In [27]:

```
import numpy as np
import matplotlib.pyplot as plt
import pynbody

from michaels_functions import center_and_r_vir, remove_bulk_velocity
from matplotlib.colors import LogNorm
```

In [28]:

```
path = "bulk1/data_2/hydro_59/output/"
data = pynbody.load(path + "output_00050")

aexp = data.properties['a']
data.physical_units()

print path
print "a =", aexp
print "z =", 1./aexp -1
```

```
bulk1/data_2/hydro_59/output/
a = 0.600005205268
z = 0.666652207716
```

In [29]:

```
r_vir = center_and_r_vir(data, aexp, path)
```

```
('shifting on Stars:', SimArray([ 0.00308878, -0.03302971, -0.046767
32], 'kpc'))
('virial radius:', SimArray(152.40432611, 'kpc'))
```

In [30]:

```
r_e = 0.1 * r_vir
print r_e
```

```
15.240432611151682
```

In [31]:

```
sph_5 = pynbody.filt.Sphere(radius = '%f kpc' %(r_e*1.4))
region = data[sph_5]
```

In [32]:

```
rho = region.gas["rho"].in_units("m_p cm^-3")
```

In [33]:

```
Z = region.gas["metal"]
```

In [34]:

```
f = open(data.filename + "/info_"+data.filename[-5:]+".txt","r")
lines = f.readlines()
f.close()

for line in lines:
    if line[0:13]=="unit_l      =":
        print line[:-1]
        unit_l = float(line[14:-1])
    if line[0:13]=="unit_d      =":
        print line[:-1]
        unit_d = float(line[14:-1])
    if line[0:13]=="unit_t      =":
        print line[:-1]
        unit_t = float(line[14:-1])
    if line[0:13]=="omega_b     =":
        print line[:-1]
        omega_b = float(line[14:-1])
```

```
omega_b     =  0.450000017881393E-01
unit_l      =  0.682025380323961E+26
unit_d      =  0.123367583719985E-28
unit_t      =  0.163687521954501E+18
```

In [35]:

```
turb = np.sqrt( region.g["turb"] * 2./3. ) * unit_l / unit_t / 1e5
turb = pynbody.array.SimArray(turb, units = "cm s**-1")
c_s = np.sqrt(region.gas["p"] / region.gas["rho"])
c_s = c_s.in_units('cm s**-1')
M = turb / c_s
region.g["mach"] = M.in_units("1")
```

In [36]:

```
turb
```

Out[36]:

```
SimArray([29.15456777, 44.73850117, 68.11498304, ..., 29.26368001,
          16.37539853, 25.15026196], 'cm s**-1')
```

In [37]:

```
c_s
```

Out[37]:

```
SimArray([ 3276673.11939746, 12738195.66072513,  9202021.83758316,
...,
          14469186.19848063, 13460450.29314729, 14119970.07822348],
'cm s**-1')
```

In [38]:

```
M
```

Out[38]:

```
SimArray([8.89761252e-06, 3.51215371e-06, 7.40217577e-06, ...,
          2.02248279e-06, 1.21655652e-06, 1.78118380e-06], '1.00e+0
0')
```

In [39]:

```
m_p_1 = pynbody.array.SimArray(1.0, pynbody.units.m_p)
n_H = rho / m_p_1
```

In [40]:

```
n_H
```

Out[40]:

```
SimArray([0.00242818, 0.00016367, 0.00038948, ..., 0.00032273, 0.000
23647,
          0.00029291], 'cm**-3')
```

In [41]:

```
m_p = pynbody.array.SimArray(1.672621777e-24, "g")
K_b = pynbody.array.SimArray(1.38064852e-16, "cm**2 g s**-2 K**-1")
G = pynbody.array.SimArray(6.67259e-8, "cm**3 g**-1 s**-2")
T_mean = pynbody.array.SimArray(10., "K")
G_o = pynbody.array.SimArray(1.0, "1")
n_H_mean = pynbody.array.SimArray(1e2, "cm**-3")
```

In [42]:

```
lambda_jeans = (c_s) / np.sqrt(4* np.pi * G * n_H * m_p)
lambda_jeans
```

Out[42]:

```
SimArray([5.61490010e+22, 8.40766919e+23, 3.93723266e+23, ...,
          6.80104626e+23, 7.39131101e+23, 6.96656552e+23], 'cm')
```

In [ ]:

```
'''vel = region.g["vel"] * unit_l / unit_t / 1e5
vel = pynbody.array.SimArray(vel, units = "cm s**-1")'''
```

In [ ]:

```
'''sigma_s = pynbody.array.SimArray(vel, "1")
sigma_s'''
```

In [ ]:

```
'''s_bar = -0.5*(sigma_s**2)
s_bar'''
```

In [ ]:

```
'''smin = -7*sigma_s + s_bar
smax = 7*sigma_s + s_bar
ds = (smax - smin)/1040042'''
```

In [ ]:

```
'''smax[-1]'''
```

In [ ]:

```
'''ds[-1]'''
```

In [ ]:

```
'''smin[-1]'''
```

In [ ]:

```
'''s = np.zeros(1040042)
for i in range(0, 1040042):
    s = smin + i*ds'''
```

