Variables

`CHIANTI_PERTURB_SPL1` : The magnitude of the pertubation applied to the collision strengths for transitions that connect to the levels in the ground configuration. The perturbations are normally distributed with a mean of zero and a standard deviation given by CHIANTI_PERTURB_SPL1. The other perturbations are similar.

`CHIANTI_PERTURB_SPL2`: The magnitude of the pertubation applied to the collision strengths for transitions that connect to the levels other than those in the ground configuration.

`CHIANTI_PERTURB_AVAL`: The magnitude of the pertubation applied to the A values (decay rates).

`ioneq_file`: The name of the CHIANTI ionization equilibrium file.

`logt`: The log temperature array.

`logt`: The log density array.

`logt_max`: The peak in the ionization fraction for this ion.

`nsim`: The number of realizations of the atomic data.

`wavelength`: The wavelength for each line of interest. Note that all transitions within 0.1 Angstroms of this wavelength will be included in the emissivity. Only a single wavelength for each line is given here.

`transition`: A list of all of the transitions that are included in the emissivity. The transitions are listed using the level numbers. For example, for O VIII 18.969 there are three transitions close in wavelength:

```
 ion = o_8
 this wave = 18.969
 nearby transiions = 3
  index      wave     dwave       emiss
     15   18.9671    0.0019    3.48e-09
     16   18.9723    0.0033    1.39e-12
     17   18.9726    0.0036    2.18e-09
 transitions = 1-4 / 1-2 / 1-3
```

`emissivity`: The emissivity as a function of logn at logt_max for each of the transitions of interest. The organization of the array will depend on the language being used to read the file (column-major vs row-major).

`emissivity_t`: The emissivity as a function of logt and logn for each of the transitions of interest. This is used for investigating the temperature sensitivity of the density ratios.

`time_stamp`: The IDL system time when the routine was run.

Reading in R

To read HDF5 files in R you need to download and compile `rhdf5`. This is done within R using

```
source("http://bioconductor.org/biocLite.R")
biocLite("rhdf5")
```

Once that is done you read the variables using statements such as

```
library("rhdf5")
fname <- "fe_13.monte_carlo.h5"
emissivity <- h5read(fname, 'emissivity')
logn <- h5read(fname, 'logn')
wavelength <- h5read(fname, 'wavelength')
```

Reading in Python

Use the `h5py` package, which is included in the Anaconda distribution of python. Converting to numpy arrays is also useful.

```
import h5py
import numpy as np
file = 'o_8.monte_carlo.h5'
mc = h5py.File(file,'r')
emissivity = np.array(mc['emissivity'])
logn = np.array(mc['logn'])
wavelength = np.array(mc['wavelength'])
```

Reading in IDL

Here is some pseudocode for reading a file in IDL:

```
file = 'o_8.monte_carlo.h5'
file_id = h5f_open(file)
dset_id = h5d_open(file_id, 'emissivity')
emissivity = h5d_read(dset_id)
h5d_close, dset_id
```