`fargo2radmc3d` documentation

Clément Baruteau

1 What is `fargo2radmc3d`?

It is a python program that post-processes the results of Dusty FARGO-ADSG hydrodynamical simulations with the 3D radiative transfer code RADMC-3D (version 2.0, Dullemond et al. 2015). Dusty FARGO-ADSG is an extended version of the 2D hydrodynamical code FARGO-ADSG, which solves the gas hydrodynamical equations on a polar grid and models dust as Lagrangian test particles. In its current implementation, `fargo2radmc3d` uses the gas surface density as input to compute images of polarised scattered light or moment maps of molecular line emission, and the dust's spatial distribution obtained in the simulations as input to compute synthetic maps of continuum emission. The code works with both the 2.X and 3.X versions of python.

The paragraphs below explain how `fargo2radmc3d` converts the outputs of Dusty FARGO-ADSG simulations into inputs for RADMC-3D, and describe the various parameters of the configuration file `params.dat`. Since the code calls RADMC-3D for the radiative transfer calculations, a basic knowledge of RADMC-3D is definitely helpful. The reader is referred to RADMC-3D's manual. The file fargo2radmc3d.py is heavily commented, so the main steps of the code described below should be easily identified in the program.

2 From Dusty FARGO-ADSG outputs to RADMC-3D inputs

This section describes the different steps to convert the results of 2D Dusty FARGO-ADSG gas+dust hydrodynamical simulations into inputs for the 3D radiative transfer code RADMC-3D. This description closely follows § 2.2 of Baruteau et al. (2019) for dust radiative transfer calculations, and § 2.2.1 of Baruteau et al. (submitted) for gas radiative transfer calculations.

2.1 Spatial grid

The spatial grid used in RADMC-3D is a 3D extension in spherical coordinates of the 2D polar grid used in the hydrodynamical simulations. The structure of the simulations grid (size, number of cells in both the radial and azimuthal directions) is found by reading the simulations outputs. The vertical extent of the 3D grid on each side of the disc midplane is specified in units of the gas pressure scale height via `zmax_over_H` in params.dat, and the number of cells in colatitude by `ncol`. The gas pressure scale height is obtained via the .par parameter file used to run the Dusty FARGO-ADSG simulation, and which is always located in the simulations directory. The name of the simulations directory is set by `dir`, and the output number of the simulation is set by `on`. Grid cells are evenly spaced in colatitude, except for dust radiative transfer calculations of the continuum emission, for which a logarithmic spacing is assumed.

2.2 Gas radiative transfer calculations

To call a line radiative transfer calculation with RADMC-3D, `RTdust_or_gas` must be set to *gas*. The number density, temperature and velocity field (in physical units) of a gas species need to be specified. We describe how this is done in Sections 2.2.1 to 2.2.3. We then detail in Section 2.2.4 the parameters to be specified for RADMC-3D calculations.

2.2.1 Number density of the gas species

From the gas surface density $\Sigma_{\rm gas}$ in the simulation, we first infer the gas volume density $\rho_{\rm gas}$ on the 3D grid by assuming (i) vertical hydrostatic equilibrium, (ii) that the gas temperature is that in the hydrodynamical simulation (`Tdust_eq_Thydro` set to *Yes*) and (iii) is independent of altitude (z). The gas number density is then $n_{\rm gas} = \rho_{\rm gas}/\mu m_{\rm p}$ with $m_{\rm p}$ the proton mass and μ the mean molecular weight of the gas mixture (assumed to be 2.3). The number density of the gas species that is modelled is first taken to be proportional to the gas number density : $n_{\rm species} = \chi n_{\rm gas}$, where χ can be seen as the fractional abundance of the molecular species with respect to hydrogen nuclei. It is set by the parameter `abundance` in params.dat.

