Haverford College

Department of Physics and Astronomy

POWDERDAY **Manual**

Desika Narayanan
dnarayan@haverford.edu

Last Compiled: March 2, 2015

Table of Contents

I Installation 3

I.1 Overview . . . 3

I.1.1 YT installation . . . 3

I.1.2 Python Dependencies . . . 4

I.1.3 HYPERION Installation . . . 4

I.1.4 HYPERION Dust Files . . . 4

I.1.5 FSPS . . . 5

I.1.6 PYTHON-FSPS . . . 5

I.1.7 TroubleShooting . . . 5

I.1.7.1 FSPS and FSPS . . . 5

I.1.8 HYPERION . . . 6

I.1.9 POWDERDAY . . . 6

II Quick Start 7

II.1 Gadget . . . 7

III Description of Parameters 8

IV Algorithm Description 9

V Known Issues 10

Chapter I

Installation

I.1 Overview

We are working on an all-in-one installer script for POWDERDAY that will work on a variety of platforms. For the time being, for better or worse, the installation is manual. The fundamental packages you will need to have installed are:

1. YT
2. Python Dependencies (listed below)
3. HYPERION
4. FSPS
5. PYTHON-FSPS

Here, we'll detail the rough order of operations, though it's always wise to check the parent website (referenced in each subsection) for details.

I.1.1 YT installation

The parent website is http://yt-project.org. There, you will find at http://yt-project.org/#getyt a variety of ways to install YT. We recommend the Development branch, though POWDERDAY is currently known to work with the Stable branch. Note - the Legacy branch of YT is incompatible with POWDERDAY. You can either install from the install scripts, or use conda or pip. Note that to use YT, you will either need to activate YT via:

```
 > source YT\_DEST/bin/activate
```

or, in your .bashrc:

```
export PATH = \$HOME/yt-x86_64/bin:\$PATH
```

By doing this, the python you use (and install packages to, subsequently) will be the YT python. If you don't want this to happen, just ensure that the yt packages are in your python path, and install the forthcoming packages to whichever python distribution you would like.

I.1.2 Python Dependencies

POWDERDAY depends on both SCIPY, ASTROPY, and ATPY. The easiest way to install these is with pip, following:

```
> pip install scipy
> pip install astropy
> pip install atpy
```

You can also install them straight from source, which you can get from http://scipy.org, and http://astropy.org.

I.1.3 HYPERION Installation

The HYPERION installation is described in full on the HYPERION website, and therefore we direct you there: http://docs.hyperion-rt.org/en/stable/installation/installation.html for this. Note, Tom Robitaille's general host website for HYPERION is at: http://www.hyperion-rt.org. A few notes:

- First install the Fortran code dependencies: http://docs.hyperion-rt.org/en/stable/installation/fortran_dependencies.html
- Second, install any remaining python dependencies: http://docs.hyperion-rt.org/en/stable/installation/python_dependencies.html. Note, if you're using the python that comes with YT, then you should already have all of these installed.
- Finally, install HYPERION itself: http://docs.hyperion-rt.org/en/stable/installation/installation.html#hyperion
- Make sure that all of the tests that the HYPERION installation page suggests actually work (i.e. when you type:

  ```
  >hyperion\_sph
  ```

  you get something like:

  ```
  Usage: hyperion input\_file output\_file
  ```

I.1.4 HYPERION Dust Files

Unless you've written your own dust files (a tutorial for which will come later in this manual), you might want to use the pre-compiled dust files that are developed by Tom Robitaille (though don't ship with HYPERION). To install these, download and install from http://docs.hyperion-rt.org/en/stable/dust/dust.html.

