Using_pyCloudy_1

June 22, 2016

```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt
```

```
In [2]: import pyCloudy as pc
```

```
In [3]: # Define verbosity to high level (will print errors, warnings and messages)
        pc.log_.level = 3
```

```
In [4]: # The directory in which we will have the model
        # You may want to change this to a different place so that the current dire
        # will not receive all the Cloudy files.
        dir_ = './'
```

```
In [5]: # Define some parameters of the model:
        model_name = 'model_1'
        full_model_name = '{0}{1}'.format(dir_, model_name)
        dens = 2. #log cm-3
        Teff = 45000. #K
        qH = 47. #s-1
        r_min = 5e17 #cm
        dist = 1.26 #kpc
```

```
In [6]: # these are the commands common to all the models (here only one ...)
        options = ('no molecules',
                    'no level2 lines',
                    'no fine opacities',
                    'atom h-like levels small',
                    'atom he-like levels small',
                    'COSMIC RAY BACKGROUND',
                    'element limit off -8',
                    'print line optical depth',
                    )
```

```
In [7]: emis_tab = ['H  1  4861',
                    'H  1  6563',
                    'He 1  5876',
                    'N  2  6584',
                    'O  1  6300',
                    'O II  3726',
                    'O II  3729',
                    'O  3  5007',
                    'TOTL  4363',
                    'S II  6716',
                    'S II 6731',
                    'Cl 3 5518',
                    'Cl 3 5538',
                    'O  1 63.17m',
                    'O  1 145.5m',
                    'C  2 157.6m']
```

```
In [8]: abund = {'He' : -0.92, 'C' : 6.85 - 12, 'N' : -4.0, 'O' : -3.40, 'Ne' : -4.
                 'S' : -5.35, 'Ar' : -5.80, 'Fe' : -7.4, 'Cl' : -7.00}
```

```
In [9]: # Defining the object that will manage the input file for Cloudy
        c_input = pc.CloudyInput(full_model_name)
```

```
In [10]: # Filling the object with the parameters
         # Defining the ionizing SED: Effective temperature and luminosity.
         # The lumi_unit is one of the Cloudy options, like "luminosity solar", "q
         c_input.set_BB(Teff = Teff, lumi_unit = 'q(H)', lumi_value = qH)
```

```
In [11]: # Defining the density. You may also use set_dlaw(parameters) if you have
         c_input.set_cste_density(dens)
```

```
In [12]: # Defining the inner radius. A second parameter would be the outer radius
         c_input.set_radius(r_in=np.log10(r_min))
         c_input.set_abund(ab_dict = abund, nograins = True)
         c_input.set_other(options)
         c_input.set_iterate() # (0) for no iteration, () for one iteration, (N) fo
         c_input.set_sphere() # () or (True) : sphere, or (False): open geometry.
         c_input.set_emis_tab(emis_tab) # better use read_emis_file(file) for long
         c_input.set_distance(dist=dist, unit='kpc', linear=True) # unit can be 'kp
```

```
In [13]: # Writing the Cloudy inputs. to_file for writing to a file (named by full_
         c_input.print_input(to_file = True, verbose = False)

     CloudyInput: Input writen in ./model_1.in
```

```
In [14]: # Printing some message to the screen
         pc.log_.message('Running {0}'.format(model_name), calling = 'test1')

     test1: Running model_1
```

```
In [15]: # Running Cloudy with a timer. Here we reset it to 0.