Photodissociation by UV irradiation can be accounted for in the calculation of $n_{\rm species}$. For this you need to set `photodissociation` to *Yes*. It is then simply modelled by dropping $n_{\rm species}$ by 5 orders of magnitude wherever the local gas column number density falls below 10^{21} cm^{-2} (see, e.g., Flaherty et al. 2015) :

$$\int_{z_0}^{\infty} n_{\rm gas}(R,z) dz < 10^{21}{\rm cm}^{-2}, \tag{1}$$

with

$$n_{\rm gas}(R,z) = \frac{\Sigma_{\rm gas}(R)}{\mu m_{\rm p}\sqrt{2\pi}H}\exp(-z^2/2H^2). \tag{2}$$

Using that

$$\int_{z_0}^{\infty} \frac{1}{\sqrt{2\pi}H}\exp(-z^2/2H^2)dz = \frac{1}{2}{\rm erfc}(z_0/\sqrt{2}H), \tag{3}$$

where erfc denotes the complementary error function (see, e.g., page 348 of Krumholz 2015), note that Eq. (1) is conveniently recast as

$$\frac{1}{2}{\rm erfc}(z_0/\sqrt{2}H) \times \frac{\Sigma_{\rm gas}(R)}{\mu m_{\rm p}} < 10^{21}{\rm cm}^{-2}. \tag{4}$$

Freezeout onto dust grains can also be accounted by setting `freezeout` to *Yes*. Right now, it only works for CO isotopologues and simply consists in dropping $n_{\rm species}$ by 5 orders of magnitude wherever the gas temperature falls below 20 K.

The calculation of the gas species' number density requires to set `recalc_density` to *Yes* in params.dat. It produces a rather large file, named numberdens_`gasspecies`.inp, which contains the number density for the gas species in every grid cell.

2.2.2 Temperature of the gas species

As said above, gas line radiative transfer calculations assume by default that the gas temperature is that in the hydrodynamical simulation. To compute the gas temperature, you only need to set `recalc_temperature` to *Yes*. It produces a rather large file, named gas_temperature.inp, which contains the gas temperature in every grid cell. There is therefore no preliminary Monte-Carlo thermal calculation to be carried out by RADMC-3D in order to obtain the gas temperature. This is something that could be changed, but it would need to modify the program.

2.2.3 Velocity field of the gas species

To expand in 3D the gas velocity field of our simulations, we simply assume that the radial and azimuthal components of the gas velocity are independent of altitude, and the vertical velocity component is zero. A rather large file, named gas_velocity.inp, is written, which contains the radial, azimuthal and latitudinal components of the gas velocity in every grid cell.

2.2.4 RADMC-3D parameters

Which gas species is modelled is set by `gasspecies`. This can be, for instance, *co*, *13co*, *hco+*, *c18o*... The rotation line is specified by `iline`. For instance, setting `iline` to 3 means that the J=3-2 rotational line emission is computed. How the level populations are computed depends on the parameter `lines_mode`. It set to 1, local thermodynamic equilibrium (LTE) is assumed and RADMC-3D computes the partition function for the level populations from the molecular data files provided by the Leiden LAMBDA database. The code automatically downloads the relevant files. For instance, if `gasspecies` is set to *c18o*, the code will find the file c18o.dat on the aforementioned database and rename it in a way that RADMC-3D can understand. Non-LTE calculations simply require to change the value for `lines_mode`. For instance, set it to 3 to use the Large Velocity Gradient approximation (Sobolev method), or to 4 to use the optically thin non-LTE population method in RADMC-3D. In non-LTE calculations, collisions are assumed to involve only H_2 molecules, and an additional file, named numberdens_h2.inp, is produced that contains the number density of H_2 molecules throughout the grid.

RADMC-3D computes the specific intensity of the gas line emission in `linenlam` channel maps covering a velocity range of $\pm$ `widthkms` around the systemic velocity (`widthkms` is positive and its value in km s^{-1}). Note that a single channel map can also be computed by putting `linenlam` to 1 and by specifying the velocity centroid in km s^{-1} via `vkms`. A turbulent broadening of the lines can be set by specifying `turbvel` in km s^{-1}. The calculation of the final gas image is described in Section 3.

2.3 Dust radiative transfer calculations

To call a dust radiative transfer calculation with RADMC-3D, `RTdust_or_gas` must be set to *dust*. Such calculations are used to produce images of polarized scattered light (if `polarized_scat` is set to *Yes*) or of continuum emission (`polarized_scat` set to *No*). For polarized scattered light, the dust's mass volume density is simply proportional to the gas mass volume density inferred from the simulation's data, whereas for continuum emission it is inferred from the dust's spatial distribution. This is detailed in Sections 2.3.1 and 2.3.2. We then describe in Sections 2.3.3 and 2.3.4 how the dust temperature and opacities are calculated.

