3D-RT: 3D Radiative Transfer

Background

Frederik De Ceuster

Abstract

This report gives an overview of some technical aspects of the 3D-RT code. The goal is to motivate why and explain how some things are coded the way they are.

Contents

1 General

3D-RT is a multi purpose 3D radiative transfer code
The code is mainly written in C with hsome features of C++.

1.1 Storing multi-dimensional arrays as lists

All multi-dimensional arrays in the code are stored as one-dimensional lists. On the lowest level, this is the case in every computer code. However, we chose to explicitly write the one-dimensional lists and define the relations between list index and the rows and columns.

2 Ray tracing

To be able to simulate 3-dimensional clouds of gas on a computer we must find a way to represent this system in computer language. The first step is to chop up space into a bunch of cells. The (continuous) physical variables like e.g. temperature and density are assumed to be constant within each cell.

Space is discretized into `ngrid` grid points.

2.1 Efficiently storing the evaluation points

THE ORIGIN IS NOT AN EVALUATION POINT!
`raytot[RINDEX(n,r)]` gives the total number of evaluation points on a ray `r` through a gridpoint `n`. Here the origin is not counted as an evaluation point. Otherwise we would store the origin each time as an evaluation point, resulting in `ngrid` times `NRAYS` unnecessary doubles. In the radiative transfer part of the code we do want to consider the origin as an eveluation point. Therefore we will systematically add one to `raytot` in that part of the code.

2.2 Equivalent rays

of

3 Radiative transfer

4 Thermal balance

4.1 Heating

4.2 Cooling

4.2.1 Radiative cooling