         pc.log_.timer('Starting Cloudy', quiet = True, calling = 'test1')
         c_input.run_cloudy()
         pc.log_.timer('Cloudy ended after seconds:', calling = 'test1')
```

```
  run_cloudy: running: cd . ;  cloudy.exe
  run_cloudy: ending: cd . ;  cloudy.exe
test1: Cloudy ended after seconds: in 26.9458298683
```

In [16]:
```
# Reading the Cloudy outputs in the Mod CloudyModel object
Mod = pc.CloudyModel(full_model_name)
```

```
CloudyModel ./model_1: Creating CloudyModel for ./model_1
CloudyModel ./model_1: Li abundance not defined
CloudyModel ./model_1: Be abundance not defined
CloudyModel ./model_1: B abundance not defined
CloudyModel ./model_1: Sc abundance not defined
CloudyModel ./model_1: ./model_1.rad read
CloudyModel ./model_1: Number of zones: 118
CloudyModel ./model_1: ./model_1.phy read
CloudyModel ./model_1: ./model_1.ele_H read
CloudyModel ./model_1: filling H with 3 columns
CloudyModel ./model_1: ./model_1.ele_He read
CloudyModel ./model_1: filling He with 3 columns
CloudyModel ./model_1: ./model_1.ele_C read
CloudyModel ./model_1: filling C with 13 columns
CloudyModel ./model_1: ./model_1.ele_N read
CloudyModel ./model_1: filling N with 8 columns
CloudyModel ./model_1: ./model_1.ele_O read
CloudyModel ./model_1: filling O with 12 columns
