Documentation for rac2dm

Fujun Du
fujun.du@gmail.com

October 16, 2018

Contents

1 Guideline

The rac2dm package is designed to model a static disk structure and its thermochemistry. It is derived from the rac2d package. The letter "m" in the name stands for "modularized". Different modules of this package are designed to be as independent to each other as possible. For the ease of implementation, the interface to each module is implemented with files. File IO is slow, but calculations usually take more time, hence IO should not be a bottleneck. Using script languages such as `Python`, different modules can be chained and iterated to achieve a specific goal.

2 Description of modules

2.1 Radiative transfer and bulk radiation field solution: `rtrf`

2.1.1 Overall configuration

```
&conf_nml_rt
  conf_rt%struct_file = "struct.dat"
  conf_rt%opti_files = "sil.dat"  "gra.dat"
  conf_rt%rad_source_files = "s1.dat"  "s2.dat"
  conf_rt%star_mass_Msun = 0.6D0
  conf_rt%nphoton = 1000000
  conf_rt%nmax_encounter = 1999999999
  conf_rt%randnumber_save_file = "rd_saved.dat"
  conf_rt%escaped_photons_file = "escaped_ph.dat"
  conf_rt%collect_photon   = .true.
  conf_rt%collect_lam_min_AA  = 1D0
  conf_rt%collect_lam_max_AA  = 1D8
  conf_rt%collect_nmu      = 3
  conf_rt%collect_ang_mins_deg = 0D0 4D0  80D0
  conf_rt%collect_ang_maxs_deg = 3D0 10D0 90D0
  conf_rt%nlen_lut         = 4096
  conf_rt%Tdust_min_K      = 1D0
  conf_rt%Tdust_max_K      = 5D3
  conf_rt%dist_to_Earth_pc = 51D0
/
```

Notes:

- Each type of dust grain has a separate entry in the `opti_files` list.

2.1.2 Specification of matter distribution: `struct_file`

```
format ! Default to 1
number_of_species  number_of_data_points
r1  z1  n11  n21  n31 ...
r2  z2  n12  n22  n32 ...
...
```

Notes:

- Here `nij` means the density of the ith species at the jth location.
- A species does not have to be a type of dust grain; it can also be a molecule, as far as the right optical file is used.

2.1.3 Specification of optical properties: `opti_files`

```
number_of_data_points
angstrom1  Q_abs1  Q_sca1  g1
angstrom2  Q_abs2  Q_sca2  g2
...
```

2.1.4 Specification of radiation sources: `rad_source_files`

Multiple radiation sources are allowed. Each source can have a different type. At present the following types are supported:

- `point`: point source
- `sphere`: photons are emitted from the surface of a sphere
- `isotropic`: photons are emitted from an isotropic background
- `directional`: photons are emitted from a single direction

The following are example radiation source files:

```
point
x_au  y_au  z_au
number_of_data_points_of_spectrum total_luminosity_cgs
angstrom1  F1_cgs
angstrom2  F2_cgs
...
```

```
sphere
x_cen_au  y_cen_au  z_cen_au  r_sphere_Rsun
number_of_data_points_of_spectrum total_luminosity_cgs
angstrom1  F1_cgs
angstrom2  F2_cgs
...
```

```
isotropic
! Empty line
number_of_data_points_of_spectrum energy_density_cgs
angstrom1  F1_cgs
angstrom2  F2_cgs
...
```

```
directional
v_x  v_y  v_z
number_of_data_points_of_spectrum flux_density_cgs
```

```
angstrom1  F1_cgs
angstrom2  F2_cgs
...
```

2.1.5 Output

- Temperature distribution
- Internal radiation field
- Escaped radiation field

2.2 Disk structure adjustment: `structadj`

2.2.1 Overall configuration

```
&conf_nml_struct_adj
  conf_struct_adj%density_file = "dens.dat"
  conf_struct_adj%density_file_new = "dens_new.dat"
  conf_struct_adj%temperature_file = "temperature.dat"
/
```

2.2.2 Output

- New density distribution

2.3 Thermochemical calculation: `thermochem`

Here we assume the thermochemical calculation is only done for a single cell. To calculate for the whole disk just call the module in a loop or use the `map` function in `Python`. Note that mutual and self-shielding effects are non-local in nature and must be treated with care in the big loop.

2.3.1 Overall configuration

```
&conf_nml_thermochem
  conf_thermochem%network_file = "rate12.dat"
  conf_thermochem%init_file = "init.dat"
  conf_thermochem%enthalpy_file = "enthalpy.dat"
  conf_thermochem%ngas = 1D9
  conf_thermochem%Tgas = 20D0
  conf_thermochem%Tdust = 20D0
  conf_thermochem%sigdust_cm2 = 20D0
  conf_thermochem%G0 = 1D0
  conf_thermochem%Xray = 0D0
```

```
  conf_thermochem%zeta_cosmicray_H2 = 1.36D-17
  conf_thermochem%PAH_abundance = 1.6D-9
  conf_thermochem%Av = 10D0
  conf_thermochem%filename_CII = ""
  conf_thermochem%filename_OI = ""
  conf_thermochem%filename_NII = ""
  conf_thermochem%filename_SiII = ""
  conf_thermochem%filename_FeII = ""
  conf_thermochem%dt_first_step = 1D-8
  conf_thermochem%t_max = 1D6
  conf_thermochem%ratio_tstep = 1.1D0
  conf_thermochem%max_runtime_allowed = 60D0
  conf_thermochem%RTOL = 1D-6
  conf_thermochem%ATOL = 1D-30
/
```

Notes:

- Need time dependence of physical parameters? The most flexible way to do it may be to restart with new physical parameters at different time steps, which admittedly will be slower than the "built-in" approach.

2.3.2 Output

- For a single location, output the gas temperature and chemical abundances as a function of time.
- Optional output

2.4 Make images: `makeimage`

2.4.1 Overall configuration

```
&conf_nml_image
  conf_img%molecule_density_file = "mol_dens.dat"
  conf_img%Tgas_file = "Tgas.dat"
  conf_img%continuum_file = "cont.dat"
  conf_img%transition_file = "12C16O_H2.dat"
  conf_img%database_name = "lamda"
  conf_img%molecule_name = "CO"
  conf_img%useLTE = .true.
  conf_img%image_type = "cube"
  conf_img%out_dir = "images"
  conf_img%maxx = 200D0
  conf_img%maxy = 200D0
  conf_img%nx = 401
  conf_img%ny = 401
```

```
  conf_img%nf = 201
  conf_img%nfreq_window = 3
  conf_img%freq_mins = 0.9D11  3.4D11  6.90D11
  conf_img%freq_maxs = 3.0D11  3.6D11  6.92D11
  conf_img%E_min = 0D0
  conf_img%E_max = 3D3
  conf_img%n_theta = 1
  conf_img%view_thetas = 7D0
  conf_img%dist_to_Earth_pc = 51D0
  conf_img%vel_width_kms = 30D0
/
```

2.4.2 Output

- 3D data cube, 2D image, or 1D spectrum