation rate, magnetic field strength and resistivity in the disk and compared the changes in results in dependence on those parameters. With such a setup we obtained a quasi-stationary solution suitable for investigation of the dependence of stellar angular momentum on the physical parameters. A zoom in the closest vicinity of the central object and accretion column in the example result from those results is shown in Fig. 1.2, with the mass and angular momentum fluxes in Fig.1.3 showing the quasi-stationarity of the simulation after the relaxation from initial and boundary conditions. Trends in such results, which were used to find numerical expressions for torques on the central star and their component exerted by the magnetospheric ejections are shown in the Fig. 1.4.

Results with the simplified setup presented here are a good starting point for further development of an individual setup. A physical motivation for the cooling source term should be found, and the closest module from this part of the code chosen–see the PLUTO User Guide, given in `PLUTO/Doc` directory with the code.

Chapter 2

Accretion disk in General Relativity

Thin accretion disks are observed or assumed in many objects at different scales, from young stellar objects and protoplanetary disks, to close binaries and Active Galactic Nuclei (AGNs). The most often used and cited works about thin disks, [SS73] α-viscous solution and [BH91] disk with magneto-rotational instability (MRI) both operate with stress in the radial direction proportional to pressure. Such disks are often found unstable in simulations where the observational counterpart is stable-e.g. in [LE74; Pir78] the luminous disks were unstable, while corresponding black hole X-ray binaries exhibit stable accretion throughout the disk. Similarly, results in thin disk simulations do not match the flows anticipated in Active Galactic Nuclei (AGN) ([DB19]).

Thicker accretion disks could, together with larger accretion rate than predicted by [SS73], solve some of the problems. One of the ways to ensure the larger disk thickness is through development of large toroidal fields, which increase the vertical magnetic pressure. A novel solution was also found in the form of puffy disk in [Lan+19], which includes the radiative transport.

In this chapter we will use the non-magnetic solution of [KK00] from Newtonian solution presented in the previous Chapter as an initial condition in the General Relativistic KORAL code [Sad+14]. We will add the magnetic field and set the simulation for a magnetized thin disk around a stellar black hole in two often used initial magnetic field configurations: with the poloidal loops of magnetic field in the disk, and with the open, hour-glass shaped interstellar magnetic field. KORAL is often used for radiative setups, so we present the equations with radiation included, but we do not include radiation in our simulations here. This part of work is still in preparation for the thin disk case and reader should follow our later publications.

2.1 General relativistic setup

We set Kluźniak-Kita thin disk around a $10M_\odot$ black hole, using the Modified Kerr-Schild (MKS) [MG04] coordinates in the KORAL code [Sad+14], as shown in Fig. 2.2. Modified KS coordinates were introduced with the purpose of stretching the grid radially and concentrating resolution in the equatorial region.

$$\begin{aligned} r_{\mathrm{KS}}(s) &= R_0 + e^S \\ \theta_{\mathrm{KS}}(\theta) &= \theta + \frac{h}{2}\sin(2\theta) \end{aligned} \tag{2.1}$$

General relativistic radiative MHD (GRRMHD) equations are governed by conservation of mass density and stress-energy tensor with the addition of radiation tensor:

$$(\rho u^\mu)_{;\mu} = 0,\ (T^\mu_\nu + R^\mu_\nu)_{;\mu} = 0, \tag{2.2}$$

with

$$(T^\mu_\nu)_{;\mu} = G_\nu,\ (R^\mu_\nu)_{;\mu} = -G_\nu. \tag{2.3}$$

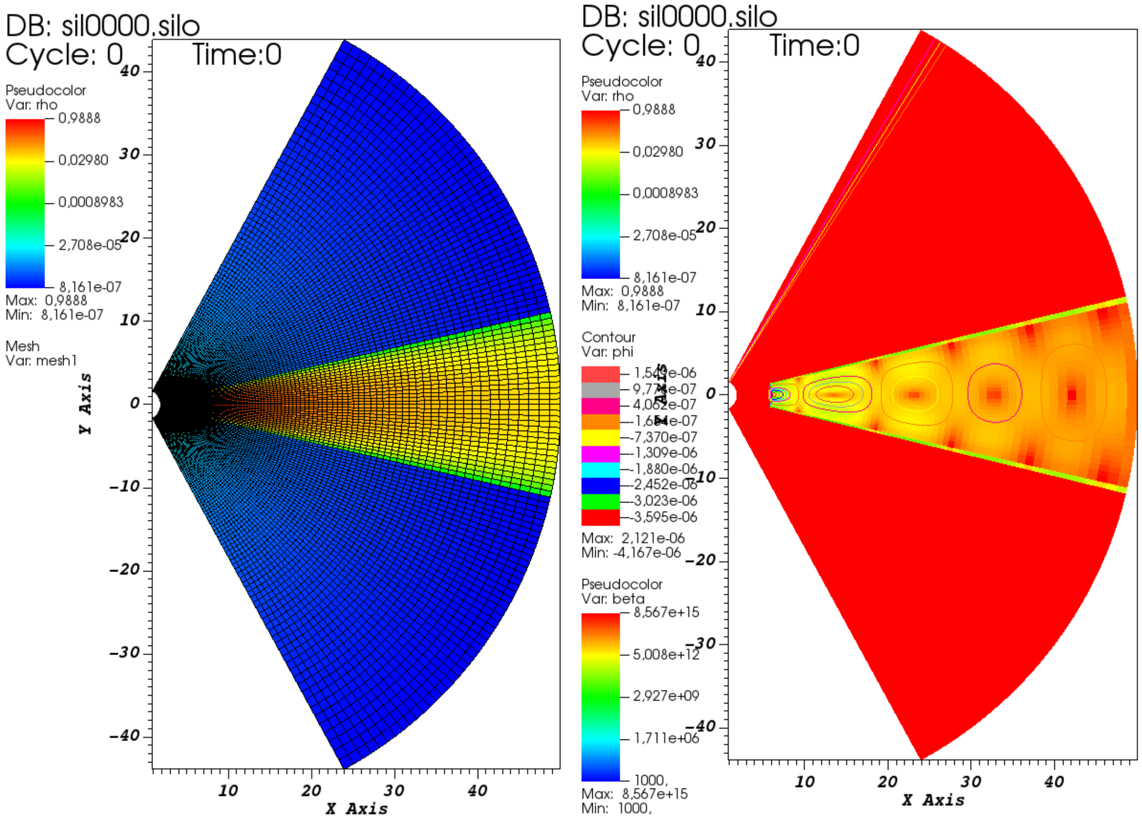

Figure 2.1: Initial density distribution in our simulations, shown in logarithmic color grading, with the grid mesh shown in black solid lines (*Left panel*), and initial beta plasma distribution (*Right panel*).