CloudyModel ./model_1: ./model_1.ele_Ne read
CloudyModel ./model_1: filling Ne with 11 columns
CloudyModel ./model_1: ./model_1.ele_Ar read
CloudyModel ./model_1: filling Ar with 19 columns
CloudyModel ./model_1: ./model_1.ele_S read
CloudyModel ./model_1: filling S with 17 columns
CloudyModel ./model_1: ./model_1.ele_Cl read
CloudyModel ./model_1: filling Cl with 18 columns
CloudyModel ./model_1: ./model_1.ele_Fe read
CloudyModel ./model_1: filling Fe with 27 columns
CloudyModel ./model_1: ./model_1.ele_Si read
CloudyModel ./model_1: filling Si with 15 columns
CloudyModel ./model_1: ./model_1.emis read
CloudyModel ./model_1: Number of emissivities: 16
CloudyModel ./model_1: ./model_1.cont read
```

In [17]:
```
# Use TAB to know all the methods and variables for CloudyModel class
# Mod.TAB
dir(Mod) # This is the online answering way
# Description of this class is available here: http://pythonhosted.org//py
```

Out[17]:
```
['C3D_comments',
 'H0_mass',
 'H_mass',
 'H_mass_cut',
 'H_mass_full',
 'Hbeta',
 'Hbeta_cut',
 'Hbeta_full',
 'Hp_mass',
 'Phi',
 'Phi0',
 'Q',
 'Q0',
 'T0',
 'Teff',
 '_CloudyModel__H_mass_cut',
 '_CloudyModel__Hbeta_cut',
 '_CloudyModel__r_in_cut',
 '_CloudyModel__r_out_cut',
 '_CloudyModel__r_range',
 '__class__',
 '__delattr__',
 '__dict__',
