

INTRODUCTION

Physical conditions in nebulae (planetary nebulae, HII regions etc.) are usually obtained using so called diagnostic methods, that allows to obtain average the electron temperature T_e and density N_e from different emission line intensities ratios (see, f.e., DIAGN method [1]). However, in case of diagnostic methods obtained T_e and N_e are constant within investigated ionization zone. For detailed analysis of the ionization structure of nebular environment the transfer of the ionizing radiation taking into account all elementary processes in nebular plasmas important for such transfer should be calculated. Such calculations are performed during PhM of nebular environment.

In case of nebulae with compact ionization source (f.e star, compact stars cluster, etc) radiation can be separated into two components:

1. Direct component, coming from ionizing source (star or star formation knot);
2. Diffuse component, originating from nebular gas.

While transfer of the direct component of the ionizing radiation can be easy calculated (because of no source terms), the diffuse radiation transfer calculation is a very time-consuming even for modern supercomputers, since it requires iterative process of ionizing radiation flux integration over volume in each modelling elementary volume of nebular environment. For time economy purposes approximate methods of diffuse radiation calculation are frequently used. The most popular are QTS approximation, that assumes that ionizing radiation is consumed into the same volume where it was emitted, and Outward Only approximation, that assumes that ionizing radiation is propagated in radial direction, from ionizing source to outer surface of object. However, QTS method is good only in case of optically thick objects. Outward only approximation is more precise, however it can be inaccurate in case of objects with complicated morphology. For example, dense clumps can cause the origination of shaded regions behind of them. Non-radial DIR can play main role in formation of the ionization structure in these shaded regions.

To avoid usage of approximate methods for DIR calculation and reduce time, required for PhM we developed DiffRay3D) software, that allows to integrate radiation fluxes using 2D or 3D emissivity maps, and, as result, to implement method for detailed diffuse radiation calculation, described in [2] and displayed in Figure 1.

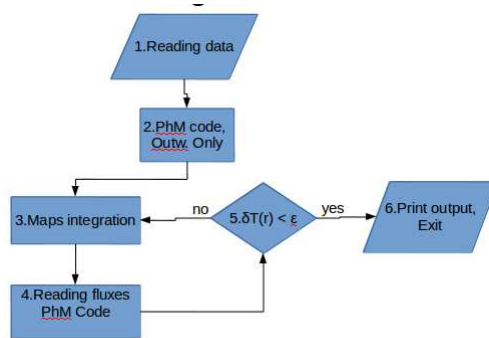


Figure 1: Our approach algorithm

- 1 - Reading data, such as required precision, maximum iterations, etc.
- 2 - Running PhM code in Outward Only mode, generating and saving emissivity and opacity maps.
- 3 - Running integration procedure using previously generated maps.
- 4 - Running PhM code, using fluxes, obtained after step 3. Generating and saving new emissivity and opacity maps.
- 5 - Check difference between electron temperatures obtained from current iteration and the previous one. Once difference is greater than is required by precision, pointed in 1, then returning to step 3.
- 6 - Generating output fluxes and ionization structure, obtained from PhM on last iteration.

1 DiffRay3D. General algorithm and limitations

The current version of DiffRay3D provides tools for integrating radiation fluxes using 2D maps over a 3D volume (support for 3D maps will be implemented in future versions). The nebula is divided into 80 sectors (20 per quarter, assuming cylindrical symmetry, and physical conditions in each quarter are the same). Each sector is divided into slabs. The shell structure is defined from input emissivity/opacity maps.

For the implementation of the diffuse radiation fluxes integration procedure, the DiffRay code was developed. The code's algorithm is following:

- 1 - Setting up initial integration steps for angles, reading emissivity and opacity maps that should be integrated.
- 2 - Integration of diffuse radiation fluxes for each elementary volume of the nebula environment.
- 3 - Comparing the difference between fluxes obtained on the current iteration and the previous one. If it fits the required precision, proceed to step 5. Otherwise, decrease the integration step (4), and return to step 2.
- 5 - Print fluxes in a format that can be read by the PhM code. Exit.

2 How to run the code

To run the code, you need a working compiled code version and valid object data with a commands file provided.

Download the code archive. Unzip files in any folder you wish. DiffRay3D can be started in any folder, the only requirement is that the code should have sufficient permissions to create/delete folders and files. Once unzipped, go to the "source/v*..." folder and run MakeFile.all. This will create the DiffRay3D.exe executable in the installation root folder.