2.3.1 Dust's surface density

Dust is modelled as Lagrangian test particles in Dusty FARGO-ADSG, so dust drag on the gas is discarded. For this reason, the mass of the dust particles in the simulation has no dynamical role, and only the spatial distribution of the dust particles matters for dust radiative transfer calculations of the continuum emission. This is why the simulations use an arbitrary size distribution between a minimum and a maximum sizes (the number density of super-particles is usually taken inversely proportional to dust size, so that there is approximately the same number of particles per decade of size).

The dust's size distribution $n(a)$ adopted in the radiative transfer calculations is a power-law function of particles size, $n(a) \propto a^{-p}$, between a minimum and a maximum sizes, which are set by `amin` and `amax` in params.dat (both values in metres). The opposite of the power law exponent p is set by `pindex` (it should be positive). The total dust mass is specified via the dust-to-gas mass ratio, which is denoted by `ratio` in params.dat. The mass of the disc gas is that in the hydrodynamical simulation at the desired output number (recall that the output number is set by `on`).

The dust's size range is decomposed into `nbin` logarithmically spaced size bins, and from the spatial distribution of the dust particles in the simulation we compute the dust's surface density for each size bin i, which we denote by $\sigma_{i,\mathrm{dust}}$. The quantity $\sigma_{i,\mathrm{dust}}$ can be expressed as

$$\sigma_{i,\mathrm{dust}}(r,\varphi) = \frac{N_i(r,\varphi)}{\mathcal{A}(r)} \times \frac{M_{i,\mathrm{dust}}}{\sum\limits_{r,\varphi} N_i(r,\varphi)}, \tag{5}$$

where N_i denotes the number of dust particles per bin size and in each grid cell of the simulation, $\mathcal{A}$ is the surface area of each grid cell, and $M_{i,\mathrm{dust}}$ is the dust mass per bin size, which takes the form

$$M_{i,\mathrm{dust}} = \xi M_{\mathrm{gas}} \times \frac{a_{i+1}^{4-p} - a_i^{4-p}}{a_{\mathrm{max}}^{4-p} - a_{\mathrm{min}}^{4-p}} = \xi M_{\mathrm{gas}} \times \frac{a_i^{4-p}}{\sum\limits_i a_i^{4-p}}, \tag{6}$$

where [a_i,a_{i+1}] is the size range of the i^{th} size bin, a_{min} and a_{max} are the minimum and maximum particle sizes, $-p$ is the power-law exponent of the dust's size distribution $n(a)$, M_{gas} the total mass of gas in the simulation, and ξ the dust-to-gas mass ratio.

When calculating polarised light images, the spatial distribution of the (small) dust particles is assumed to be the same as that of the gas in the simulation. In this case, the dust's surface density for each size bin simply reads

$$\sigma_{i,\mathrm{dust}}(r,\varphi) = \xi \Sigma_{\mathrm{gas}}(r,\varphi) \times \frac{a_{i+1}^{4-p} - a_i^{4-p}}{a_{\mathrm{max}}^{4-p} - a_{\mathrm{min}}^{4-p}} = \xi \Sigma_{\mathrm{gas}}(r,\varphi) \times \frac{a_i^{4-p}}{\sum\limits_i a_i^{4-p}}, \tag{7}$$

with Σ_{gas} the gas surface density in the simulations grid.

2.3.2 Dust's mass volume density

For the vertical distribution of the dust's mass volume density, hydrostatic equilibrium is assumed and for each size bin a Gaussian profile is adopted in which the dust's scale height $H_{i,\mathrm{dust}}$ of the i^{th} size bin is given by :

$$\begin{aligned} H_{i,\mathrm{dust}} &= H \times \left(\frac{\alpha}{\alpha + \mathrm{St}_i} \right)^{1/2} \quad \text{if } \texttt{z\_expansion} \text{ is set to T,} && (8) \\ & H \quad \text{if } \texttt{z\_expansion} \text{ is set to G,} && (9) \end{aligned}$$

where H denotes the gas pressure scale height, St_i the average Stokes number of the dust particles in the i^{th} size bin, and α the alpha turbulent viscosity in the hydrodynamical simulation (if a constant kinematic viscosity ν is used instead of an α-viscosity in the simulation, α is simply evaluated from ν). Note that `z_expansion` is automatically set to G if polarised images are computed. The dust's mass volume density is finally re-normalized such that the sum over the 3D grid's volume of the dust's mass volume density times the volume of each grid cell does give us the right total dust mass, equal to ξM_{gas} with above notations.

