Using_pyCloudy_3

August 6, 2025

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
import os
home_dir = os.environ['HOME'] + '/'
import pyCloudy as pc
print(pc.__version__)
```

```
warng pyCloudy config: pyCloudy works better with matplotlib Triangulation
0.9.16
```

[2]:
```python
pc.config.cloudy_exe = '/usr/local/Cloudy/c25.00_rc2/source/cloudy.exe'
```

[3]:
```python
dir_ = '/tmp/models/'
pc.print_make_file(dir_)
```

[4]:
```python
def set_models(dir_, model_name):
    emis_tab = ['H  1  4861.32A',
                'H  1  6562.80A',
                'Ca B  5875.64A',
                'N  2  6583.45A',
                'O  1  6300.30A',
                'O  2  3726.03A',
                'O  2  3728.81A',
                'O  3  5006.84A',
                'O  3  4363.21A',
                'O 3R  4363.00A',
                'O 3C  4363.00A',
                'S  2  6716.44A',
                'S  2  6730.82A',
                'Cl 3  5517.71A',
                'Cl 3  5537.87A',
                'O  1  63.1679m',
                'O  1  145.495m',
                'C  2  157.636m']
    a = 2.
    b = 1.0
    thetas = np.linspace(0., 90., 6)
    thetas_rad = np.pi / 180. * thetas
    fact_elli = a * b / np.sqrt((b * np.sin(thetas_rad))**2 + (a * np.
↪cos(thetas_rad))**2)
    rs_in = 16.5 + np.log10(fact_elli)
    densities = 4 - np.log10(fact_elli) * 2

    model = pc.CloudyInput()
    model.set_BB(80000., 'q(H)', 47.3)
    model.set_grains()
    model.set_emis_tab(emis_tab)

    for theta, r_in, density in zip(thetas, rs_in, densities):
        model.model_name = '{0}/{1}_{2:.0f}'.format(dir_, model_name,theta)
        model.set_cste_density(density)
        model.set_radius(r_in)
        model.set_theta_phi(theta)
        model.print_input(to_file = True, verbose = False)
```

[5]:

```
def def_profiles(m3d):
    """
    This uses the default velocity law (polynome) and default profile (gaussian)
    """
    m3d.set_velocity(params = [20.,60.])
    m3d.config_profile(size_spectrum = 51, vel_max = 50, v_turb = 0.01)
```

[6]:

```
def def_profiles_user(m3d):
    """
    Use this to define your own expansion velocity
    """
    def velo_polynome(params):
        """
        USer defined expansion velocity
        """
        # params is a 2 elements table, the first element is a table of␣
↪parameters, the second one the cob_coord
        # which is needed to know r, x, y and z to define the velocity.
        coeffs = params[0]
        cub_coord = params[1]
        tmp = 0.
        for i, coeff in enumerate(coeffs):
            # for each parameter we add the corresponding coeff * R**power
            tmp = tmp + coeff * cub_coord.r**i
        tmp = tmp / cub_coord.r
        # to avoid the singularity:
        tt = (cub_coord.r == 0.)
        tmp[tt] = 0
        # Projecting on each one of the 3 axes to obtain the velocity components
        vel_x = tmp * cub_coord.x / np.max(cub_coord.x)
        vel_y = tmp * cub_coord.y / np.max(cub_coord.y)
        vel_z = tmp * cub_coord.z / np.max(cub_coord.z)
        return vel_x, vel_y, vel_z

    def Hb_prof(x, zeta_0):
        """
        The Hbeta profile is sum of 2 blocks of lines (actually 3 + 4 lines)
        """
        res1 = .41 /zeta_0 / np.sqrt(np.pi) * np.exp(-(((x-2.7)/zeta_0)**2))
        res2 = .59 /zeta_0 / np.sqrt(np.pi) * np.exp(-(((x+2.0)/zeta_0)**2))
        return res1 + res2

    m3d.set_velocity(velocity_law='user', params = [[20.,60.], m3d.cub_coord],␣
 ↪user_function = velo_polynome)
    m3d.config_profile(size_spectrum = 41, vel_max = 25, profile_function =␣
 ↪Hb_prof, v_turb = 0.01)
```

[7]:
```
def plot_profiles(m3d, x_pos, y_pos):
    plt.plot(m3d.vel_tab,m3d.get_profile('H__1_486132A', axis='x')[:
 ↪,x_pos,y_pos] * 5, label = r'H$\beta$')
    plt.plot(m3d.vel_tab,m3d.get_profile('N__2_658345A', axis='x')[:
 ↪,x_pos,y_pos] * 5, label = r'[NII]$\lambda$6584')
    plt.plot(m3d.vel_tab,m3d.get_profile('O__3_500684A', axis='x')[:
 ↪,x_pos,y_pos], label = r'[OIII]$\lambda$5007')
    plt.legend()
```

[8]:
```
def other_plots(m3d, proj_axis):
    plt.subplot(331)
    plt.imshow(m3d.get_emis('H__1_486132A').sum(axis = proj_axis)*m3d.cub_coord.
 ↪cell_size)
    plt.title('Hb')
    plt.colorbar()