 '__doc__',
 '__format__',
 '__getattribute__',
 '__hash__',
 '__init__',
 '__module__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_get_H_mass_cut',
 '_get_Hbeta_cut',
 '_get_r_in_cut',
 '_get_r_out_cut',
 '_i_emis',
 '_i_line',
 '_init_all2zero',
 '_init_cont',
 '_init_emis',
 '_init_grains',
 '_init_heatcool',
 '_init_ionic',
 '_init_lin',
 '_init_opd',
 '_init_phy',
 '_init_rad',
 '_l_emis',
 '_quiet_div',
 '_r_out_cut_doc',
 '_read_stout',
 '_res',
 '_set_H_mass_cut',
 '_set_Hbeta_cut',
 '_set_r_in_cut',
 '_set_r_out_cut',
 'aborted',
 'abund',
 'calling',
 'cautions',
 'cloudy_version',
 'cloudy_version_major',
 'comments',
 'cool',
 'cool_full',
 'date_model',
 'depth',
 'depth_full',
 'distance',
 'dr',
 'dr_full',
 'drff',
 'dv',
 'dv_full',
 'dvff',
 'emis_from_pyneb',
 'emis_full',
 'emis_is_log',
 'emis_labels',
 'empty_model',
 'ff',
 'ff_full',
 'gabund',
 'gabund_full',
 'gabund_labels',
 'gas_mass_per_H',
 'gasize',
 'gdgrat',
 'gdgrat_full',
 'gdgrat_labels',
 'gdsize',
 'get_G0',
 'get_Ha_EW',
 'get_Hb_EW',
 'get_Hb_SB',
 'get_T0_emis',
 'get_T0_emis_rad',
 'get_T0_ion_rad',
 'get_T0_ion_rad_ne',
 'get_T0_ion_vol',
 'get_T0_ion_vol_ne',
 'get_ab_ion_rad',
 'get_ab_ion_rad_ne',
 'get_ab_ion_vol',
 'get_ab_ion_vol_ne',
 'get_cont_x',
 'get_cont_y',