3 Input data format

To function correctly, the code must have information about the object's geometry, emissivity, and opacity maps, and so on. This information is read from the data folder. In general, the filename looks like:

Emis_Lines_SectorNo{**Sector number**}_Age{**Object age**}Myr.dat

For example, Emis_Lines_SectorNo7_Age10.00Myr.dat

Here, the Sector number is the number of sectors represented by the file. The Age can be used as any identifier for the model data set. (For example, once you have maps for several models, you can enumerate them and use this enumeration in the age field.

A complete list of supported data files and their format can be found in the following subsections.

3.1 Continuum Mesh

Continuum{SectorNo}_Last_Age{ObjectAge}.dat is required file with the continuum energy mesh structure. Each row should contain, in the first column, continuum cell energy in Rydbergs. The structure of the file is as follows:

```
cell 1 energy ...  
cell 2 energy ...  
cell 3 energy ...  
cell 4 energy ...  
...          ...
```

Continuum cell energy should be represented in Rydberg units. The file can contain an arbitrary amount of columns, but only the first one is used, while the rest will be ignored. If you want to change the reading behavior, see the *CContinuum::readMesh* method in the "continuum.cpp" file.

3.2 Incident Continuum

Continuum{SectorNo}_SB99_Age{ObjectAge}.dat is required file with the incident continuum of the ionizing radiation source. Each row contains continuum cell energy in Rydbergs in the first column, and incident continuum flux ($\text{erg} \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$) in the second one. The structure of the file is as follows:

```
cell 1 energy  flux
cell 2 energy  flux
cell 3 energy  flux
cell 4 energy  flux
...           ...
```

If you want to change the reading behavior, see the *CContinuum::readInwardFluxes* method in the "continuum.cpp" file.

3.3 Continuum emissivities map

Emis_Cont_SectorNo{SectorNo}_Age{ObjectAge}.dat is required file with continuum emissivity data. Each row corresponds to sector slab. First row contains only the value of inner radius. Next rows contain slab outer radius in first column, and net volume emissivity values for each energy cell in corresponding slab in following columns. Note, that amount of columns in this file should be equal to amount of rows in

Continuum{SectorNo}_Last_Age{ObjectAge}.dat + 1 column for radius. The structure is following:

```
inner radius
Radius 1    emissivity  ...
Radius 2    emissivity  ...
Radius 3    emissivity  ...
...         ...         ...
```

Slab radius should be specified in centimeters. Emissivities are in $\text{erg} \cdot \text{cm}^{-3} \cdot \text{sec}^{-1}$). If you want to change reading behaviour, see *CContinuum::readEmissivity* method in "continuum.cpp" file.

3.4 Continuum opacities map

Opac_Cont_SectorNo{SectorNo}_Age{ObjectAge}.dat is required file with opacity data for each continuum cell. Each row corresponds to a sector slab. The first row has only one value - the inner radius. Subsequent rows contain the slab's outer radius in the first column, and opacity values for each energy cell in the following columns. This file is used both for calculating continuum and line optical depths. The structure is as follows:

```
inner radius
Radius 1    opacity  ...
Radius 2    opacity  ...
Radius 3    opacity  ...
...         ...         ...
```

Slab radius should be specified in centimeters. Opacities are expressed in cm^{-1} . If you want to change reading behavior, see the *CContinuum::readOpacity* method in the "continuum.cpp" file.

3.5 Lines Emissivity

Emis_Lines_SectorNo{SectorNo}_Age{ObjectAge}.dat is required file with lines emissivity data. Each row corresponds to a sector slab. The first two rows have headings, according to the standard Cloudy "punch line emissivity" command, [3]. Subsequent rows contain the slab's outer radius in the first column, and emissivity values for each emission line in the following columns. The structure is as follows:

heading			
Col. label	line name	line name	...
Radius 1	emissivity	emissivity	...
Radius 2	emissivity	emissivity	...
Radius 3	emissivity	emissivity	...
...

While slab radius should be specified in centimeters, the emissivities are represented in $\text{erg} \cdot \text{cm}^{-3} \cdot \text{sec}^{-1}$. If you want to change reading behavior, see the `CLines::readLines` method in the "lines.cpp" file.