    plt.subplot(332)
    plt.imshow(m3d.get_emis('N__2_658345A').sum(axis = proj_axis)*m3d.cub_coord.
 ↪cell_size)
    plt.title('[NII]')
    plt.colorbar()

    plt.subplot(333)
    plt.imshow(m3d.get_emis('O__3_500684A').sum(axis = proj_axis)*m3d.cub_coord.
 ↪cell_size)
    plt.title('[OIII]')
    plt.colorbar()

    plt.subplot(334)
    plt.imshow(m3d.get_emis('N__2_658345A').sum(axis = proj_axis)/m3d.
↪get_emis('H__1_486132A').sum(axis = proj_axis))
    plt.title('[NII]/Hb')
    plt.colorbar()

    plt.subplot(335)
    plt.imshow(m3d.get_emis('O__3_500684A').sum(axis = proj_axis)/m3d.
↪get_emis('H__1_486132A').sum(axis = proj_axis))
    plt.title('[OIII]/Hb')
    plt.colorbar()

    plt.subplot(336)
    plt.imshow(m3d.get_ionic('O',1)[n_cut,:,:])
    plt.title('O+ cut')
    plt.colorbar()

    plt.subplot(337)
    plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.
↪get_ionic('O',1).ravel(),
                c=np.abs(m3d.cub_coord.theta.ravel()), edgecolors = 'none')
    plt.title('Colored by |Theta|')
    plt.xlabel('O+ / O')
    plt.ylabel('N+/O+ / N/O')
    plt.colorbar()

    plt.subplot(338)
    plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.
↪get_ionic('O',1).ravel(),
                c=m3d.relative_depth.ravel(),vmin = 0, vmax = 1, edgecolors =␣
↪'none')
    plt.title('Colored by position in the nebula')
    plt.xlabel('O+ / O')
    plt.ylabel('N+/O+ / N/O')
    plt.colorbar()

    plt.subplot(339)
    C1 = (m3d.get_ionic('N',1)/m3d.get_ionic('O',1)*m3d.get_ionic('N',2))
    C2 = (m3d.get_ionic('N',2))
    tt = (m3d.get_ionic('O',1) == 0)
    C1[tt] = 0
    C2[tt] = 0
    V = C1.sum(axis = proj_axis) / C2.sum(axis = proj_axis)
    plt.imshow(V)
    plt.colorbar()
    plt.title('N+/O+ / N/O weighted by NII')
    plt.contour(V,levels=[1.0])
```

[9]:
```
model_name = "M3D_1"
pc.log_.calling = 'Model3D : ' + model_name
pc.log_.level = 3
```

[10]:
```
dim = 101
n_cut = int((dim-1) /2)
proj_axis = 0
```

[11]:
```
set_models(dir_, model_name)
```

```
     CloudyInput: Input writen in /tmp/models//M3D_1_0.in
     CloudyInput: Input writen in /tmp/models//M3D_1_18.in
     CloudyInput: Input writen in /tmp/models//M3D_1_36.in
     CloudyInput: Input writen in /tmp/models//M3D_1_54.in
     CloudyInput: Input writen in /tmp/models//M3D_1_72.in
     CloudyInput: Input writen in /tmp/models//M3D_1_90.in
```

[12]:
```
pc.print_make_file(dir_ = dir_)
pc.run_cloudy(dir_ = dir_, n_proc = 6, model_name = model_name, use_make = True)
```

```
     print_make_file: make_file_printed with cloudy.exe =
/usr/local/Cloudy/c25.00_rc2/source/cloudy.exe
     run_cloudy: running: make -j 6 name="M3D_1"
     run_cloudy: ending: make -j 6 name="M3D_1"
```

[13]:
```
liste_of_models = pc.load_models('{0}/{1}'.format(dir_, model_name),␣
  ↪list_elem=['H', 'He', 'C', 'N', 'O', 'Ar', 'Ne'],
                                           read_cont = False, read_grains =␣
  ↪False)
```