 'get_emis',
 'get_emis_rad',
 'get_emis_vol',
 'get_ionic',
 'get_line',
 'get_ne_emis',
 'get_ne_ion_rad_ne',
 'get_ne_ion_vol_ne',
 'get_t2_emis',
 'get_t2_ion_rad_ne',
 'get_t2_ion_vol_ne',
 'gsize',
 'gtemp',
 'gtemp_full',
 'gtemp_labels',
 'heat',
 'heat_full',
 'info',
 'intens',
 'ionic_full',
 'ionic_names',
 'is_valid_ion',
 'line_is_log',
 'lines',
 'liste_elem',
 'log_',
 'log_U',
 'log_U_mean',
 'log_U_mean_ne',
 'model_name',
 'model_name_s',
 'nH',
 'nH_full',
 'nH_mean',
 'nHff_full',
 'n_elements',
 'n_emis',
 'n_gabund',
 'n_gdgrat',
 'n_gtemp',
 'n_ions',
 'n_lines',
 'n_zones',
 'n_zones_full',
 'ne',
 'ne_full',
 'nenH',
 'nenH_full',
 'nenHff2_full',
 'opd_absorp',
 'opd_energy',
 'opd_scat',
 'opd_total',
 'out',
 'out_exists',
 'phi',
 'plan_par',
 'print_lines',
 'print_stats',
 'r_in',
 'r_in_cut',
 'r_out',
 'r_out_cut',
 'r_range',
 'rad_integ',
 'rad_mean',
 'radius',
 'radius_full',
 'read_outputs',
 'rlines',
 'slines',
 't2',
 'te',
 'te_full',
 'tenenH',
 'tenenH_full',
 'theta',
 'thickness',
 'thickness_full',
 'vol_integ',
 'vol_mean',
 'warnings',
 'zones',
 'zones_full']
```

```
In [18]: Mod.print_stats()
```

```
 Name of the model: ./model_1
 R_in (cut) = 5.000e+17 (5.000e+17), R_out (cut) = 1.952e+18 (1.952e+18)