Important Note: You can specify only lines defined in the known lines database. To check if emission line exists and is supported, you can search the `data/database/line_ch.data` for the line label or in `data/database/line_en.data` for energy in Rydbergs. If you cannot find the desired emission line, you can add it by editing these files. However, if you edit these files, don't forget to copy them before you update DiffRay. Otherwise, the data you changed will be overwritten during an update.

3.6 Chemical Abundance

Abund{SectorNo}_Age{ObjectAge}.dat is a file with abundances of the chemical elements. Each row corresponds to a sector slab. The first row should have valid column headings with element names, provided as the first four symbols of the full name. (`#Depth H HELI LITH BERY...`, Garry Ferland's Cloudy output format, see [3]). Subsequent rows contain the slab's outer radius in the first column, and relative abundances for each chemical element in the following columns. The structure is as follows:

Col. label	Element 1 label	Element 2 label	...
Radius 1	Abundance Value	Abundance Value	...
Radius 2	Abundance Value	Abundance Value	...
Radius 3	Abundance Value	Abundance Value	...
...

Slab radius should be specified in centimeters. Abundances are represented as $\log(n(\text{Element})/n_{\text{Total}})$, where $n(\text{Element})$ represents the total chemical element's density, n_{Total} is total density. If you want to change reading behavior, see the `Abund::getElementList` method in the "abund.cpp" file.

3.7 Physical Conditions Overview File

Overview{SectorNo}_Age{ObjectAge}.dat is a file with additional data about physical conditions. Each row corresponds to sector slab. The first row is a heading. (`#Depth Te Htot hden eden ...`, Garry Ferland's Cloudy output of the *punch overview* format, see [3]). Subsequent rows contain the slab's outer radius in the first column, and corresponding values, like electron temperature, electron density, and so on. The structure is as follows:

Heading				
Radius 1	Electron Temperature	Heating Rate	Density	Electron Density
Radius 2	Electron Temperature	Heating Rate	Density	Electron Density
Radius 3	Electron Temperature	Heating Rate	Density	Electron Density
...

Slab radius should be specified in centimeters. Physical values are mostly represented on a logarithmic scale. If you want to change reading behavior, column order, and so on, see the `Physics::readPhysics` method in the "physics.cpp" file.

4 Commands file

Once you have compiled the code and prepared the data files, you should write some instructions for how and what the code should calculate. These instructions are contained in the `commands.ini` file in the installation root directory.

4.1 IO settings

- Set the path to the folder where output files will be located:
print directory <path to your output folder, relative to the installation root directory>
- Set the path to the folder with input data:
input directory <path to input data files folder, relative to the installation root directory>

Example:

```
print directory result/ngc/ism0/140
input directory data/ngc1569nograin
```

Please note that paths should be provided without quotes!

4.2 Object geometry and distance

Commands that control the object's position in space.

- Set the distance to the object:
distance <log(dist)> <units>
Supported units are meters, kilometers, centimeters.
- Set the azimuthal coordinate for the destination point wherein the fluxes should be integrated :
object phi <azimuthal coordinate, in radians>
- Set the equatorial coordinate for the destination point wherein fluxes should be integrated :
object theta <azimuthal coordinate, in radians>
- Set the object age :
object age <object age, Myr>
Currently, it doesn't have any impact on calculations and can be used as models bundle identifier

Example:

```
distance 19.75 cm
object phi 0.76
object theta 0.21
object age 50
```

4.3 Calculation and integration options

Commands that control integration options and customize calculations.

- Set the maximum allowable number of iterations in the iterative process of searching for the optimal integration step of ionizing radiation flows:
integration maxiterations <x>
- Set the maximum permissible deviation between the integral flows of ionizing radiation in adjacent directions of integration (if the deviation exceeds the specified value, the integration step by angles will be reduced):
integration precision <precision>
Here precision 0.1 means 10%
- Turn on an additional iteration that runs before the main process of splitting the object into spatial angles and solving radiative transfer for those:
integration predictive <on / off>
It is recommended to turn on only in the case of an object with complex geometry and/or significant spatial inhomogeneities.

- Turn on/off opacity calculation (turning this off assumes that the environment is very optically thin for all quanta):
calculation opacity <on/off>
- Turn on/off radiative transfer for the continuum. Turning off will make DiffRay calculate only line fluxes:
calculation continuum <on/off>
- Turn on/off calculation of weighted elemental abundances within the volume cut by the aperture/along the observer sightline. Once you turn this on, abundance data files are required (See 3.6 for details):
calculation abundance <on/off>
- Turn on/off calculation of average electron temperature and density within the volume cut by the aperture/along the observer sightline. Once you turn this on, overview data files are required (See 3.7 for details):
calculation overview <on/off>

Example:

```
integration precision 0.005
integration maxiterations 12
integration predictive off
calculation opacity on
calculation abundance on
calculation overview off
```

4.4 Isophotes

Prints a map of fluxes for a selected line within a square cut by the aperture.