```
     CloudyModel /tmp/models/M3D_1_18: Creating CloudyModel for
/tmp/models/M3D_1_18
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.rad read
     CloudyModel /tmp/models/M3D_1_18: Number of zones: 181
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.phy read
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_H read
     CloudyModel /tmp/models/M3D_1_18: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_He read
     CloudyModel /tmp/models/M3D_1_18: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_C read
     CloudyModel /tmp/models/M3D_1_18: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_N read
     CloudyModel /tmp/models/M3D_1_18: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_O read
     CloudyModel /tmp/models/M3D_1_18: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_Ar read
     CloudyModel /tmp/models/M3D_1_18: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.ele_Ne read
     CloudyModel /tmp/models/M3D_1_18: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_18: /tmp/models/M3D_1_18.emis read
     CloudyModel /tmp/models/M3D_1_18: Number of emissivities: 18
     CloudyModel /tmp/models/M3D_1_36: Creating CloudyModel for
/tmp/models/M3D_1_36
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.rad read
     CloudyModel /tmp/models/M3D_1_36: Number of zones: 180
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.phy read
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_H read
     CloudyModel /tmp/models/M3D_1_36: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_He read
     CloudyModel /tmp/models/M3D_1_36: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_C read
     CloudyModel /tmp/models/M3D_1_36: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_N read
     CloudyModel /tmp/models/M3D_1_36: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_O read
     CloudyModel /tmp/models/M3D_1_36: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_Ar read
     CloudyModel /tmp/models/M3D_1_36: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.ele_Ne read
     CloudyModel /tmp/models/M3D_1_36: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_36: /tmp/models/M3D_1_36.emis read
     CloudyModel /tmp/models/M3D_1_36: Number of emissivities: 18
     CloudyModel /tmp/models/M3D_1_90: Creating CloudyModel for
/tmp/models/M3D_1_90
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.rad read
     CloudyModel /tmp/models/M3D_1_90: Number of zones: 177
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.phy read
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_H read
     CloudyModel /tmp/models/M3D_1_90: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_He read
     CloudyModel /tmp/models/M3D_1_90: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_C read
     CloudyModel /tmp/models/M3D_1_90: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_N read
     CloudyModel /tmp/models/M3D_1_90: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_O read
     CloudyModel /tmp/models/M3D_1_90: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_Ar read
     CloudyModel /tmp/models/M3D_1_90: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.ele_Ne read
     CloudyModel /tmp/models/M3D_1_90: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_90: /tmp/models/M3D_1_90.emis read
     CloudyModel /tmp/models/M3D_1_90: Number of emissivities: 18
     CloudyModel /tmp/models/M3D_1_0: Creating CloudyModel for
/tmp/models/M3D_1_0
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.rad read
     CloudyModel /tmp/models/M3D_1_0: Number of zones: 181
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.phy read
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_H read