Without radiation, the second of Eqs. 2.2 becomes simply

$$(T^\mu_\nu)_{;\mu} = 0. \tag{2.4}$$

Here ρ is the gas density in a co-moving fluid frame, u^μ is the gas four velocity measured in a lab frame, R^μ_ν is the radiation tensor and T^μ_ν is the energy-momentum tensor:

$$T^\mu_\nu = (\rho + u + p + b^2)u^\mu u_\nu + (p + \frac{1}{2}b^2)\delta^\mu_\nu - b^\mu b_\nu, \tag{2.5}$$

also measured in a lab-frame [GMT03]. The specific internal energy is denoted u, $p = (\Gamma - 1)u$ is the gas pressure, and b^μ is the magnetic field four-vector.

The radiation 4-force density

$$G_\nu = (\chi_\nu I_\nu - \eta_\nu)d\nu d\Omega N^i \tag{2.6}$$

in the fluid frame simplifies to[1]:

$$\widehat{G} = \begin{bmatrix} \kappa(\widehat{E} - 4\pi\widehat{B}) \\ \chi\widehat{F}^i \end{bmatrix} \tag{2.7}$$

where $\widehat{B} = \sigma T^4/\pi$ is the (integrated) Planck function which corresponds to the gas temperature T, σ is the Stefan-Boltzmann constant, χ_ν and η_ν are the frequency-dependent opacity and emissivity, respectively, and κ, χ are the frequency-integrated absorption and total opacity components, respectively. The radiation stress-energy tensor in orthonormal frame contains different moments of the specific intensity I_ν, and writes

$$\widehat{G} = \begin{bmatrix} \widehat{E} & \widehat{F}^i \\ \widehat{F}^j & \widehat{P}^{ij} \end{bmatrix} \tag{2.8}$$

in the fluid frame, with

$$\widehat{E} = \int I_\nu d\nu d\Omega,\ \widehat{F}^i = \int I_\nu d\nu d\Omega N^i, \tag{2.9}$$
$$\widehat{P}^{ij} = \int I_\nu d\nu d\Omega N^i N^j,$$

representing the radiation energy density, the radiation flux and the radiation pressure tensor, respectively. N^i is unit vector in direction x^i.

To close the set of equations, we need the complete radiation stress-energy tensor $R^{\mu\nu}$, but we can compute only the radiative energy density R^{tt} and radiative fluxes R^{ti}. The simplest way of computing them is to use the so called Eddington scheme, in which a nearly isotropic radiation field is assumed, with

$$\widehat{P}^{ij} = \frac{1}{3}\widehat{E}\delta^{ij}. \tag{2.10}$$

Such an assumption holds only in optically thick medium, while we are interested in radiation escaping photosphere. We use the so-called M1 closure scheme, in which the Eddington closure is satisfied, but not in the fluid frame, but in the orthonormal "rest frame" of the radiation. This frame is defined as the one in which the radiative flux vanishes. Then we have R^{tt}, $R^{ii} = E/3$ and the rest of the R components are zero. E is the energy density in the radiation rest frame. In the radiation rest frame

$$R^{\mu\nu} = \frac{4}{3}Eu^\mu_R u^\nu_R + \frac{1}{3}Eg^{\mu\nu}, \tag{2.11}$$

[1] All the quantities with "wide hats" $\widehat{X}$ are measured in the fluid frame.

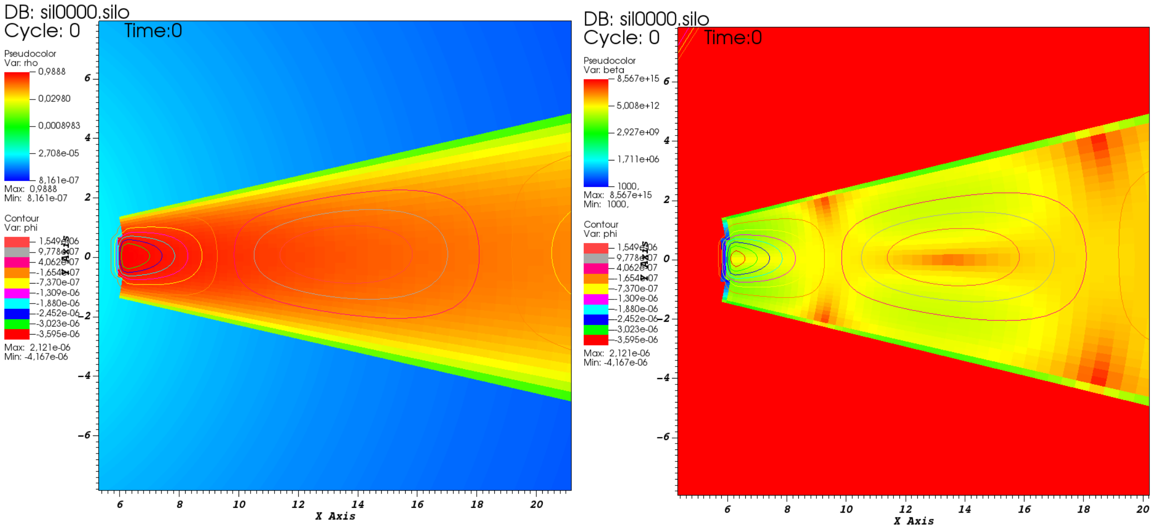

Figure 2.2: Zoom into the initial distribution of density and poloidal magnetic field loops in our simulations (*Left panel*) and a zoom into the initial β plasma (*Right panel*).

with $g^{\mu\nu}$ the contravariant metric tensor, which is in this frame the flat space Minkowski metric, and $u^{\mu}_{R} = 1, 0, 0, 0$. Being covariant, the Eq. 2.11 is valid also in the lab frame, with u^{μ}_{R} as a 4-velocity of the radiation rest frame, measured by an observer in the lab frame.