 H+ mass = 2.41e+00, H mass = 2.56e+00
 <H+/H> = 0.97, <He++/He> = 0.00, <He+/He> = 0.84
 <O+++/O> = 0.00, <O++/O> = 0.28, <O+/O> = 0.68
 <N+++/O> = 0.00, <N++/O> = 0.39, <N+/O> = 0.59
 T(O+++) = 7640, T(O++) = 7505, T(O+) = 7903
 <ne> = 104, T0 = 7790, t2=0.0026
 <log U> = -2.80
```

```
In [19]: Mod.print_lines()
```

```
H__1__4861A 4.678883e+34
H__1__6563A 1.386719e+35
HE_1__5876A 8.099992e+33
N__2__6584A 7.919793e+34
O__1__6300A 1.918884e+33
O_II__3726A 5.010339e+34
O_II__3729A 6.737289e+34
O__3__5007A 5.459634e+34
TOTL__4363A 1.231720e+32
S_II__6716A 8.090348e+33
S_II__6731A 6.296074e+33
CL_3__5518A 1.130610e+32
CL_3__5538A 8.092211e+31
O__1_6317M 9.619285e+32
O__1_1455M 9.462694e+31
C__2_1576M 1.799563e+32
```

```
In [20]: Mod.get_ab_ion_vol_ne('O',2)

Out[20]: 0.2848508383202098

In [21]: Mod.get_T0_ion_vol_ne('O', 2)

Out[21]: 7504.9928224483492

In [22]: Mod.log_U_mean

Out[22]: -2.8012415090484888
```

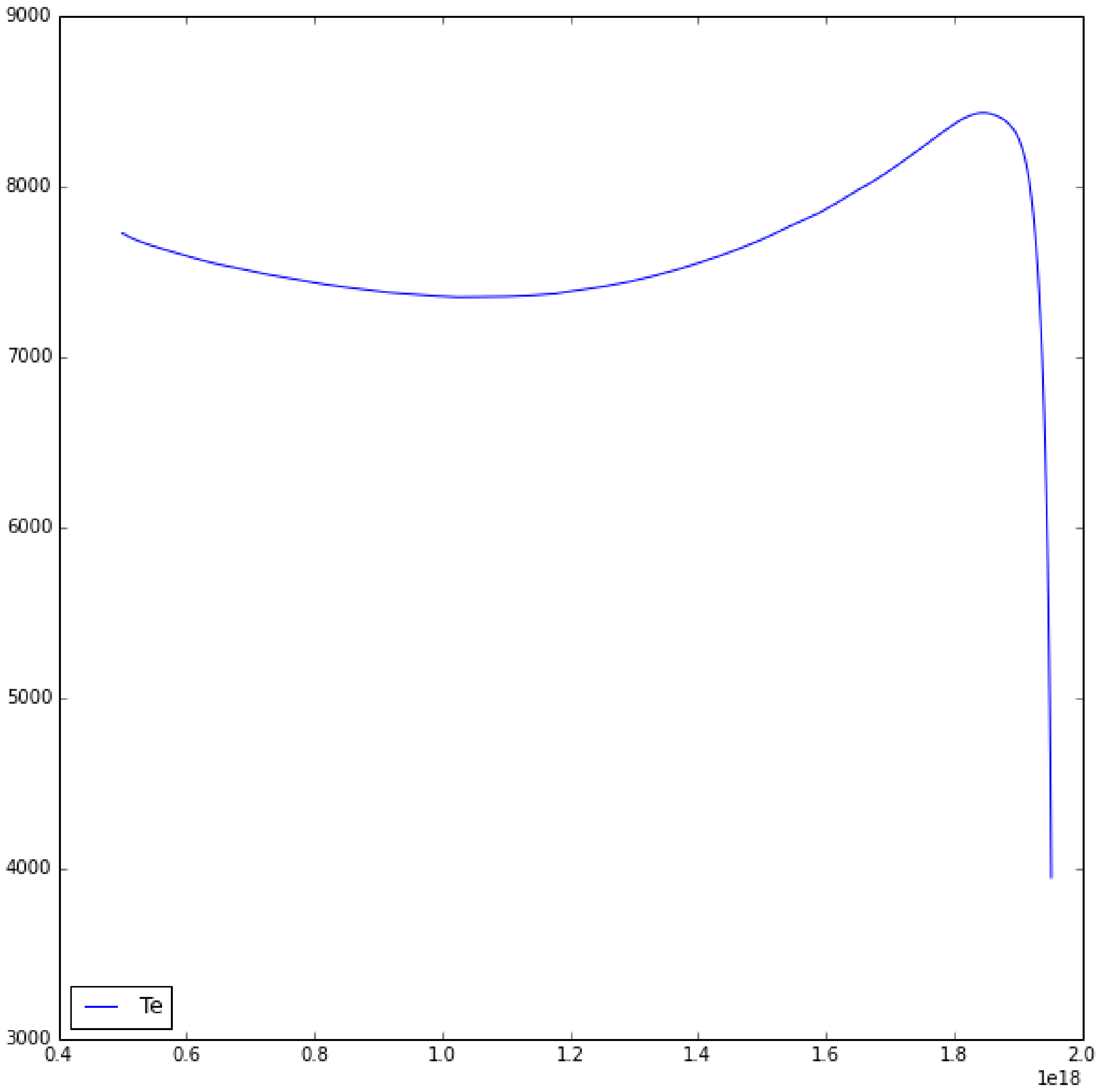
```
In [23]: Mod.log_U_mean_ne

Out[23]: -2.7838194930777269
```

```
In [24]: print('T0 = {0:7.1f}K, t2 = {1:6.4f}'.format(Mod.T0, Mod.t2))

T0 =  7789.5K, t2 = 0.0026
```

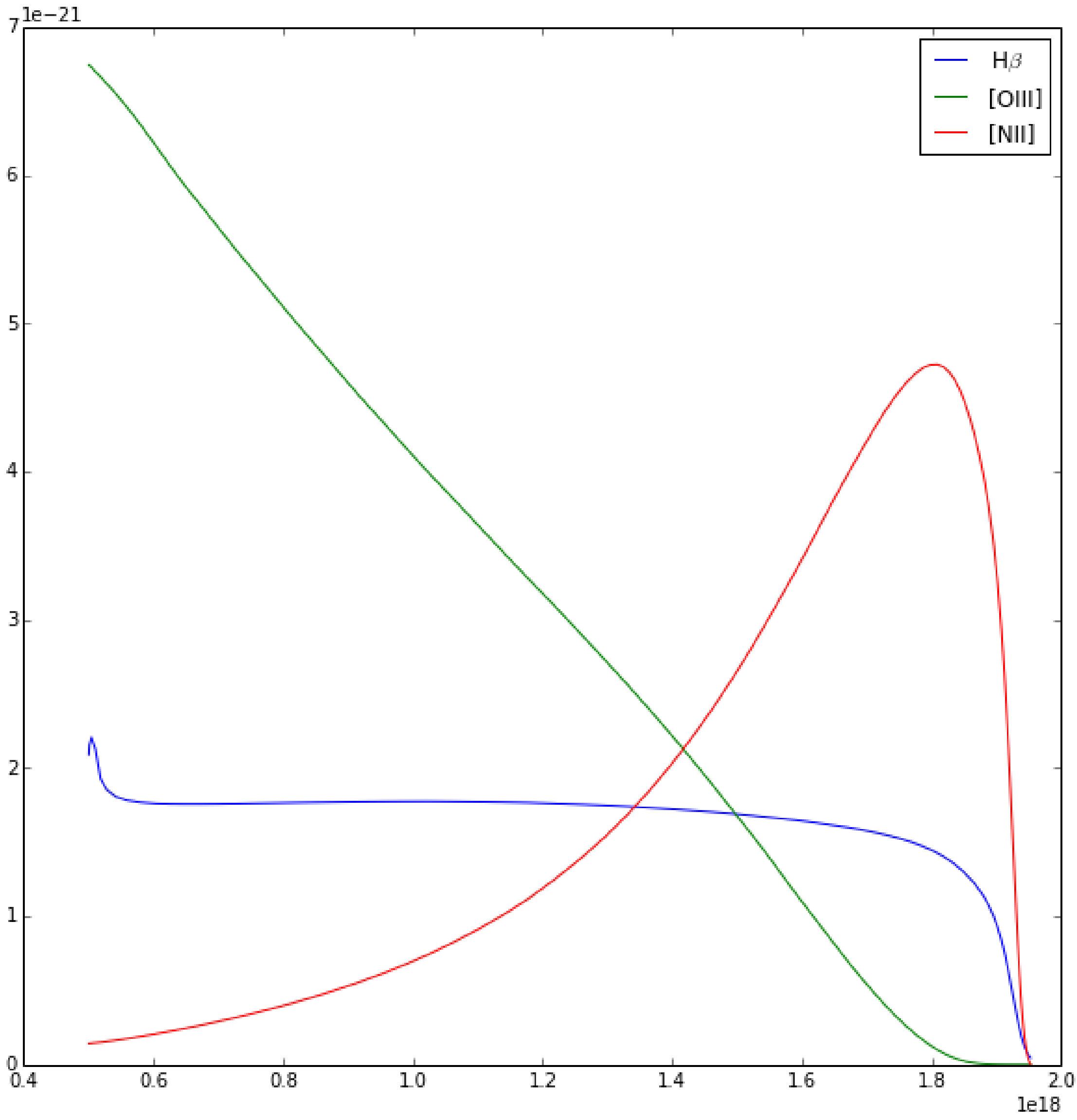
```
In [25]: print('Hbeta Equivalent width = {0:6.1f}, Hbeta Surface Brightness = {1:4

Hbeta Equivalent width = -720.8, Hbeta Surface Brightness = 9.18e-14