In [ ]:

```
'''n_H_2 = n_H * np.exp(s)'''
```

In [ ]:

```
'''pdf = (1/np.sqrt(2*np.pi*(sigma_s**2))) * (np.exp(-0.5*(((s - s_bar)/sigma_s)
**2)))
pdf'''
```

In [ ]:

```
'''integral1 = 0.0
for i in range(0, 1000):
    integral1 += np.exp(s[i]) * pdf[i] * ds[i]   #this should be = 1
    #plotting(n_H, pdf, lambda_jeans, X_H2)
integral1'''
```

In [ ]:

```
'''plt.scatter(np.log10(n_H), pdf)
plt.xlabel('log(n_H)')
plt.ylabel('log(pdf)')
plt.grid(b=True, which='both', axis='both')
plt.title('log(n_H) vs log(pdf)')
plt.show()'''
```

In [43]:

```
def calc_n_LW(n_H, G_o, lambda_jeans):
    kappa = 1000 * m_p
    rad_field_outside = G_o #in solar units
    exp_tau = np.exp(-kappa * n_H * lambda_jeans)
    n_LW = rad_field_outside * exp_tau
    return n_LW
```

In [44]:

```
def calc_n_LW_ss(n_H, n_H2, G_o, lambda_jeans):
    kappa = 1000 * m_p
    rad_field_outside = G_o #in solar units
    exp_tau = np.exp(-kappa * n_H * lambda_jeans)
    N_H2 = n_H2*lambda_jeans
    term1 = (0.965/((1+(N_H2/5e14))**2))
    term2 = ( (0.035/np.sqrt(1+(N_H2/5e14))) * np.exp(-1*np.sqrt(1+(N_H2/5e14))/
1180) )
    S_H2 = term1 + term2
    n_LW_ss = rad_field_outside * exp_tau * S_H2
    return n_LW_ss, S_H2, N_H2
```

In [45]:

```
def calc_X_H2(n_H, Z, n_LW):
    DC = 1.7e-11
    CC = 2.5e-17            #cm3 s-1
    numerator = DC * n_LW
    denominator = CC * Z * n_H
    X_H2 = 1 / (2 + (numerator/denominator) )
    return X_H2
```

In [46]:

```
def calc_X_CO(n_H, n_H2, n_LW):
    rate_CHX = 5.e-10 * n_LW
    rate_CO = 1.e-10 * n_LW
    x0 = 2.e-4
    k0 = 5.e-16 #cm3 s-1
    k1 = 5.e-10 #cm3 s-1
    factor_beta = rate_CHX/(n_H*k1*x0)
    beta = 1./(1.+factor_beta)
    factor_CO = rate_CO/(n_H2*k0*beta)
    X_CO = 1./(1.+factor_CO)
    return X_CO
```

In [47]:

```
def calc_n_CO(n_H, X_CO):
    abundance_Ctot = 1e-4 # n_C/n_H as defined by nucleosynthesis
    return n_H * abundance_Ctot * X_CO # CO/cc
```

In [48]:

```
n_LW = calc_n_LW(n_H, G_o, lambda_jeans)
X_H2_a = calc_X_H2(n_H, Z, n_LW)
n_H2_a = n_H * X_H2_a
```

In [49]:

```
n_LW_1, S_H2_1, N_H2_1 = calc_n_LW_ss(n_H, n_H2_a, G_o, lambda_jeans)
X_H2_1 = calc_X_H2(n_H, Z, n_LW_1)
n_H2_1 = n_H * X_H2_1
```

In [50]:

```
n_LW_2, S_H2_2, N_H2_2 = calc_n_LW_ss(n_H, n_H2_1, G_o, lambda_jeans)
X_H2_2 = calc_X_H2(n_H, Z, n_LW_2)
n_H2_2 = n_H * X_H2_2
```

In [51]:

```
n_LW_3, S_H2_3, N_H2_3 = calc_n_LW_ss(n_H, n_H2_2, G_o, lambda_jeans)
X_H2_3 = calc_X_H2(n_H, Z, n_LW_3)
n_H2_3 = n_H * X_H2_3
```

In [52]:

```
n_LW_4, S_H2_4, N_H2_4 = calc_n_LW_ss(n_H, n_H2_3, G_o, lambda_jeans)
X_H2_4 = calc_X_H2(n_H, Z, n_LW_4)
n_H2_4 = n_H * X_H2_4
```

In [53]:

```
n_LW_5, S_H2_5, N_H2_5 = calc_n_LW_ss(n_H, n_H2_4, G_o, lambda_jeans)
X_H2_5 = calc_X_H2(n_H, Z, n_LW_5)
n_H2_5 = n_H * X_H2_5
```

In [54]:

```
n_LW_6, S_H2_6, N_H2_6 = calc_n_LW_ss(n_H, n_H2_5, G_o, lambda_jeans)
X_H2_6 = calc_X_H2(n_H, Z, n_LW_6)
n_H2_6 = n_H * X_H2_6
```