M1 closure scheme works well at the low and high extremes of the optical depth, and is also a fairly good approximation at intermediate values of the optical depth. It is a good method with a single source of light, with multiple sources it is less appropriate.

GRMHD equations are solved using the KORAL code [Sad+14], using the HLLC Riemann solver and the monotonized central difference limiter MC. Time is evolved using the second order Runge-Kutta method RK2. To ensure $\nabla \cdot \boldsymbol{B} = 0$, we use the flux-constrained transport (CT) method.

2.2 Basics of `github` use

If you share the build of KORAL called KKDISC with another researcher AAA:
In the new directory where you want the complete code, do

```
git clone https://github.com/AAA/KKDISC.git
```

then copy the files (make sure you have `.git` directory and files) into your folder (or keep AAA's) and copy your Makefile from working distribution (AAA's is different from yours) and run the setup and compilation (`./mser` for serial, `./mpar` for parallel).

If you wish only to update from the latest AAA's changes:

```
git pull
```

will update all the files. If you wish to exclude some files from update, like `Makefile`, then create the file `.gitignore` and put there the names of the files you wish to keep non-updated:
e.g. if you wish to keep `Makefile`, put inside the `.gitignore`:

```
.gitignore:
Makefile
```

This will save the `.gitignore` itself, and `Makefile`.

2.3 Basics of KORAL use

-in `Koral` directory, where you have a running test version, create one for your problem e.g. `KKDISC`, by copying the running one as a template into the new name.
-copy the most similar of available problems from `PROBLEMS` directory into a new problem. Here we make a `KKdisk` directory as an example.
-modify `PROBLEM/KKdisk/define.h` and `PROBLEM/KKdisk/init.c`
-you can add some part of the setup into `tools.c`, which is also read during the compilation of the code
-clean `KKDISC/dumps` and `KKDISC/analysis`
-add the paths to your `gsl` and `silo` libraries in the file `Makefile`
-be careful with `dumps` directory, to have it in the folder (or create if not there), as KORAL will not create it by itself, it will crash if it does not find this folder.
-for serial compilation, use `./mser.sh`, then run with `./ko` (you need `silo` for serial runs, but KORAL may run without producing silo files when compiled with"`-DNOSILO`" flag in the `Makefile`.)
-for serial run, you can go to `/KKDISC/dumps` and plot using `silo` output with VisIt visualisation package
-for parallel compilation (you do not need `silo`), use `./mpar.sh`, then run with

```
mpirun -np 4 ./ko
```

for e.g. 4 processors. Take care that NT(XYZ) has to be divisible with the number of processors (there will be a message if not).
-for plotting after serial run, go to `dumps` folder and plot with visit the `*.silo` files.
-for plotting after parallel run, you need to compile in serial `./mser.sh` and then under `KKDISC` execute

```
./ana 0 12 1
```

where numbers are n_0 n_{max} n_{step} (12 for 12 timesteps), and you get inside the `/KKDISC/analysis` directory (which you need to create, it will not be created by default!) the silo output which you can plot with visit. You do not need need to recompile `ana` with change of number, just run it again for e.g. 150 files of output.
-for restart, in `define.h` change to have
#define restart
and put after the `RESTARTNUM X` instead of X the number of file from which you want to restart. This change needs recompilation. If you define `RESTARTNUM -1` it will restart from the last file.

After other changes, a recompilation is needed:

```
//restart #define RESTART
#define RESTARTNUM 52
```

2.4 Setup with loops of poloidal disk magnetic field

We made our example numerical experiment with the initial magnetic field setup of loops of poloidal magnetic field entrained inside the disk. The field direction is changing in the loops so to weaken the field for the Standard and Normal Evolution (SANE) setup, in which MRI develops.

In the working directory with the setup, e.g. KKDISC, in the `problem.h` file are defined calls to the KORAL modules we wish to use in the simulation:

//KORAL - problem.h //choice of the problem plus some definitions
//available problems:
//1 RADBEAM2D - beam of light
//…
//142 PARTIALTDE – INFDISK modified for partial TDE binding energy distribution non-constant
#define PROBLEM 143