```

```
In [26]: # printing line intensities
         for line in Mod.emis_labels:
             print('{0} {1:10.3e} {2:7.2f}'.format(line, Mod.get_emis_vol(line), Mo

H__1__4861A  4.679e+34  100.00
H__1__6563A  1.387e+35  296.38
HE_1__5876A  8.100e+33   17.31
N__2__6584A  7.920e+34  169.27
O__1__6300A  1.919e+33    4.10
O_II__3726A  5.010e+34  107.08
O_II__3729A  6.737e+34  143.99
O__3__5007A  5.460e+34  116.69
TOTL__4363A  1.232e+32    0.26
S_II__6716A  8.090e+33   17.29
S_II__6731A  6.296e+33   13.46
CL_3__5518A  1.131e+32    0.24
CL_3__5538A  8.092e+31    0.17
O__1_6317M  9.619e+32    2.06
O__1_1455M  9.463e+31    0.20
C__2_1576M  1.800e+32    0.38
```

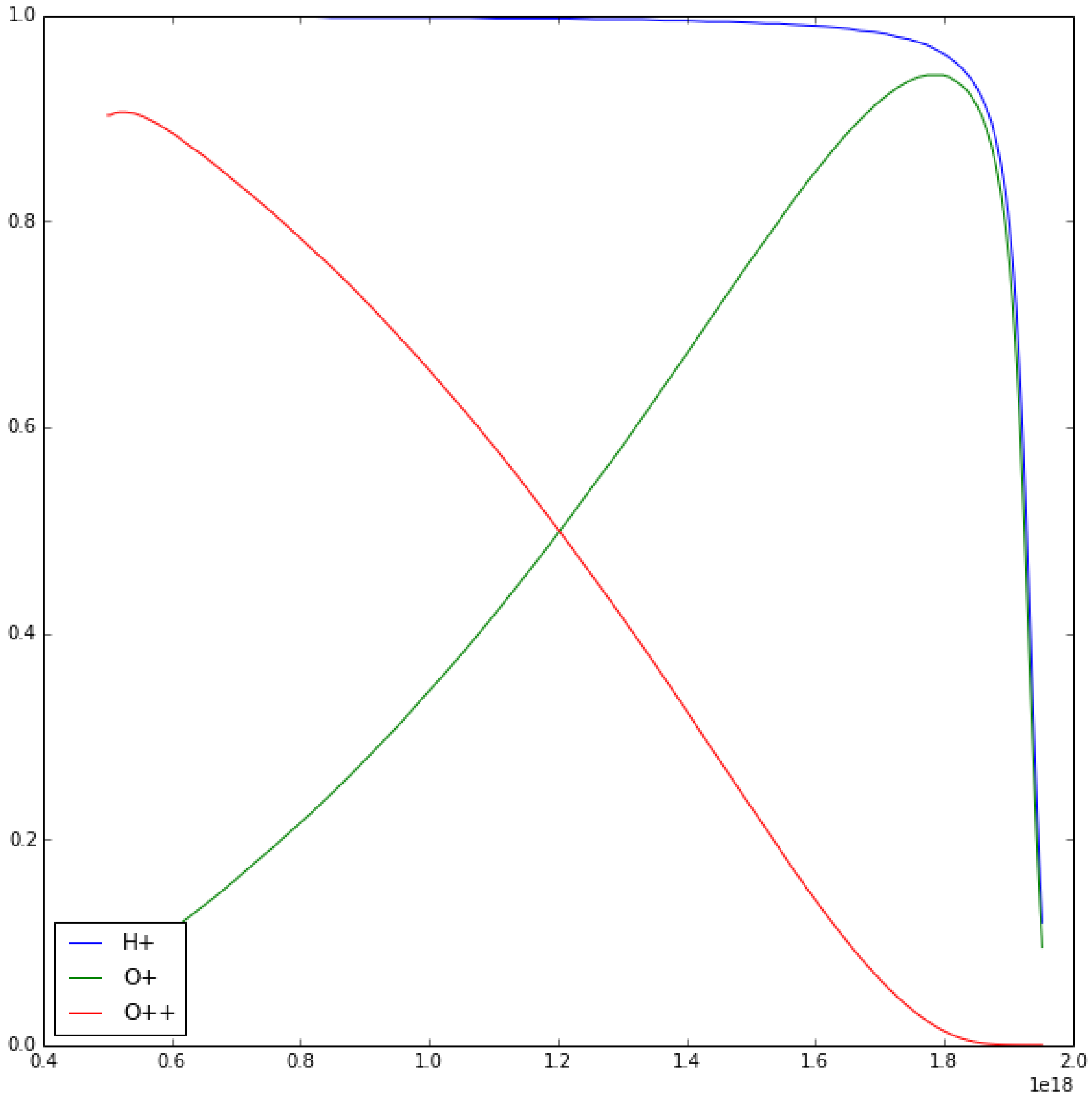
```
In [27]: plt.figure(figsize=(10,10))
         plt.plot(Mod.radius, Mod.te, label = 'Te')
         plt.legend(loc=3)

Out[27]: <matplotlib.legend.Legend at 0x108fb7510>
```



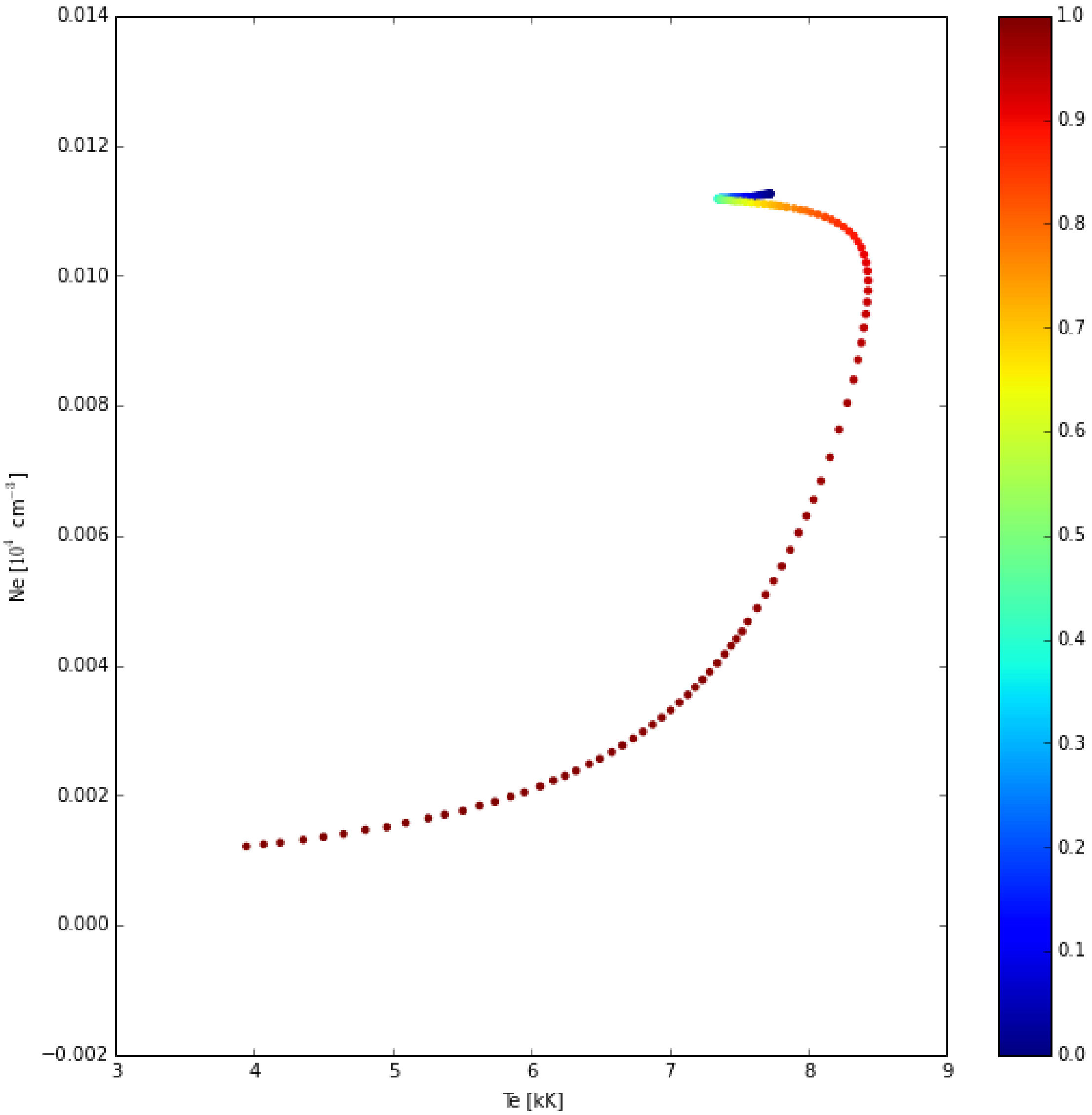
```
In [28]: plt.figure(figsize=(10,10))
         plt.plot(Mod.radius, Mod.get_emis('H__1__4861A'), label = r'H$\beta$')
         plt.plot(Mod.radius, Mod.get_emis('O__3__5007A'), label = '[OIII]')
         plt.plot(Mod.radius, Mod.get_emis('N__2__6584A'), label = '[NII]')
         plt.legend()

Out[28]: <matplotlib.legend.Legend at 0x108fd85d0>
