Sedona

A general purpose astrophysical radiation transport code

User's Guide

August 16, 2018

CHAPTER 1

Getting Started

1.1 Building the Code

To compile Sedona you will need a C++ compiler with MPI, the gsl library, hdf5, and the lua scripting language. (Many systems will already have all but lua already installed, and can be simply loaded using module load.)

1. Set the environment variable `SEDONA_HOME` to point to the base directory of sedona. In bash, for example, you can add to your .bash profile the line:

   ```
   export SEDONA_HOME=/Users/kasen/sedona6/
   ```

2. Install lua. Source code can be found in the `src/external/` directory

3. Download and install the Gnu Scientific Library (gsl) available at https://www.gnu.org/software/gsl/

4. Download and install hdf5 from https://support.hdfgroup.org/downloads/index.html

5. Go to the `src` directory and edit the make.inc file so that `CXX` is your C++ compiler with options, and the variables point to the location of the installed libraries. Some make.inc files are already included for common systems (e.g., `make.inc.nersc`) and can simply be copied over to `make.inc`.

6. Type `make`.

7. If compilation is successful, the executable file `sedona6` will appear in the src/ directory. Copy this to the directory where you would like to run the code.

Python is currently used for plotting and testing scripts, but is not needed to build and run Sedona itself (NB: Python may eventually replace lua for parameter files). Python 2.7 is being used. On linux to get the necessary packages try:

```
sudo apt-get install python-numpy python-scipy python-matplotlib python h5py
```

1.1.1 Running the Code

To get started, you can try running a simple transport calculation. Copy the `sedona6` executable in the `examples/simple_examples/spherical_lightbulb/1D` directory.

To run the code type

```
./sedona6.ex param.lua
```

where `param.lua` is the file name of the run time parameter file (if no file name is given, the name `param.lua` is assumed). The parameter file must point to three files

1. A model file (an ascii .mod file for 1D runs or a hdf5 file for multi-D) giving the physical conditions (e.g., density, composition) of the setup. The location is specified
2. An atomic data file in hdf5 format. Such files are available in the `data/` directory, with data compiled from various sources.
3. A defaults file, giving the default settings for all runtime parameters. The standard is `defaults/sedona_defaults.lau`, although users can point to their own modified defaults file.

The code generates several files. The `plt_.?????.h5` files contain data in hdf5 format describing the grid properties (e.g., density, temperature, radiation field). If hdf5 tools are installed, type

```
h5ls plt_00001.h5
```

to see the file contents. For 1D models, a `plt_?????.dat` gives some of this data in ascii format.

The code also generates a `spectrum_?????.h5` file giving the output spectrum.

You will find in the `tools/` directory a python library file `sedona.py` that provides functions to read and analyze the data in the plot and spectrum files (NB: this needs to be developed)

CHAPTER 2

Runtime Parameters

Defaults for all parameters are given in a `defaults/sedona_defaults.lua` file

2.1 Time-stepping Parameters

tstep_max_steps	integer number of time steps to take before exiting before stopping
tstep_time_start	start time (in seconds)
tstep_time_stop	stop time (in seconds)
tstep_max_dt	maximum size of a timestep (in seconds)
tstep_min_dt	minimum size of a timestep (in seconds)
tstep_max_delta	maximum fractional size of a timestep (multiply this by the current time to get the limit on the timestep).

2.2 Transport Parameters

transport_nu_grid	frequency grid to calculate opacities/emissivites. In the format of nu_start, nu_stop, nu_delta
transport_radiative_equilibrium	= 0 or 1. If 1, determine gas temperature after each time step from radiative equilibrium, i.e., heating equals radiative cooling
transport_steady_iterate	= integer. Do not step in time, rather iterate the radiation transport (in steady state) the number of times given.

CHAPTER 3

Test Problems

There is a test suit. We should describe each test problem.

3.0.1 Spherical Lightbulb

Setup: A spherical inner boundary emits blackbody radiation into an extremely low density medium, with optical depth so low it is essentially vacuum.

Test #1 - *Emergent Spectrum*: This should be a blackbody at the input inner core temperature. Tests general sampling of

Test #2 - *Radiation Temperature Structure*:. The radiation field outside a spherical emitter should be given by the dilution factor

$$J = \frac{1}{2}\left[1 - \sqrt{1 - R_0^2/r^2}\right] \tag{3.1}$$