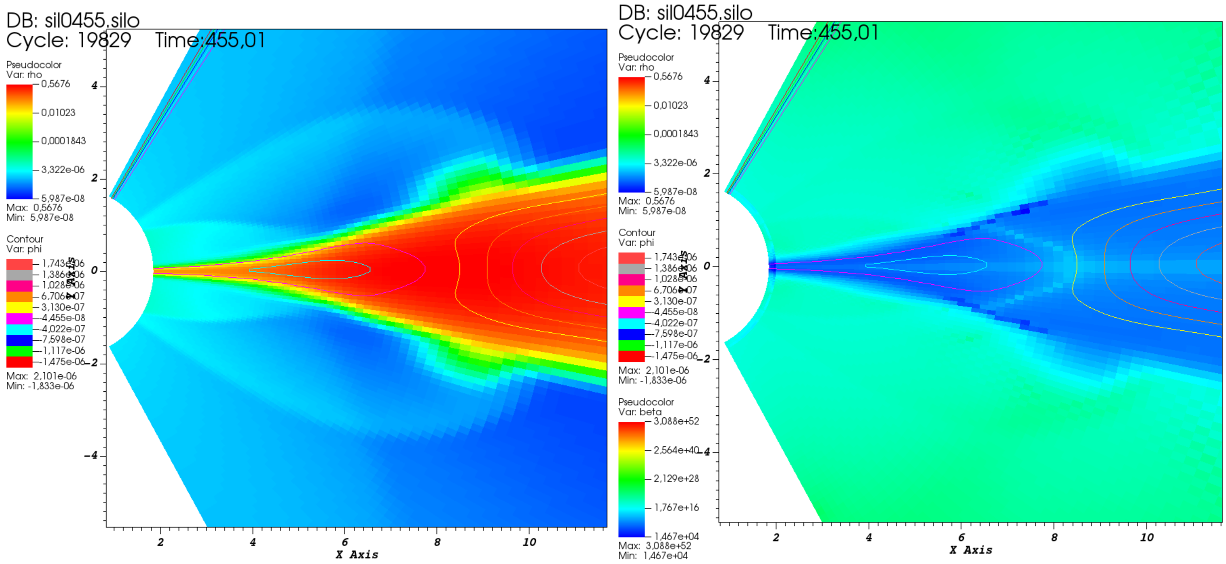

Figure 2.3: *Left panel*: Zoom into the density distribution in our simulations after T=455 r_g/c. *Right panel*: Zoom into the beta plasma distribution at the same time.

```
#if(PROBLEM==143)
...
#define PR_DEFINE ”PROBLEMS/KKDISC/define.h”
#define PR_BC ”PROBLEMS/KKDISC/bc.c”
#define PR_INIT ”PROBLEMS/KKDISC/init.c”
#define PR_POSTINIT ”PROBLEMS/KKDISC/postinit.c”
#define PR_TOOLS ”PROBLEMS/KKDISC/tools.c”
    #endif

    //including problem specific definitions from PROBLEMS/XXX/define.h
/**/
#ifdef _OPENMP
#define OMP
#endif

    #include PR_DEFINE
```

An example of `define.h` file is given below:

```
//general
#define BHDISK_PROBLEMTYPE
//restart
#define RESTART
#define RESTARTGENERALINDICES
#define RESTARTNUM -1
    //magnetic choices
#define MAGNFIELD
#define GDETIN 1
#define VECPOTGIVEN
```

```
    //#define QUADLOOPS
#define MULTIPLELOOPS
#define HLOOPS 0.5
#define MAXBETA 1e-3 //target pmag/pgas int the midplane
    /* //We skip dynamo now, but for longer run in 2D it is needed
#define MIMICDYNAMO
#define CALCHRONTHEGO
#define THETAANGLE 0.25
#define ALPHAFLIPSSIGN
#define DYNAMORADIUS 15.
#define ALPHADYNAMO 0.314
#define DAMPBETA
#define BETASATURATED 0.1
#define ALPHABETA 6.28
#define MAXBETA .01 //target pmag/pgas int the midplane
*/
    //reconstruction / Courant
#define INT_ORDER 1
#define TIMESTEPPING RK2IMEX
#define TSTEPLIM 0.5
#define FLUXLIMITER 0
#define MINMOD_THETA 1.5
/**/ //blackhole
#define MASS 10.//MSUNCM
#define BHSPIN 0.

    //physics
#define GAMMA (5./3.)
//coordinates
//#define myMKS3COORDS // use for 3D
#define myMKS2COORDS // good for 2D
#define MKSR0 0.
#define MKSH0 0.6
#define MKSMY1 0.0025
#define MKSMY2 0.025
#define MKSMP0 1.5
#define METRICAXISYMMETRIC
#define RMIN 1.85
#define RMAX 50
    //setup for MKS2 coords
#ifdef myMKS2COORDS
#define MYCOORDS MKS2COORDS
#define MINX (log(RMIN-MKSR0))
#define MAXX (log(RMAX-MKSR0))
#define MINY (0.1) // 0.001 for bigger domin closer to polar axis
#define MAXY (1.-MINY)
#endif
    #ifdef myMKS3COORDS //modified Kerr-Shild further from axis
```

```
#define METRICNUMERIC
#define MYCOORDS MKS2COORDS
#define MINX (RMIN)
#define MAXX (RMAX)
//#define Y_OFFSET 0.009
#define MINY (0.)
#define MAXY (M_PI-MINY)
#endif
    #define PHIWEDGE (M_PI/2.)
#define MINZ (-PHIWEDGE/2.)
#define MAXZ (PHIWEDGE/2.)
    //resolution
#define TNX 124 //16*17
#define TNY 124 //16*12
#define TNZ 1//32//128 //16*8

    //number of tiles for MPI
#define NTX 1//32//16//32//16//17
#define NTY 1//16//16//32//8//12
#define NTZ 1//2//8
    //boundary conditions
#define SPECIFIC_BC //in bc.c
#define PERIODIC_ZBC
//#define PERIODIC_XBC
//#define PERIODIC_YBC
    //output
#define DTOUT1 1. //res - files
#define DTOUT2 1000. //avg - files
    #define OUTCOORDS BLCOORDS #define OUTVEL VEL4
#define ALLSTEPSOUTPUT 0
#define NSTEPSTOP 1.e10
#define NOUTSTOP 5000
#define SILOOUTPUT 1
#define SIMOUTPUT 1
#define RADOUTPUT 0
#define SCAOUTPUT 0
#define AVGOUTPUT 0
#define THOUTPUT 0
#define THPROFRAD1US 30

    #if(TNZ==1)//if 2D
#define SILO2D_XZPLANE
#else
#define FULLPHI
#endif
    //initial disk
//This for later choices
#define TDISK 1
```

```
    //disk choices

    #if(TDISK==1) //Kluzniak&Kita(2000) thin disk in HD
// Inner edge of the disk
#define RINNER r_ISCO_BL(BHSPIN)
// Max density in the disk centre
#define RHO_DISC_MAX 1.0
// max atm. density = RHO_EPS * RHO_DISC_MAX (at horizon)
#define RHO_EPS 1.0e-4
#define EPSS 0.1//thin disk height ratio
#define ALPHA_DISC 0.5 //viscous alpha
#define HR_INIT 0.1 //initial disk thickness
#endif //TDISK 1
    #if(TDISK==2) //Kluzniak&Kita(2000) thin disk with hourglass Bfield
#define EPSS 0.1//thin disk height ratio
    #endif//ending TDISK
    #if(TDISK==3) //Kluzniak&Kita(2000) thin disk in RMHD
// TBD
#endif
//#define BETANORMFULL
    //rmhd floors
#define RHOFLOOR 1.e-50
//#define UURHORATIOMIN 1.e-8//1K: uu/rho = 7.259162e+12
//#define UURHORATIOMAX 1.e20
#define CORRECT_POLARAXIS
#define NCCORRECTPOLAR 2
#define UURHORATIOMIN 1.e-10
#define UURHORATIOMAX 1.e2
#define EERHORATIOMIN 1.e-20
#define EERHORATIOMAX 1.e20
#define EEUURATIOMIN 1.e-20
#define EEUURATIOMAX 1.e20
#define B2UURATIOMIN 0.
#define B2UURATIOMAX 100000.
#define B2RHORATIOMIN 0.
#define B2RHORATIOMAX 50.
#define GAMMAMAXRAD 50.
#define GAMMAMAXHD 50.
```