```



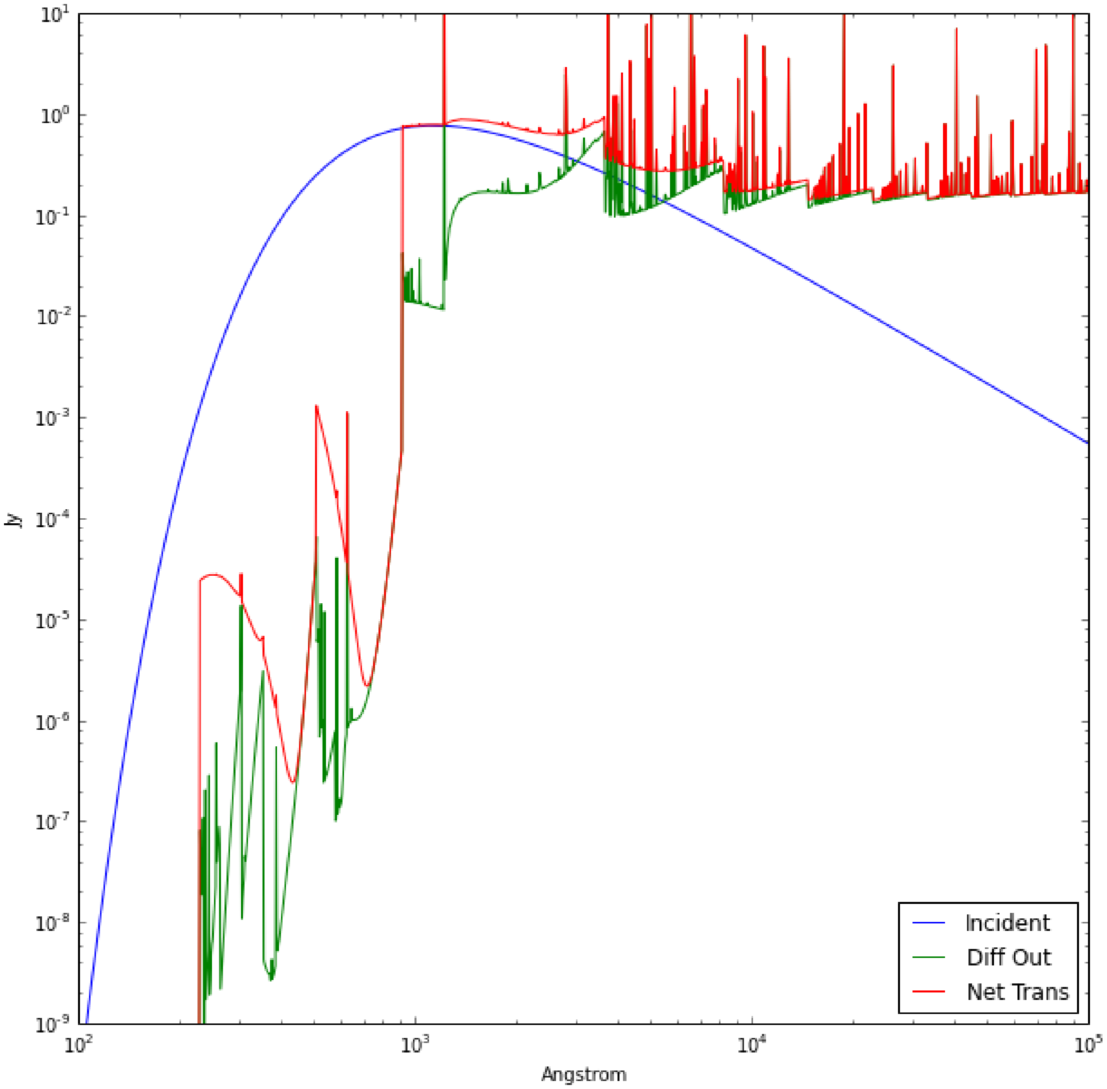
```
In [29]: plt.figure(figsize=(10,10))
         plt.plot(Mod.radius, Mod.get_ionic('H', 1), label = 'H+')
         plt.plot(Mod.radius, Mod.get_ionic('O', 1), label = 'O+')
         plt.plot(Mod.radius, Mod.get_ionic('O', 2), label = 'O++')
         plt.legend(loc=3)

Out[29]: <matplotlib.legend.Legend at 0x1091ac4d0>
```



```
In [30]: plt.figure(figsize=(10,10))
         plt.scatter(Mod.te/1e3, Mod.ne/1e4, c = Mod.depth/np.max(Mod.depth), edgec
         plt.colorbar()
         plt.xlabel('Te [kK]')
         plt.ylabel(r'Ne [$10^4$ cm$^{-3}$]')

Out[30]: <matplotlib.text.Text at 0x109173450>
```



```
In [31]: plt.figure(figsize=(10,10))
         plt.loglog(Mod.get_cont_x(unit='Ang'), Mod.get_cont_y(cont = 'incid', unit
         plt.loglog(Mod.get_cont_x(unit='Ang'), Mod.get_cont_y(cont = 'diffout', un
         plt.loglog(Mod.get_cont_x(unit='Ang'), Mod.get_cont_y(cont = 'ntrans', uni
         plt.xlim((100, 100000))
         plt.ylim((1e-9, 1e1))
         plt.xlabel('Angstrom')
         plt.ylabel('Jy')
         plt.legend(loc=4)

Out[31]: <matplotlib.legend.Legend at 0x1091ba850>
```


10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
Jy
10^2
10^3
10^4
10^5
Angstrom
Incident
Diff Out
Net Trans