     CloudyModel /tmp/models/M3D_1_0: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_He read
     CloudyModel /tmp/models/M3D_1_0: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_C read
     CloudyModel /tmp/models/M3D_1_0: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_N read
     CloudyModel /tmp/models/M3D_1_0: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_O read
     CloudyModel /tmp/models/M3D_1_0: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_Ar read
     CloudyModel /tmp/models/M3D_1_0: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.ele_Ne read
     CloudyModel /tmp/models/M3D_1_0: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_0: /tmp/models/M3D_1_0.emis read
     CloudyModel /tmp/models/M3D_1_0: Number of emissivities: 18
     CloudyModel /tmp/models/M3D_1_54: Creating CloudyModel for
/tmp/models/M3D_1_54
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.rad read
     CloudyModel /tmp/models/M3D_1_54: Number of zones: 180
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.phy read
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_H read
     CloudyModel /tmp/models/M3D_1_54: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_He read
     CloudyModel /tmp/models/M3D_1_54: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_C read
     CloudyModel /tmp/models/M3D_1_54: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_N read
     CloudyModel /tmp/models/M3D_1_54: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_O read
     CloudyModel /tmp/models/M3D_1_54: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_Ar read
     CloudyModel /tmp/models/M3D_1_54: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.ele_Ne read
     CloudyModel /tmp/models/M3D_1_54: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_54: /tmp/models/M3D_1_54.emis read
     CloudyModel /tmp/models/M3D_1_54: Number of emissivities: 18
     CloudyModel /tmp/models/M3D_1_72: Creating CloudyModel for
/tmp/models/M3D_1_72
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.rad read
     CloudyModel /tmp/models/M3D_1_72: Number of zones: 178
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.phy read
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_H read
     CloudyModel /tmp/models/M3D_1_72: filling H with 3 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_He read
     CloudyModel /tmp/models/M3D_1_72: filling He with 3 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_C read
     CloudyModel /tmp/models/M3D_1_72: filling C with 13 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_N read
     CloudyModel /tmp/models/M3D_1_72: filling N with 8 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_O read
     CloudyModel /tmp/models/M3D_1_72: filling O with 12 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_Ar read
     CloudyModel /tmp/models/M3D_1_72: filling Ar with 19 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.ele_Ne read
     CloudyModel /tmp/models/M3D_1_72: filling Ne with 11 columns
     CloudyModel /tmp/models/M3D_1_72: /tmp/models/M3D_1_72.emis read
     CloudyModel /tmp/models/M3D_1_72: Number of emissivities: 18
     load_models: 6 models read
```

[14]:
```
M=liste_of_models[0]
M.emis_labels
```

[14]:
```
array(['H__1_486132A', 'H__1_656280A', 'CA_B_587564A', 'N__2_658345A',
       'O__1_630030A', 'O__2_372603A', 'O__2_372881A', 'O__3_500684A',
       'O__3_436321A', 'O_3R_436300A', 'O_3C_436300A', 'S__2_671644A',
       'S__2_673082A', 'CL_3_551771A', 'CL_3_553787A', 'O__1_631679M',
       'O__1_145495M', 'C__2_157636M'], dtype='<U12')
```

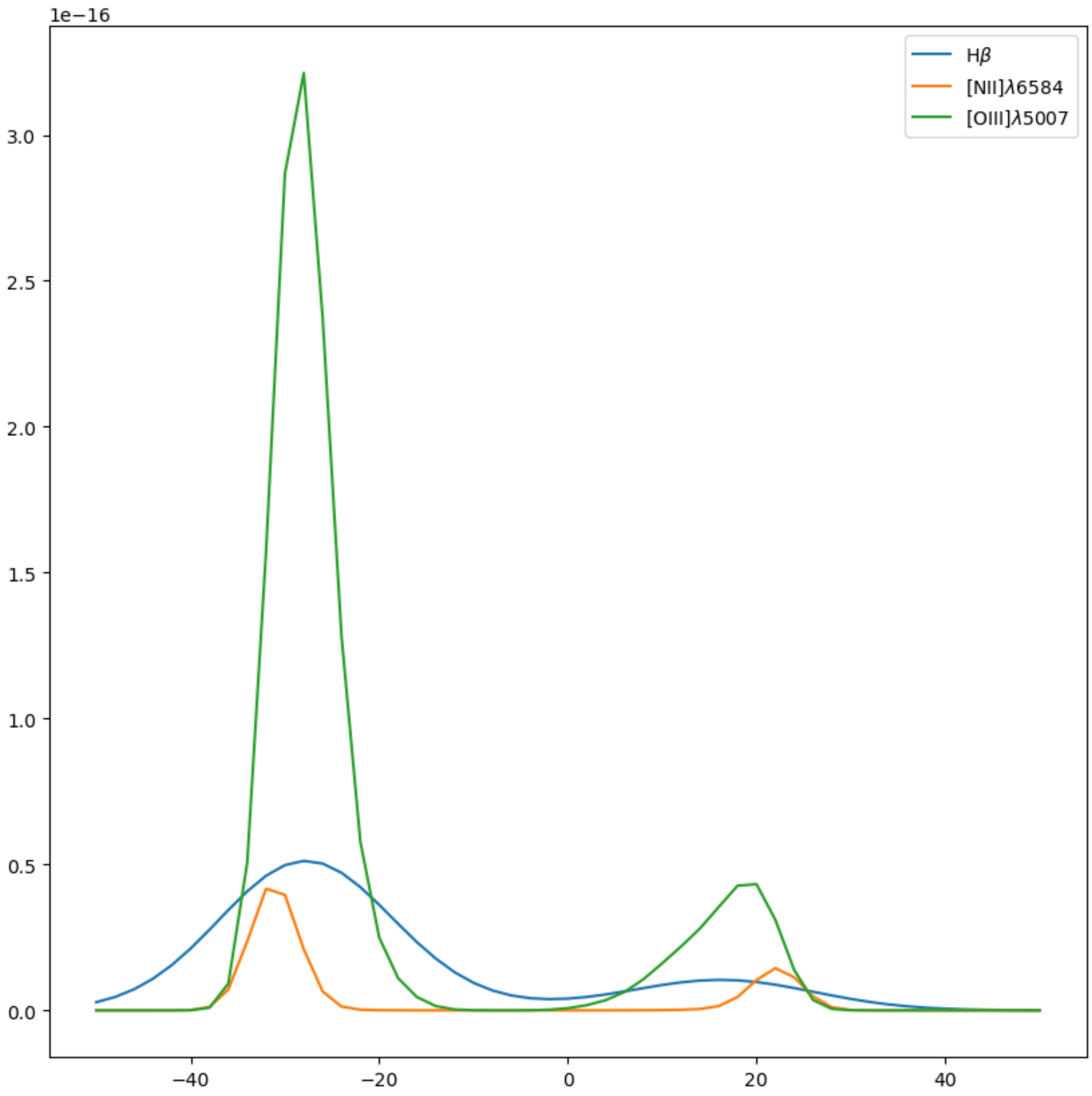
[15]:
```
m3d = pc.C3D(liste_of_models, dims = [dim, dim, dim], angles = [45,45,0],␣
 ↪plan_sym = True)
```