The init.c file is:

```
    int init_KKdisk(ldouble r, ldouble th, ldouble *rho,ldouble *uint);

    //MC240322–setup of KK00 disk, first HD version, then slowly moving towards
//a complete MHD version. Idea is to set it for NS and then shift to SMBH case.
ldouble rho,mx,my,mz,m,E,uint,pgas,Fx,Fy,Fz,pLTE,ell;
ldouble uu[NV], pp[NV],ppback[NV],T,uintorg;
ldouble Vphi,Vr;
ldouble D,W,uT,uphi,uPhi;
ldouble rcyl, pres, eps2, coeff, lambda1, rhoc, pc, rd, alphav;
ldouble mub, rminb, mmb;
int loops, openn;//to be given by hand later, for choice of loops model //geometries

    //geometry in MYCOORDS
struct geometry geom;
fill_geometry(ix,iy,iz,&geom);

    //geometry in BLCOORDS
struct geometry geomBL;
fill_geometry_arb(ix,iy,iz,&geomBL,KERRCOORDS);

    //r and theta in BL
ldouble r=geomBL.xx;
ldouble th=geomBL.yy;

    init_KKdisk(r, th, &rho, &uint);

    if (rho<0)
// for now go with the 0 atm.
set_hdatmosphere(pp,geom.xxvec, geom.gg, geom.GG,5);

else
set_hdatmosphere(ppback,geom.xxvec,geom.gg,geom.GG,5);
eps2=HR_INIT*HR_INIT;
rcyl=r*sin(th);
rd=RINNER;
alphav=ALPHA_DISC;
rhoc=RHO_EPS * RHO_DISC_MAX;
//notice that rhoc and coeff are multiplied with rho0, in KK00 rho0=1.
coeff=RHO_DISC_MAX*(2./5./eps2)*(RINNER/r-(1.-2.5*eps2)*RINNER/rcyl);
lambda1=(11./5.)/(1.+(64./25.)*alphav*alphav);
pp[RHO] = pow(coeff,3./2.);
    ldouble ucon[4];
    ucon[1] = 0;
ucon[2] = 0;
ucon[3] = 0;

    ucon[0] = sqrt(-1.0/geomBL.gg[0][0]);
pres=(1./RINNER)*eps2*pow(coeff,5./2.);
    ucon[1] = -(alphav/sin(th))*eps2*(10.-(32./3.)*lambda1*alphav*alphav-lambda1*
(5.-1./(eps2*tan(th)*tan(th))))/s,qrt(rcyl);// //ucon[1] = 0;
ucon[3] = (sqrt(1.-5.*eps2/2.)+(2./3.)*eps2*alphav*alphav*lambda1*(1.-6./(5.*eps2*tan(th)*tan(th))))/sqrt(rcyl)/r;
// //ucon[3] = sqrt(1.0/pow(r,3.0));
fill_utinucon(ucon,geomBL.gg, geomBL.GG);

    //ucon[0] = sqrt((-1.0-geomBL.gg[3][3]*ucon[3]*ucon[3])/geomBL.gg[0][0]);

    pp[UU] = pres/(GAMMA-1);
    ucon[1]*= ucon[0];
ucon[2]*= ucon[0];
ucon[3]*= ucon[0];
    fill_utinucon(ucon,geomBL.gg, geomBL.GG);
//ucon[0]=1./sqrt(-geomBL.GG[0][0]);
conv_vels(ucon,ucon,VEL4,VELPRIM,geomBL.gg,geomBL.GG);
    //printf("
    pp[VX] = ucon[1];
pp[VY] = ucon[2];
pp[VZ] = ucon[3];

    //—-finish of the HD setup by KK00——–

    trans_pall_coco(pp, pp, KERRCOORDS, MYCOORDS,geomBL.xxvec,&geomBL,&geom);

    #ifdef MAGNFIELD
ldouble Acov[4];
Acov[0]=Acov[1]=Acov[2]=0.;
    Acov[3]=my_max(pow(pp[RHO]*geomBL.xx/4.e-22,2.)-0.02,0.)*sqrt(1.e-23);
#ifdef QUADLOOPS
Acov[3]*=sin((M_PI/2.-geomBL.yy)/0.1);
#endif

    #ifdef MULTIPLELOOPS
Acov[3]*=sin(geomBL.xx/3.);
#endif

    //Acov[3]=my_max(pow(pp[RHO]*geomBL.xx*geomBL.xx/4.e-20,2.)-0.02,0.)*sqrt(1.e-23)*
pow(sin(fabs(geomBL.yy)),4.);
    pp[B1]=Acov[1];
pp[B2]=Acov[2];
pp[B3]=Acov[3];
    #endif

    //entropy
pp[5]=calc_Sfromu(pp[0],pp[1],ix,iy,iz);
//to conserved
p2u(pp,uu,&geom);
    int iv;
    for(iv=0;iv<NV;iv++)

set_u(u,iv,ix,iy,iz,uu[iv]);
set_u(p,iv,ix,iy,iz,pp[iv]);

    //entropy
update_entropy_cell(ix,iy,iz,0);
```

```
set_cflag(0,ix,iy,iz,0);
```

The `tools.c` file contains additional information about setup:

```
    int init_KKdisk(ldouble r, ldouble th, ldouble *rhoout,ldouble *uintout)

ldouble eps2=HR_INIT*HR_INIT;
ldouble rcyl=r*sin(th);
ldouble rd=RINNER;
ldouble alphav=ALPHA_DISC;
ldouble rho0=RHO_DISC_MAX;//free param. multipl. factor to KK00=1 max disk density
ldouble rhoc=RHO_EPS * RHO_DISC_MAX;

    ldouble coeff=RHO_DISC_MAX*2./5./eps2*(RINNER/r-(1.-5./2.*eps2)*RINNER/rcyl);
ldouble lambda1=11./5./(1.+64./25.*alphav*alphav);

    ldouble rho = rhoc*pow(r,-3./2.);
ldouble pres= 2./5.*rhoc*pow(r,-5./2.);
ldouble uint = pres/(GAMMA-1);
ldouble pc=pres;

    pres=(1./RINNER)*eps2*pow(coeff,5./2.);
    if (pres >= pc && rcyl > rd)
rho = pow(coeff,3./2.);

else
rho = -1.0;

*rhoout = rho;
*uintout = pres/GAMMAM1;
```