The calculation of the dust's surface and volume densities requires to set `recalc_density` to Yes in params.dat. It produces a rather large file, named dust_density.inp, which contains the dust's mass volume density for each size bin at every grid cell.

2.3.3 Dust's temperature

The usual practice in dust radiative transfer calculations is to compute the dust temperature via a thermal Monte-Carlo calculation. This is what is assumed by default, and in this case RADMC-3D produces a big binary file, named dust_temperature.bdat, which contains the dust temperature for each size bin and in each grid cell. However, you can still choose to take for the dust temperature that in the hydrodynamical simulation, independently of the dust size. This requires to set both `Tdust_eq_Thydro` and `recalc_temperature` to *Yes*.

2.3.4 Dust opacities

Dust opacities are computed via the python scripts makedustopac.py and bhmie.py, written by C. Dullemond, and based on the original Mie code by Bruce Draine. For continuum emission maps, absorption opacities, scattering opacities for anisotropic scattering and the mean scattering angle are computed as a function of wavelength and stored in dustkappa*.inp files. For polarised light images, the scattering matrix is also computed in the dustkapscatmat*.inp files. The optical constants for various dust compositions can be found in the archive opac.tar.gz (files with a .lnk extension) :

- mix_2species_60silicates_40carbons.lnk : compact dust particles made of 60% astrosilicates and 40% amorphous carbons (internal density is 2.7 g cm^{-3}),
- mix_2species_ice70.lnk : compact dust particles comprised of 70% water ices and 30% astrosilicates (internal density is 1.3 g cm^{-3}),
- mix_2species_porous_ice70.lnk : porous dust particles with an internal density of 0.1 g cm^{-3}. The dust is actually a mixture of a silicate matrix, water ices and a vacuum inclusion. For the mix aggregate to have 70% of its solids in water ices and 30% in silicates, the level of porosity (volume fraction of vacuum) needed to produce grains with a density of 0.1 g cm^{-3} is $\sim$92%. The Bruggeman rules are applied to compute the optical constants of the dust mixture.

The optical constants of water ices are obtained from the Jena database, those of astrosilicates are from Draine & Lee (1984), and those of amorphous carbons are from Li & Greenberg (1997).

The calculation of the dust opacities requires to activate `recalc_opac` in params.dat. The choice of the dust composition is set by `species`, and the directory that contains the optical constant files is specified via `opacity_dir`.

2.4 Call to RADMC-3D

Call to RADMC-3D requires `recalc_radmc` to be set to *Yes*. For dust radiative transfer calculations, this is usually done in two steps : the dust temperatures are computed first with a thermal Monte-Carlo calculation, and then the continuum or polarised light images are computed by ray-tracing. Before actually executing the radiative transfer calculation, `fargo2radmc3d` writes a number of input parameter files needed by RADMC-3D, which requires to specify the following parameters in params.dat :

- the star's radius (`rstar`, in Solar radii) and effective temperature (`teff`, in Kelvin),
- the disc's inclination (`inclination`), position angle (`posangle`) and phi angle (`phiangle`), with all three angles in degrees (the phi angle corresponds to a rotation in the disc plane, while the position angle corresponds to a rotation in the image plane),
- the number of pixels in the image (`nbpixels`),
- the number of cores to be used by RADMC-3D for the radiative transfer calculation (`nbcores`),
- for dust radiative transfer calculations, we also seed to specify (i) the wavelength at which the continuum or polarised light images are to be computed (via `wavelength`, quantity in millimetres), (ii) the number of photon packages (`nb_photons`) and that for scattering (`nb_photons_scat`), and (iii) how RADMC-3D will treat scattering, via the `scat_mode` parameter. When set to 0, dust scattering is discarded (only thermal absorption is included). When set to 2, anisotropic scattering is included via the Henyey-Greenstein approximation for the scattering phase function. When set to 5, a full treatment of polarisation is adopted using the dust's scattering matrix.

2.5 Miscellaneous options

Further options are available in the params.dat parameter file, which are briefly listed below :

- `xaxisflip` : option to add 180 degrees to the disc inclination (can be useful if the disc that you model rotates clockwise in the observations, since the disc will always rotate counter-clockwise in the simulations – this is the case for instance of the MWC 758 disc).