```
C3D: Entering C3D
CubCoord: building a cube of 101x101x101
CubCoord: Rotation matrix by 45.0, 45.0, 0.0 degrees.
C3D: CubCoord done.
C3D: interp_bi done.
C3D: Interpolation mesh done
C3D: All 3D values reset
```

[16]:
```
def_profiles(m3d)
```

[17]:
```
plt.figure(figsize=(10,10))
plot_profiles(m3d, 55, 55)
```

```
C3D: get_emis(0) interpolated using numpy-method
C3D: te interpolated using numpy-method
C3D: line H__1_486132A : profile computed on axis x
C3D: get_emis(3) interpolated using numpy-method
C3D: line N__2_658345A : profile computed on axis x
C3D: get_emis(7) interpolated using numpy-method
C3D: line O__3_500684A : profile computed on axis x
```



[18]:

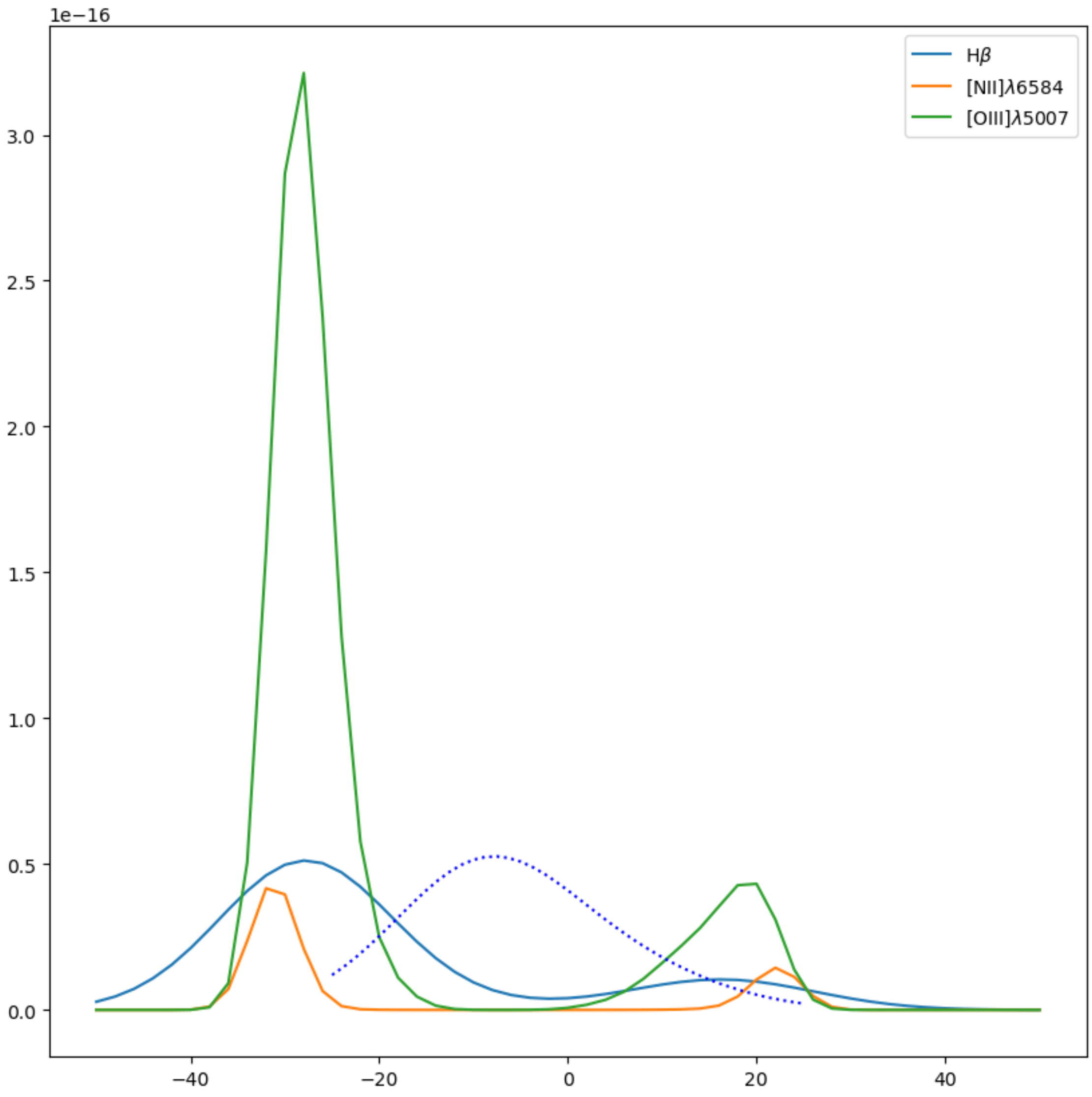
```
plt.figure(figsize=(10,10))
plot_profiles(m3d, 55, 55)
def_profiles_user(m3d)
plt.plot(m3d.vel_tab,m3d.get_profile('H__1_486132A', axis='x')[:,55,55] * 5, ':
  ↪b', label = r'H$\beta$')
```

```
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/3043622111.py:1
7: RuntimeWarning: divide by zero encountered in divide
  tmp = tmp / cub_coord.r
```