2.5 Open magnetic field setup

In simulations with a large scale hour-glass magnetic field we follow [ZS18] and [Mis+20]. The vector potential described in those papers is:

$$A_\phi = \begin{cases} \frac{1}{2} r \sin\theta B_0 (\frac{r_{\min}}{R_0})^m & \text{if r} \leq \text{r}_{\min} \\ \frac{B_0}{R_0^m} \frac{(r\sin\theta)^{m+1}}{m+2} + \frac{B_0 r_{\min}^{m+2}}{R_0^m r \sin\theta} (\frac{1}{2} - \frac{1}{m+2}) & \text{if r} > \text{r}_{\min} \end{cases} \tag{2.12}$$

They used the different power coefficient m, in BMishra it was -5/4, in Zhu & Stone -9/4. One example of such a setup in KORAL code with power law index -5/4 and a result after T=25 is shown in Fig. 2.4.

The thin disk simulations of this kind were up to now done only in [Mis+22], where highly luminous thin disks were simulated. In such disks, MRI is supposedly overcome by the strong magnetic field, which stabilizes the disk. The initial disk was even thinner than ours, with height to radius ratio of 0.05 (we set 0.1). An example of their initial conditions setup in the cases corresponding to ours here is shown in Fig. 2.5, with pseudocolor plots of the logarithm of gas density and lines of magnetic field.

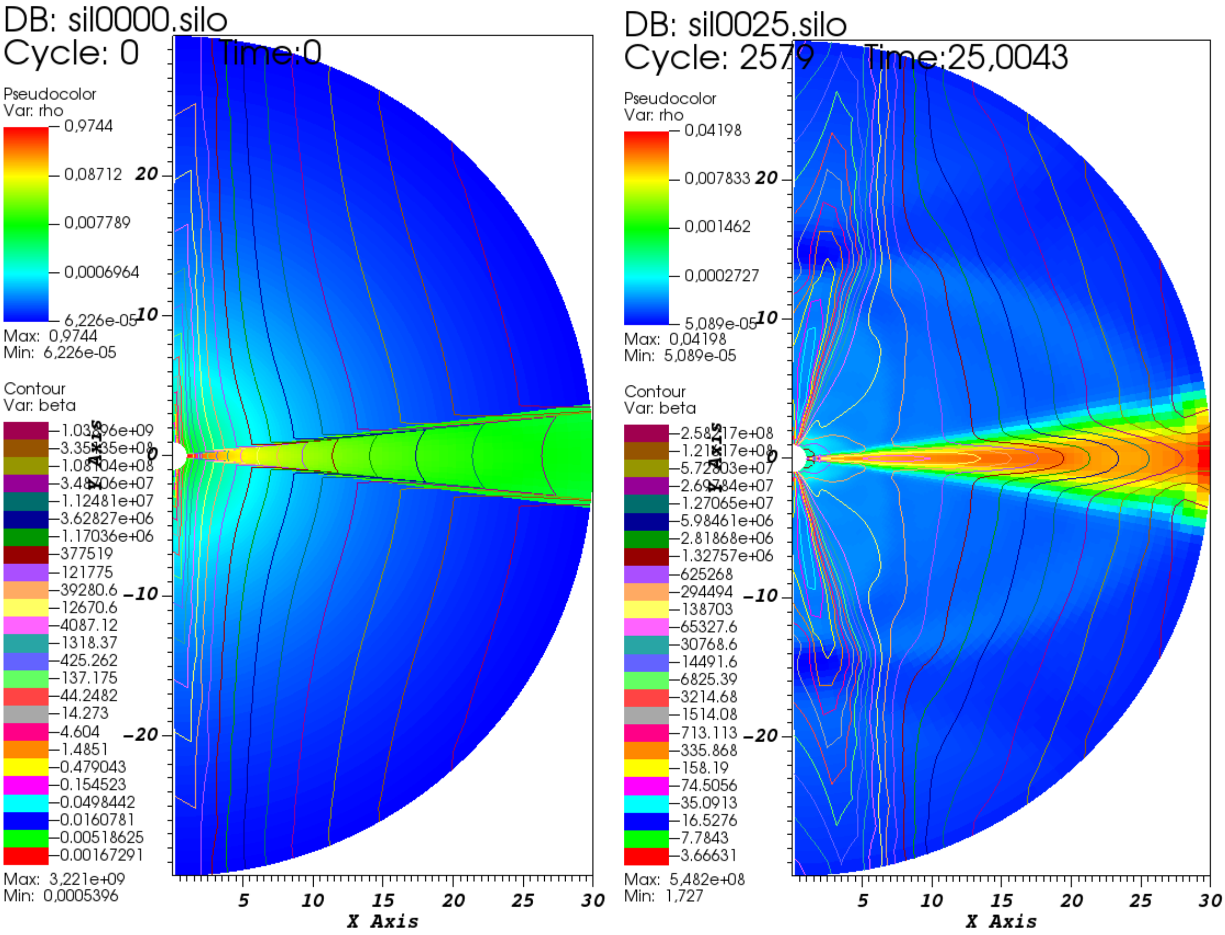

Figure 2.4: Zoom into the initial distribution of density and poloidal magnetic field loops in our simulations with open magnetic field loops at T=$0r_{\mathrm{g}}/c$ and T=$25r_{\mathrm{g}}/c$.