- override_units : by default, fargo2radmc3d looks for a file named 'units.dat' in the simulation directory to get the code units used in the simulation. It is possible to change the units, in particular for gas radiative transfer calculations. For fargo2radmc3d to override the units used in the simulation, you need to specify override_units to *Yes*, and provide new international system units : the new code's unit of length (new_unit_length) and/or that of mass (new_unit_mass). The corresponding units of time and temperature follow by the assumption that the gravitational constant is equal to unity in the code, and that of temperature by setting the universal gas constant equal to unity and by specifying a mean molecular weight μ for the gas, which is taken equal to 2.3 (corresponding to a solar nebula H_2-He gas mixture).
- plot_opac : option to plot the absorption and scattering opacities as a function of particle size at the desired wavelength.
- plot_tau : option to plot an image of the (absorption) optical depth of the dust or of the gas, instead of a specific intensity of emission.
- secondorder : use (or not) of second-order integration for ray tracing in RADMC-3D.
- recalc_rawfits : option to re-compute the final image if the raw image produced by RADMC-3D (image.out) has been calculated on a machine or a computing cluster with no python support (RADMC-3D's execution can be done via the script script_radmc). It is also useful if you would like to add noise to the specific intensity of emission (see Section 3).
- check_beam : option to check in the final image that the beam has the expected shape and size by adding a point-like source at the origin of the image grid.
- plot_density : option to plot the azimuthally-averaged number density of the gas species as a function of spherical radius and colatitude.
- plot_temperature : option to plot the azimuthally-averaged radial profile of the temperature of the largest dust particles at the surface and the midplane.
- mask_radius (arcseconds) : option to truncate the dust density within mask_radius of the star for polarised intensity images.
- truncation_radius (arcseconds) : option to reduce the dust density beyond truncation_radius for polarised intensity images (by a power-law function of r which can be changed).

3 Computation of the final image

The computation of the final image requires to set recalc_fluxmap to *Yes*. This is automatically the case when RADMC-3D is executed (via recalc_radmc). The disc distance needs to be specified in parsecs via distance.

3.1 Gas line emission

For gas line radiative transfer calculations, RADMC-3D computes a datacube with the specific intensity of the line emission in every velocity channel. Optional noise can be added in the channel intensities if add_noise is set to *Yes*. When noise is included, random numbers with a Gaussian probability distribution of zero mean and standard deviation σ are added in each channel map prior to beam convolution. After beam convolution, each convolved channel map then includes white noise with a spatial scale that is similar to the beam size. The standard deviation in Jy/beam per channel map is specified via noise_dev_std. Whether noise is included or not, fargo2radmc3d will compute bean-convolved moment maps of the line emission, with the moment order being set by moment_order. If it is equal to 0, fargo2radmc3d will compute an integrated intensity map of the line emission (zeroth moment) in mJy/beam km/s or μJy/beam km/s. If it equals 1, fargo2radmc3d will compute a rotational map (first moment) in km/s. Other moment orders are not available yet. Regarding the elliptical beam, its major axis (bmaj) and minor axis (bmin) are set in arcseconds in params.dat, and the beam's position angle (bpaangle) in degrees. The final image is saved both as .pdf and .fits files.

Note also that fargo2radmc3d stores the full datacube produced by RADMC-3D in a big .fits file, which can be used as entry for a third-party program to compute moment maps of the gas emission, like for instance the python code bettermoments (Teague & Foreman-Mackey 2018).

3.2 Dust continuum emission

When computing a dust continuum image, RADMC-3D calculates the raw flux of emission at the desired wavelength, and fargo2radmc3d then convolves the resulting image by an elliptical beam. Like for the gas line emission, the beam's major axis (bmaj), minor axis (bmin) are set in arcseconds in params.dat, and the beam's position angle (bpaangle) in degrees. The convolved flux image is returned in units of mJy/-beam or μJy/beam. It is saved as .pdf and .fits files.

Like for the gas line emission, synthetic images of dust continuum emission can include noise if add_noise is set to *Yes* in params.dat. This is done by adding white noise to the raw maps of continuum emission. More specifically, at each pixel of the raw maps a random number is added that follows a Gaussian probability distribution with zero mean and given standard deviation. The noise's standard deviation is set in Jy/beam by noise_dev_std.

As already said in Section 2.5, an image of the optical depth of continuum emission can be computed instead of the flux. This can be done by setting plot_tau to *Yes*. If scat_mode is set to 0, the absorption optical depth will be calculated. If it set to 2, the total (absorption + scattering) optical depth will be calculated.