```
    C3D: line H__1_486132A : profile computed on axis x
```

[18]:

```
[<matplotlib.lines.Line2D at 0x104bf1700>]
```



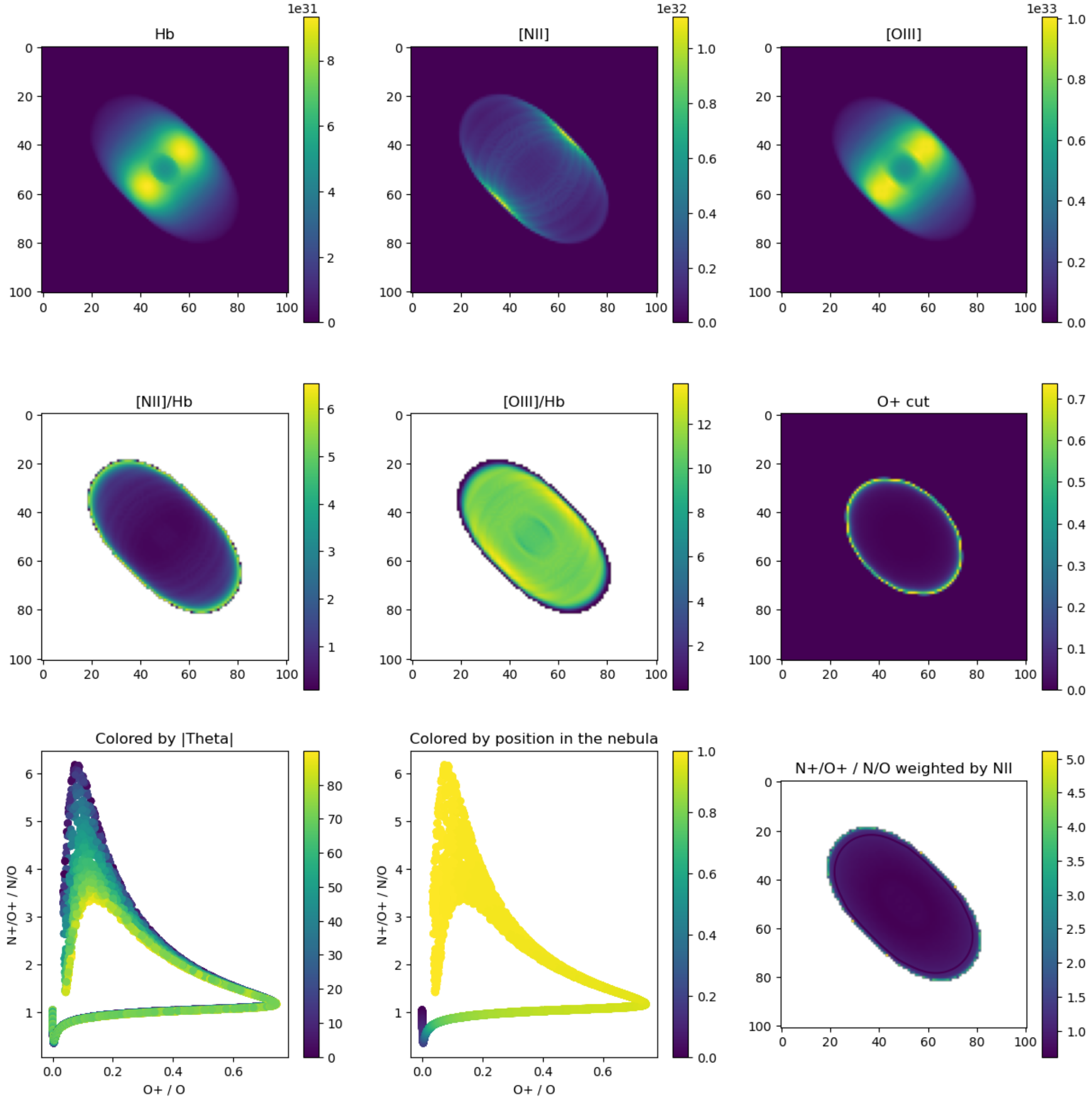
[19]:

```
plt.figure(figsize=(15,15))
other_plots(m3d, proj_axis)
```

```
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:18:
RuntimeWarning: invalid value encountered in divide
  plt.imshow(m3d.get_emis('N__2_658345A').sum(axis =
proj_axis)/m3d.get_emis('H__1_486132A').sum(axis = proj_axis))
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:23:
RuntimeWarning: invalid value encountered in divide
  plt.imshow(m3d.get_emis('O__3_500684A').sum(axis =
proj_axis)/m3d.get_emis('H__1_486132A').sum(axis = proj_axis))
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:33:
RuntimeWarning: invalid value encountered in divide
```