In order to run with the PAH model turned on, you'll additionally need the MW PAH dust files. There are three (big, very small grains and ultra small grains). These can be donwloaded here:
https://github.com/hyperion-rt/paper-galaxy-rt-model/blob/master/dust/big.hdf5
https://github.com/hyperion-rt/paper-galaxy-rt-model/blob/master/dust/vsg.hdf5
https://github.com/hyperion-rt/paper-galaxy-rt-model/blob/master/dust/usg.hdf5

I.1.5 FSPS

FSPS is hosted at google code, and can be checked out with:

```
> svn checkout http://fsps.googlecode.com/svn/trunk/ fsps
```

and directions to the installation are in the Manual: https://www.cfa.harvard.edu/~cconroy/FSPS_files/MANUAL.pdf. Note, Charlie Conroy's host website for FSPS is at: https://www.cfa.harvard.edu/~cconroy/FSPS.html.

While POWDERDAY (more accurately, it's dependencies on PYTHON-FSPS) may work with the most current version of FSPS, they are only known to work for FSPS revisison 143. Therefore, it is safest to revert to this revision with:

```
>svn update -r 143
```

Note that the SPS_HOME environment variable must be set in your path to point to the FSPS/src directory. For example, in my .bashrc, I have:

```
>export SPS\_HOME=/Users/dnarayanan/fsps/
```

I.1.6 PYTHON-FSPS

POWDERDAY depends on python hooks for FSPS written by Dan Foreman-Mackey and others, hosted at: http://dan.iel.fm/python-fsps/current/installation/. For PYTHON-FSPS, there is a pip installer that will allow you to install via:

```
>pip install fsps
```

Though you cuold also install the development version with:

```
git clone https://github.com/dfm/python-fsps.git
cd python-fsps
python setup.py install
```

I.1.7 TroubleShooting

So far, the aforementioned has been known to work for Mac OS X, Ubuntu and CentOS linux operating systems. Of course, troubles are potential. Here are a few issues that we've come across with possible solutions.

I.1.7.1 FSPS and FSPS

Problem: PYTHON-FSPS has an error related to 'fPIC' when installing.
Solution: try include that flag in the FSPS makefile. You can do this by adding -fPIC to the F90FLAGS in the Makefile for FSPS, i.e.

```
>F90FLAGS = -O -cpp -fPIC
```

in your Makefile.

Problem PYTHON-FSPS gives errors related to f2py when installing
Solution To install python-fsps, you need to have f2py installed. If you're using the YT python, this comes with the YT installation, but is named f2py2.7 and is in (e.g.)

```
\Users/desika/yt-x86_64/bin
```

So, simply go to that directory and:

```
> ln -s f2py2.7 f2py
```

You additionally need to go to:

```
>cd  yt-x86_64/lib/python2.7/site-package
> ln -s numpy/f2py/ f2py
```

I.1.8 HYPERION

Problem: HYPERION gives an error like:

```
ld: library not found for -lcrt1.10.5.o
```

when compiling (particularly with Mac OS X)
Solution: It could be that there are issues with the command line tools installation (which has cropped up especially for the Mavericks OS). Try:

```
>sudo xcode-select --install
```

(assuming you have xcode installed already). This will downlad and install the command line tools.

I.1.9 POWDERDAY

Problem: Memory Errors at the beginning of the Peeled Images setup
Solution: Change the track_origin from detailed to basic in pd_front_end.py

Chapter II

Quick Start

In the examples subdirectory, you will find example snapshots for the different hydro codes supported with Powderday. This will likely change significantly over time as the code gets cleaned up, the parameter files change, and the example snapshots migrate toward the Agora snaps. But, in the mean time, this should hopefully get you started.

For each example file there should be two parameter files that will be reasonable for the associated snapshot.

To run, type (in your source directory):

```
> python pd_front_end.py <example directory> <parameters_master_file> <parameters_model_file>
```

Note - the .py extensions on the parameter files need to be left off. So, as an example for the GADGET snapshot detailed below:

```
> python pd_front_end.py examples/gadget parameters_master parameters_model
```

II.1 Gadget

The GADGET file here is an idealized disk galaxy (actually created with GIZMO but the outputs are the same). The snapshot can be downloaded here: `https://www.dropbox.com/s/055bewn1iv7mk7r/snapshot_020.hdf5?dl=0`.

The SED as of hash 191d121 (in Jy, redshift 0) is [to come..]:

Chapter III

Description of Parameters

Chapter IV

Algorithm Description

Chapter V

Known Issues