3.3 Polarised scattered light

When computing a polarised light image, RADMC-3D calculates the raw Stokes maps I, Q, U and V, and then convolves the Stokes maps Q and U, which represent linear polarised intensities, by an elliptical beam. Like for continuum emission maps, the beam's major axis (bmaj), minor axis (bmin) need to be set in arcseconds in params.dat, and the beam's position angle (bpaangle) in degrees. Next, the convolved Stokes maps are post-processed to obtain the local Stokes Q_ϕ following the procedure described in Avenhaus et al. (2017). Each pixel of the Q_ϕ synthetic image is finally scaled with the square of the deprojected distance from the central star, and normalised such that the intensity of the strongest pixel is 1. The final image is saved as .pdf and .fits files.

White noise can also be included via the add_noise parameter. If so, at each pixel of the Stokes maps Q and U, random numbers are added that have Gaussian probability distributions with zero mean and a standard deviation of 0.4% the maximum value of each map (this value should be changed to reflect the desired rms noise level in the final synthetic map).

3.4 Deprojected images

By default, the synthetic images of line emission, dust continuum emission or of polarised light are displayed in the sky plane, and the x- and y-axes show the offset from the stellar position in the right ascension (RA) and declination (Dec) in arcseconds (i.e., north is up and east is to the left). The maximum value for x and y is set by minmaxaxis in params.dat, in arcseconds. The synthetic images can also be deprojected on the disc plane, by setting deproj_polar to *Yes*. If so, the image will show orbital radius in arcseconds in $y-$axis, with a maximum value of minmaxaxis, and position angle in $x-$axis.

4 Analytical expression for the continuum emission flux

Instead of solving the full radiative transfer equation in 3D with RADMC-3D, one can compute the analytical solution to the dust radiative transfer equation in 2D without scattering. This can be done with fargo2radmc3d by setting calc_abs_map to *Yes* in params.dat. In each cell of the simulation grid, the specific intensity $\mathcal{I}_\nu$ is calculated as

$$\mathcal{I}_\nu(r,\varphi) = \mathcal{B}_\nu(T_{\rm dust}) \times (1 - e^{-\tau}), \tag{10}$$

with τ the absorption optical depth, $\mathcal{B}_\nu$ the Planck function, and $T_{\rm dust}$ the dust's temperature, which is taken equal to the gas temperature in the simulation if Tdust_eq_Tgas is set to *Yes*. Otherwise, the code will look for the local temperature file dust_temperature.bdat computed by RADMC-3D and take the midplane temperature of the largest dust particles. Upon discretization of the dust's size range, just like for the calculation of the dust's surface density in Sect. 2.3.1, τ may be written as

$$\tau(r,\varphi;\lambda) = \sum_i \sigma_{\rm i,dust}(r,\varphi)\kappa_i(\lambda)/\cos(i), \tag{11}$$

where κ_i and $\sigma_{\rm i,dust}$ denote the absorption opacity and the surface density of dust particles in the $i^{\rm th}$ size bin, respectively. The quantity $\sigma_{\rm i,dust}$ is given by Eq. 5, except if dustdens_eq_gasdens is set to *Yes* in params.dat, in which case it is given by Eq. 7. The quantity κ_i is read from the local dustkappa*.inp files, except if precalc_opac is set to Yes, in which case the code will read the pre-calculated opacity files located in the opacity_dir directory.

The specific intensity is then projected on the sky plane, by performing first a rotation in the disc plane by phiangle, second a rotation along the line of sight by inclination, and third a rotation by posangle in the image plane. The specific intensity thus obtained is multiplied by the solid angle that subtends each pixel of the image plane. White noise can be added to the image. The resulting raw flux map is finally convolved by an elliptical beam (just like in Sect. 3). The final convolved image is saved as .pdf and .fits files, and can be further deprojected on the disc plane if deproj_polar is activated.

5 Contributing authors and acknowledgments

The code fargo2radmc3d has been written by Clément Baruteau, Sebastián Pérez, Marcelo Barraza and Gaylor Wafflard-Fernandez, with substantial contributions from Simon Casassus. If you use fargo2radmc3d in your publications for dust radiative transfer calculations, please cite Baruteau et al. (2019). If you use it for gas radiative transfer calculations, please cite Baruteau et al. (submitted).

⋆⋆⋆