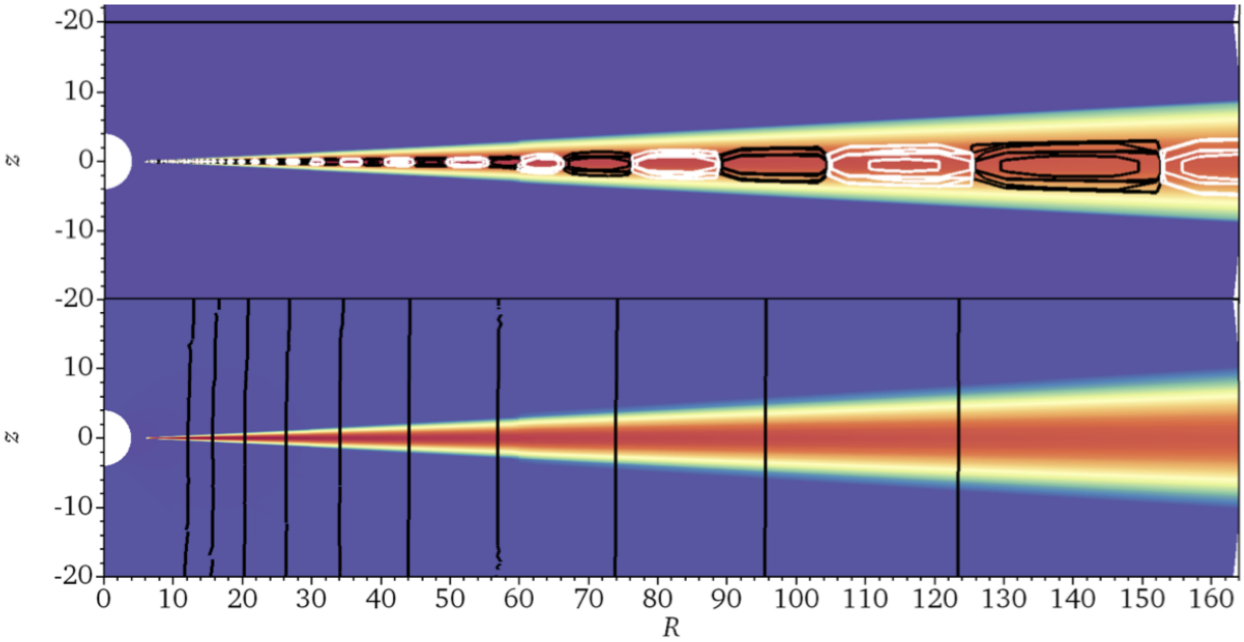

Figure 2.5: Pseudocolor plots of the logarithm of gas density, with multiloop and vertical field initial setup of magnetic field. Figure adopted from [Mis+22]. The initial disk height ratio is 0.05.

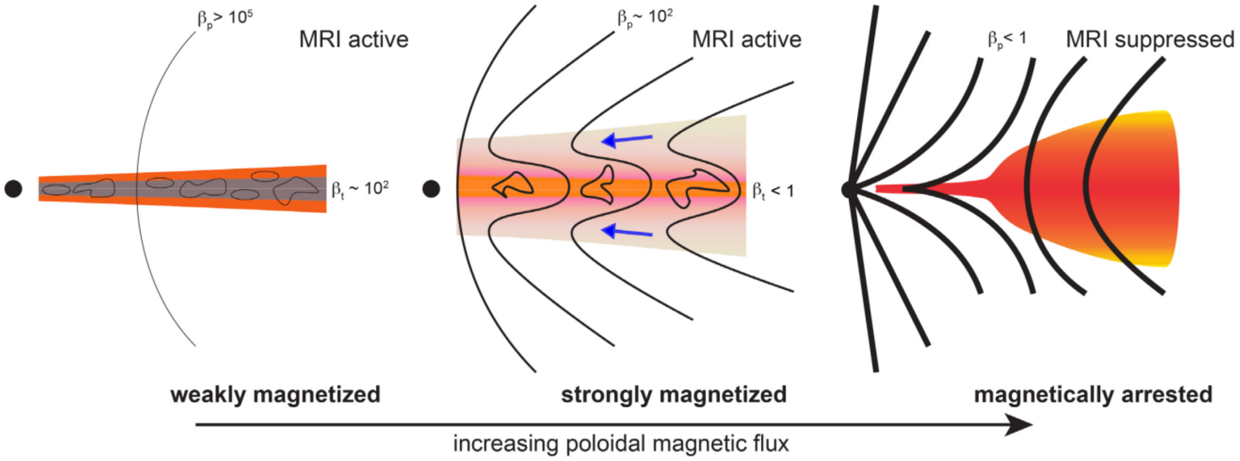

Figure 2.6: Schematic diagram of various accretion regimes, figure adopted from [Mis+20]. Panels from left to right correspond to disks with increasingly stronger magnetic field. Accretion is occurring where the magnetic field lines are pinched at the disk surface. The disk thickness variation between the left and centre images mimics the effect of strong magnetic pressure support. In the left and middle panels MRI is active, in the right panel MRI is suppressed.

Comparison of the outcomes in those simulations with their (and other groups) previous simulations, showed that that a zero-net-flux, multiloop magnetic field configuration is unable to stabilize itself in the radiation-pressure-dominated regime, independent of the initial setup.

In Fig. 2.6 is shown a schematic representation of three outcomes in the simulations with different strengths of magnetic field. In weakly magnetized disk, poloidal field is weak enough not to disturb the MRI. In strongly magnetized disks, the poloidal field can create the larger toroidal fields, which can affect disk structure in the vertical direction, but do not quench MRI or dominate the radial force balance. In magnetically arrested disks (MAD), the poloidal is strong enough to dominate the disk dynamics and suppress the MRI.

In the disk initializations with an initial torus, which initially shed mass and stabilize in some post-relaxed state, it is impossible to a priori know the final mass accretion rate. In the thin disk setups, the disk mass accretion rate is a free parameter given at the start of the simulation. Material is fed into the disk at the outer radial boundary of the simulation domain. This enables an inflow equilibrium throughout the disk after (a successful) relaxation. With such a setup, it is easier to obtain the wanted solution and perform a case study or a wider parameter study.

Afterword

In two small volumes of "Thin accretion disks" Part I and II, I presented a fast-forward through the thin disk theory and numerical simulations.

As stressed earlier, despite being investigated for half a century, thin accretion disks are still evading full description. Analytical approximations provided initial insights, which subsequently needed to be refined, and are still failing to capture the wealth of information we are currently obtaining from the (increasing) multitude of measurements with ever improving instruments.

Numerical simulations are slowly maturing to the task, especially with inclusion of the radiative transfer, which is presented here only in passing, since for thin disks it is still in development.

I hope that the material presented here will ease further development in this fascinating and rewarding topic.

Bibliography

[BH91] Steven A. Balbus and John F. Hawley. *A Powerful Local Shear Instability in Weakly Magnetized Disks. I. Linear Analysis*. 376 (July 1991), p. 214 (cit. on p. 25).