This can be useful to visualize the spatial distribution of modeling volume brightness in specific emission lines.

- Print a map of absolute fluxes for a selected line:
isophote line <line label>
- Print a map of relative fluxes for a selected line ($F(\text{line})/F(H_\beta)$):
isophote ratio <line label>

4.5 Aperture

Adds an aperture to the list of apertures to be calculated. Here *phi* - azimuthal aperture position, *theta* - vertical position relative to the ray from the observer to the center of the object. *w*, *h* - aperture width and height. All values should be provided in radians.

- Print a map of absolute fluxes for a selected line:
aperture phi theta w h

Example:

```
aperture add 0 0 0.000014537 0.000014537
aperture add 0.000464 0.0000139 0.000014537 0.000014537
aperture add -0.000271 -0.0000102 0.000014537 0.000014537
```

In this example, the code will run calculations for 3 apertures, and put results in the *app{N}* folder. N is the aperture number here. Aperture folders are created automatically in the output directory specified with the output command.

5 Output

Output files will be stored in the output directory specified in the corresponding commands file command.

5.1 Emission Lines Results

linesarr.dat is a file containing fluxes for each line specified in the line emissivities input file (see 3.5). Each row contains four columns: Line label, Flux at observer ($\text{erg} \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$), Luminosity ($\log(\text{erg} \cdot \text{s}^{-1})$), and Relative flux ($F(\text{line})/F(H_\beta)$).

llum.dat is an alternative representation of line luminosities. The file contains only two rows: header and luminosities for each line specified in the line emissivities input file (see 3.5). Luminosities are printed as ($\log(\text{erg} \cdot \text{s}^{-1})$).

lifr.dat is an alternative representation of line fluxes. The file contains only two rows: header and fluxes for each line specified in the line emissivities input file (see 3.5). Fluxes are printed as ($\text{erg} \cdot \text{cm}^{-2} \cdot \text{s}^{-1}$).

5.2 Continuum Output

contarr.dat contains continuum fluxes at different photon energies. It contains three columns: 1 - Photon energy, 2 - amount of quanta of attenuated incident continuum ($\text{cm}^{-2} \cdot \text{s}^{-1}$), 3 - amount of quanta of diffuse continuum ($\text{cm}^{-2} \cdot \text{s}^{-1}$).

More output files with additional formats are planned for the future.

5.3 Chemical Abundances

abmass.txt contains the abundances of elements within a volume cut by an aperture within the object, weighted by mass. It consists of two rows: heading and list of weighted abundances ($12 - \log(n(\text{element})/n(H))$).

abemso.txt contains the abundances of elements within a volume cut by an aperture within the object, weighted by oxygen lines emissivity ($e(O35007\text{\AA}) + e(O34363\text{\AA}) + e(O34959\text{\AA}) + e(O23726\text{\AA}) + e(O23729\text{\AA})$). It consists of two rows: heading and list of weighted abundances ($12 - \log(n(\text{element})/n(H))$).

5.4 Isophotes

isophote_{index}.dat is a file containing 2D data for drawing flux maps for corresponding lines (as specified in commands.ini file). The 1st and 2nd columns represent spatial angles, and the 3rd column lists the fluxes calculated from the direction determined by these spatial angles.

When 'isophote' commands are present, gnuplot scripts for visualization are generated and located in the "plots" folder within your output directory.

References

- [1] Golovaty V. V., Dmiterko V. I., Mal'kov Yu. F., & Rokach O. V. 1993, Astronomical Report, Vol. 37, Issue 4, pp. 346-354
- [2] O. S. Buhajenko, B. Ya. Melekh 2016, 'METHOD FOR DETAILED CALCULATION OF THE DIFFUSE IONIZING RADIATION IN NEBULAR ENVIRONMENTS', Ivan Franko National University of Lviv, 4901 (13 p.)
- [3] Ferland G.J. 1999, 'Hazy, a Brief Introduction to Cloudy 94', University of Kentucky, Physics Department Internal Report; <http://www.nublado.org>