Disk *Configuration of the gas disk physic.*
{
AspectRatio [optional, default:0.050000000000000003] *Disk aspect ratio*
FlaringIndex [optional, default:0] *Disk flaring index*
Boundaries *List of coma separated boundaries conditions handlers. Possible values are: evanescent evanescent2D evanescent2DforGiants evanescent3DforGiants open reflectingR reflectingZ.*
Transport *TBD*
HillCutFactor [optional, default:0.59999999999999998] *Apply Hill cut on a region of size HillCutFactor in unit of Hill radius.*
CFLSecurity [optional, default:0.5] *As a security check consider the timestep obtained with CFL condition normalized by a factor¡1 (suggested 1/D D=dimension of the disk)*
Grid *Description of the polar grid approximation of this disk. A two dimension (flat) grid is obtained by specifying a one layer (default).*
{
Sectors *Number of azimuthal sectors.*
Layers *Number of vertical layers.*
Sector *Azimuthal disk extension, default 2pi*
Half *(optional)* In 3D default is full disk, set this parameter to true to have only half disc (useful for vertical symmetric cases, i.e. non inclined planet)*
Opening [optional, default:1.3613568] *The colatitude angle in radians*
Radii *Number of rings.*
{
Min *Radial inner disk boundary*
Max *Radial outer disk boundary*
Spacing *Radial spacing type, can be *ARITHMETIC* (default), *LOGARITHMIC*, *FILE* (from a file).*
}
Cavity *TBD*
{
Radius *TBD*
Ratio *TBD*
Width *TBD*
}
}
Density *Configuration of the initial disk density radial profile*
{
Slope *Slope of the power law's surface density, default 0*
Minimum [optional, default:1e-25] *Volume (surface in 2D) density floor*
Start *Surface density value at r=1, code units (typical value 6.e-4, i.e.* $200 g/cm^2$ *in the default case with unit of distance R0=5.2AU)*
Cavity *Cavity modeled as in Robert Meheut 2020 2D case, here also modeled in 3D*
{
Radius [optional, default:0] *Maximum distance from the star at which the cavity is operating*
Ratio [optional, default:1] *Normalization factor applied to the surface density for* $r < Cavity.Radius$
Width [optional, default:1] *Width of the cavity.*
}
}
StarAccretion *Radial transport of gas to account for the observed stars' accretion rates.*
{
Type [optional, default:custom] *Choose CONSTANT for Mdot disk accrating at RATE, choose WIND for accretion occurring in the ionized layers*
Rate [optional, default:0] *accretion rate in Solar mass per year (typical values in the range 1.e-8-1.e-7)*
}
Viscosity *Model disk viscosity*
{
Artificial [optional, default:0] *add a term of artificial viscosity to avoid shocks.*
Type *Viscosity must be of type constant or alpha*
Value *Value of viscosity in code units (typical 1.e-5 for constant, 1.e-3 for alpha viscosity)*
}
Smoothing *Smoothing the planet's potential to avoid the singularity for r-¿0*

{
Change `[optional, default:0]` *if true: smoothly divide by 2 the smoothing lenght in a type provided by Smoothing.Taper*
Flat `[optional, default:0]` *true = define the planet potential as sqrt(GMp/(r+rsmoothing)), recommended in 2D simulations. False: the potential is set by a cubic function (recommended in 3D simulations)*
Size `[optional, default:0.5]` *size of the smoothing lenght in unit of Hill radius*
Taper `[optional]` *time in unit of 2pi for the change of the smoothing lenght by a factor of 2 (needs Change = true)*
}
Referential *Properties of the the referential frame*
{
Type *Referential frame, must be one of Constant (user specified constant) Circular3Body (match the constant speed of the first planet DEAD CODE TO BE REMOVED) CoRotating (match the speed of the first planet).*
IndirectForces `[optional, default:1]` *indirect forces act on planets and on disk because the system is centred on the star (non barycentric frame).*
Omega `[optional, default:0]` *reference frame's rotation frequency.*
NewOmega *TBD*
GuidingPlanet *If co-rotating, the name of the guiding planet).*
}
Velocities *Modify the initial velocity field*
{
AddRandomNoise `[optional, default:0]` *Add a random noise to the initial velocities.*
}
AdiabaticEoS `[optional, default:0]` *Adiabatic equation of state. If false, the code has a locally isothermal equation of state.*
{
Index `[optional, default:1.3999999999999999]` *Adiabatic Index.*
FullEnergyEoS `[optional, default:0]` *Full energy equation (compressional+viscous)heating+coolingterms)*
{
ZBoundaryTemperature `[optional, default:3]` *Temperature at the vertical disk boundaries.*
DustToGas `[optional, default:0.01]` *Set the dust to gas ratio*
SolidAccretion `[optional, default:0]` *With a full energy equation add the luminosity contribution from Solid accretion onto the planet (typical value 1.e27 erg/s)*
EnergyTimeStep `[optional]` *Desired times step for energy module, default is dynamical timestep computed with CFL condition*
StarRadiation `[optional, default:0]` *add energy irradiated from the star.*
{
TStar `[optional]` *Temperature of the star (kelvin, typical value 4370)*
RStar `[optional, default:1]` *Radius of the star (unit of Solar radius, default is 1, for a young sun choose 1.5)*
Epsilon `[optional, default:4.9999999999999998e-07]` *Desired control on the precision of implicitly computed energy*
ShadowAngle `[optional, default:0]` *Opening angle (degrees) for shadowed stellar radiation (recommended value is 7 degrees)*
}
OpacityLaw `[optional, default:BELL_LIN]` *With full energyequation of state opacity can be either constant (1) or prescribed by Bell and Lin tables (0); default is (0)*
{
Value `[optional, default:1]` *With constant opacity choose the value (default is 1 cm^2/g)*
}
}
}
}
Output *TBD*
{
TimeStep *TBD*
NbStep *TBD*
Trackers *TBD*
{

step_timer *TBD*
gas_tracer *TBD*
hdf5_logger *TBD*
{
modulo *TBD*
ofile_fmt *TBD*
verbose *TBD*
}
gas_txt_logger *TBD*
{
ofile *TBD*
}
}
}