[Čem19] M. Čemeljić. *"Atlas" of numerical solutions for star-disk magnetospheric interaction*. 624, A31 (Apr. 2019), A31. arXiv: 1811.02808 [astro-ph.SR] (cit. on pp. 9, 12, 24).

[ČPK17] M. Čemeljić, V. Parthasarathy, and W. Kluźniak. *MHD simulations of resistive viscous accretion disk around millisecond pulsar*. Journal of Physics Conference Series. Vol. 932. Journal of Physics Conference Series. Dec. 2017, 012028, p. 012028. arXiv: 1802.00261 [astro-ph.HE] (cit. on p. 5).

[DB19] Jason Dexter and Mitchell C. Begelman. *Extreme AGN variability: evidence of magnetically elevated accretion?* 483.1 (Feb. 2019), pp. L17–L21. arXiv: 1807.03314 [astro-ph.GA] (cit. on p. 25).

[FKR02] Juhan Frank, Andrew King, and Derek J. Raine. *Accretion Power in Astrophysics: Third Edition*. 2002 (cit. on p. 7).

[GMT03] Charles F. Gammie, Jonathan C. McKinney, and Gabor Toth. *HARM: A Numerical Scheme for General Relativistic Magnetohydrodynamics*. The Astrophysical Journal 589.1 (May 2003), pp. 444–457 (cit. on p. 27).

[Kit95] David B. Kita. "A Study of the Vertical and Radial Structure of Polytropic Accretion Disks". PhD thesis. THE UNIVERSITY OF WISCONSIN - MADISON., Jan. 1995 (cit. on p. 7).

[KK00] W. Kluzniak and D. Kita. *Three-dimensional structure of an alpha accretion disk*. arXiv e-prints, astro-ph/0006266 (June 2000), astro–ph/0006266. arXiv: astro-ph/0006266 [astro-ph] (cit. on pp. 7, 25).

[Kot+20] Aleksandra Kotek et al. *Asymmetric Jet launching*. XXXIX Polish Astronomical Society Meeting. Ed. by Katarzyna Małek et al. Vol. 10. Oct. 2020, pp. 275–278. arXiv: 2006.04083 [astro-ph.HE] (cit. on p. 10).

[Lan+19] Debora Lančová et al. *Puffy Accretion Disks: Sub-Eddington, Optically Thick, and Stable*. The Astrophysical Journal 884.2 (Oct. 2019), p. L37 (cit. on p. 25).

[LE74] Alan P. Lightman and Douglas M. Eardley. *Black Holes in Binary Systems: Instability of Disk Accretion*. 187 (Jan. 1974), p. L1 (cit. on p. 25).

[MČK20a] Ruchi Mishra, Miljenko Čemeljić, and Wlodek Kluźniak. *Backflow in simulated MHD accretion disks*. arXiv e-prints, arXiv:2012.13194 (Dec. 2020), arXiv:2012.13194. arXiv: 2012.13194 [astro-ph.SR] (cit. on p. 10).

[MČK20b] Ruchi Mishra, Miljenko Čemeljić, and Włodek Kluźniak. *Backflow in Accretion Disk*. XXXIX Polish Astronomical Society Meeting. Ed. by Katarzyna Małek et al. Vol. 10. Oct. 2020, pp. 147–150. arXiv: 2006.01851 [astro-ph.SR] (cit. on p. 10).

[MG04] Jonathan C. McKinney and Charles F. Gammie. *A Measurement of the Electromagnetic Luminosity of a Kerr Black Hole*. 611.2 (Aug. 2004), pp. 977–995. arXiv: `astro-ph/0404512 [astro-ph]` (cit. on p. 25).

[Mig+07] A. Mignone et al. *PLUTO: A Numerical Code for Computational Astrophysics*. 170 (May 2007), pp. 228–242. eprint: `astro-ph/0701854` (cit. on pp. 9, 11).

[Mis+20] Bhupendra Mishra et al. *Strongly magnetized accretion discs: structure and accretion from global magnetohydrodynamic simulations*. 492.2 (Feb. 2020), pp. 1855–1868. arXiv: `1907.08995 [astro-ph.HE]` (cit. on pp. 36, 38).

[Mis+22] Bhupendra Mishra et al. *The Role of Strong Magnetic Fields in Stabilizing Highly Luminous Thin Disks*. The Astrophysical Journal 939.1 (Oct. 2022), p. 31 (cit. on pp. 36, 38).

[Pir78] T. Piran. *The role of viscosity and cooling mechanisms in the stability of accretion disks.* 221 (Apr. 1978), pp. 652–660 (cit. on p. 25).

[Pow+99] K. G. Powell et al. *A Solution-Adaptive Upwind Scheme for Ideal Magnetohydrodynamics*. Journal of Computational Physics 154 (Sept. 1999), pp. 284–309 (cit. on p. 13).

[Sad+14] Aleksander S adowski et al. *Numerical simulations of super-critical black hole accretion flows in general relativity*. Monthly Notices of the Royal Astronomical Society 439.1 (Jan. 2014), pp. 503–520 (cit. on pp. 25, 28).

[SS73] N. I. Shakura and R. A. Sunyaev. *Black holes in binary systems. Observational appearance.* 24 (Jan. 1973), pp. 337–355 (cit. on pp. 7, 20, 25).

[Tan94] T. Tanaka. *Finite volume TVD scheme on an unstructured grid system for three-dimensional MHD simulation of inhomogeneous systems including strong background potential fields*. Journal of Computational Physics 111 (Apr. 1994), pp. 381–390 (cit. on p. 13).

[ZF09] C. Zanni and J. Ferreira. *MHD simulations of accretion onto a dipolar magnetosphere. I. Accretion curtains and the disk-locking paradigm*. 508.3 (Dec. 2009), pp. 1117–1133 (cit. on pp. 9, 15).

[ZS18] Zhaohuan Zhu and James M. Stone. *Global Evolution of an Accretion Disk with a Net Vertical Field: Coronal Accretion, Flux Transport, and Disk Winds*. 857.1, 34 (Apr. 2018), p. 34. arXiv: `1701.04627 [astro-ph.EP]` (cit. on p. 36).