```
  plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.get_
ionic('O',1).ravel(),
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:41:
RuntimeWarning: invalid value encountered in divide
  plt.scatter(m3d.get_ionic('O',1).ravel(),m3d.get_ionic('N',1).ravel()/m3d.get_
ionic('O',1).ravel(),
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:49:
RuntimeWarning: invalid value encountered in divide
  C1 = (m3d.get_ionic('N',1)/m3d.get_ionic('O',1)*m3d.get_ionic('N',2))
/var/folders/7b/7gktm_g91hn54p3gxj15kb1m0000gn/T/ipykernel_10174/32881221.py:54:
RuntimeWarning: invalid value encountered in divide
  V = C1.sum(axis = proj_axis) / C2.sum(axis = proj_axis)
```

```
    C3D: get_ionic('O', 1) interpolated using numpy-method
    C3D: get_ionic('N', 1) interpolated using numpy-method
    C3D: get_ionic('N', 2) interpolated using numpy-method
```

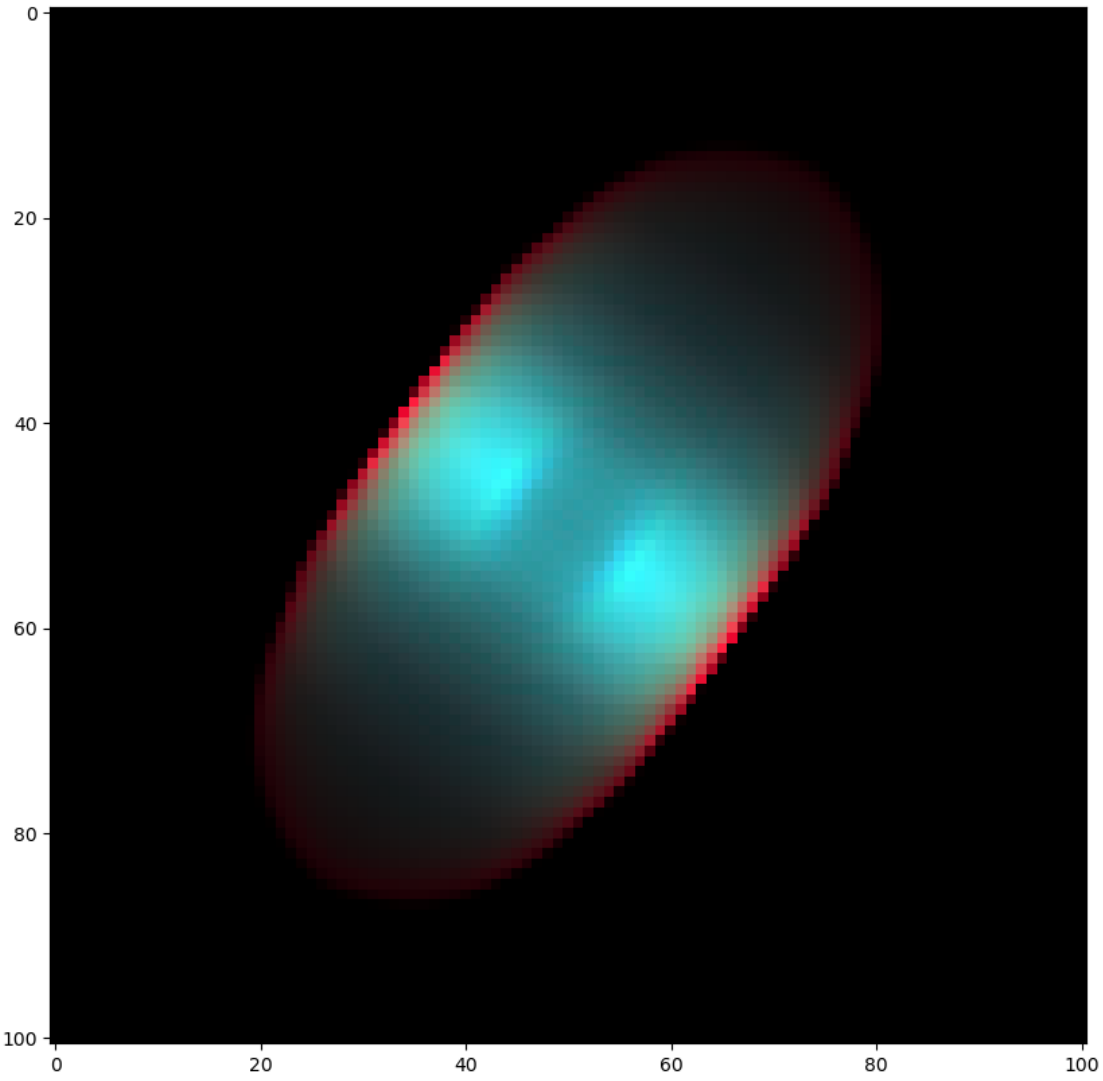


[25]:

```
im = m3d.get_RGB(list_emis = ['N__2_658345A', 'O__3_500684A', 'H__1_486132A'])
plt.figure(1, figsize=(10,10))
plt.imshow(im)
```

[25]:

```
<matplotlib.image.AxesImage at 0x176124f50>
```



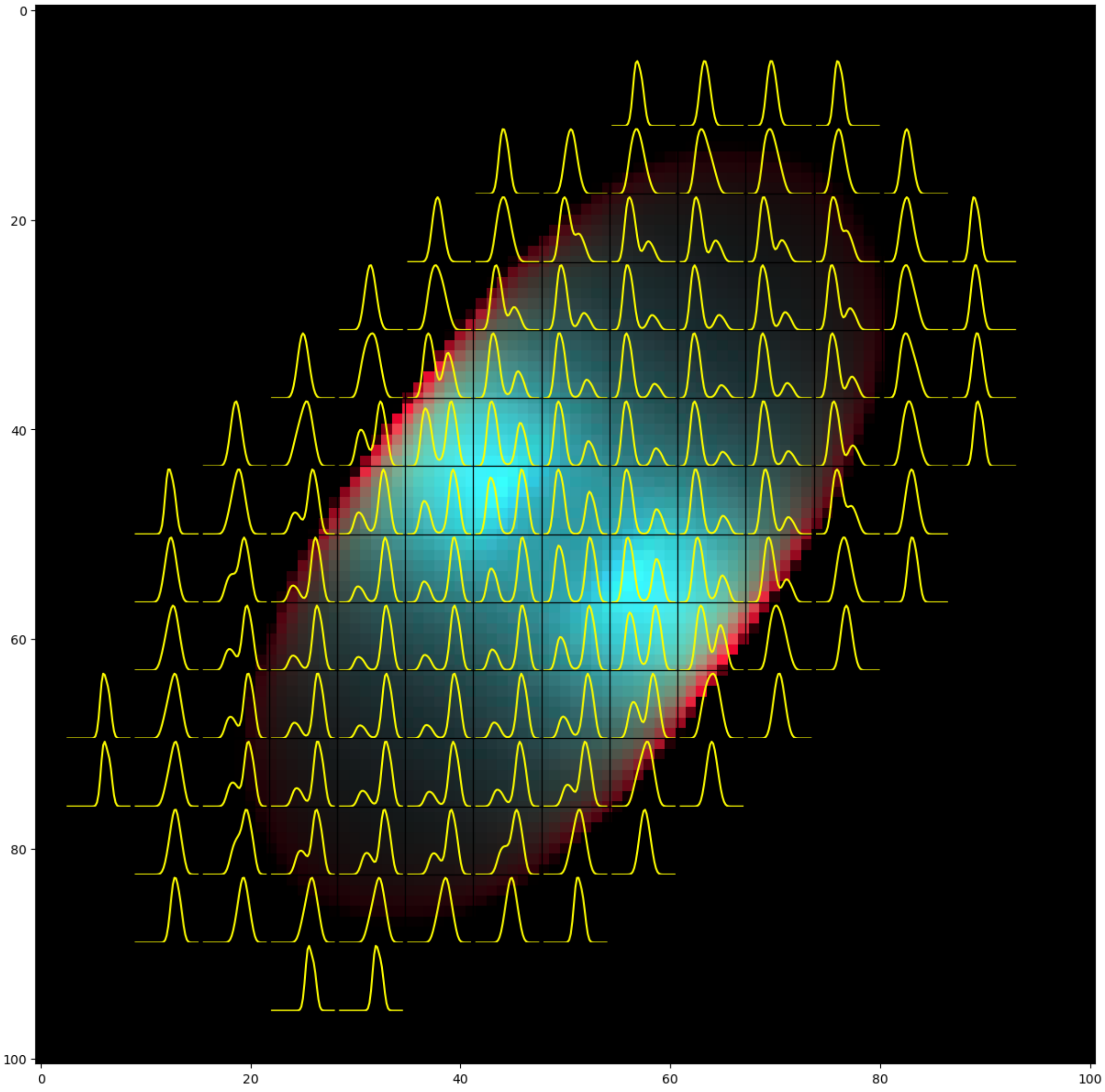
[26]:

```
im = m3d.get_RGB(list_emis = ['N__2_658345A', 'O__3_500684A', 'H__1_486132A'])
plt.figure(1, figsize=(15,15))
plt.imshow(im)
m3d.plot_profiles(ref = 3, i_fig = 1, Nx=20, Ny=20)
```

```
C3D: line N__2_658345A : profile computed on axis x
```

```
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
```

```
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
/Users/christophemorisset/Google
Drive/Pro/pyCloudy/pyCloudy/c3d/model_3d.py:946: RuntimeWarning: invalid value
encountered in divide
  prof /= np.max(prof)
```


0
20
40
60
80
100
0
20
40
60
80
100