In [55]:

```
n_LW_7, S_H2_7, N_H2_7 = calc_n_LW_ss(n_H, n_H2_6, G_o, lambda_jeans)
X_H2_7 = calc_X_H2(n_H, Z, n_LW_7)
n_H2_7 = n_H * X_H2_7
```

In [56]:

```
n_LW_8, S_H2_8, N_H2_8 = calc_n_LW_ss(n_H, n_H2_7, G_o, lambda_jeans)
X_H2_8 = calc_X_H2(n_H, Z, n_LW_8)
n_H2_8 = n_H * X_H2_8
```

In [57]:

```
n_LW_9, S_H2_9, N_H2_9 = calc_n_LW_ss(n_H, n_H2_8, G_o, lambda_jeans)
X_H2_9 = calc_X_H2(n_H, Z, n_LW_9)
n_H2_9 = n_H * X_H2_9
```

In [58]:

```
n_LW_10, S_H2_10, N_H2_10 = calc_n_LW_ss(n_H, n_H2_9, G_o, lambda_jeans)
X_H2_10 = calc_X_H2(n_H, Z, n_LW_10)
n_H2_10 = n_H * X_H2_10
```

In [59]:

```
n_LW_ss, S_H2, N_H2 = calc_n_LW_ss(n_H, n_H2_10, G_o, lambda_jeans)
X_H2 = calc_X_H2(n_H, Z, n_LW_ss)
n_H2 = n_H * X_H2
```

In [60]:

```
X_CO = calc_X_CO(n_H, n_H2_a, n_LW)
```

In [61]:

```
n_CO = calc_n_CO(n_H, X_CO)
```

In [62]:

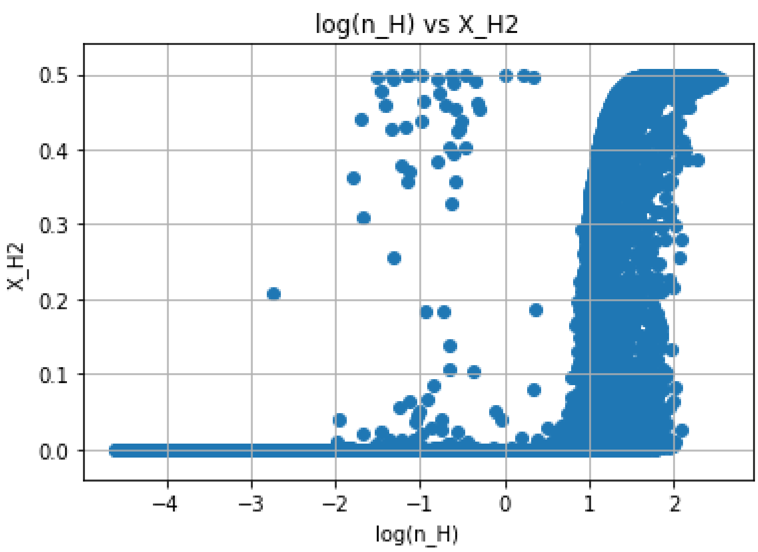
```
X_H2
```

Out[62]:

```
SimArray([2.42520898e-12, 1.85451271e-13, 4.17934370e-13, ...,
          3.49206881e-13, 2.15039016e-13, 2.99809526e-13])
```

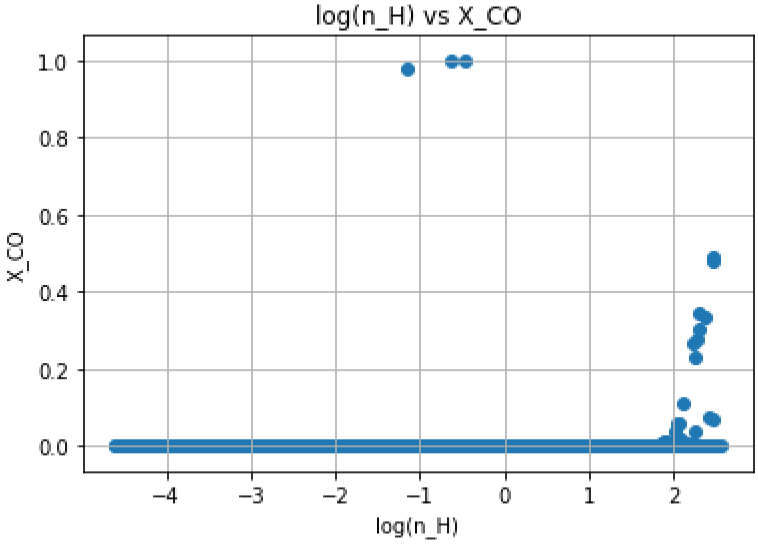
In [63]:

```
plt.scatter(np.log10(n_H), X_H2)
plt.xlabel('log(n_H)')
plt.ylabel('X_H2')
plt.grid(b=True, which='both', axis='both')
plt.title('log(n_H) vs X_H2')
#plt.savefig('log(n_H)vsX_H2.png'.format())
plt.show()
```



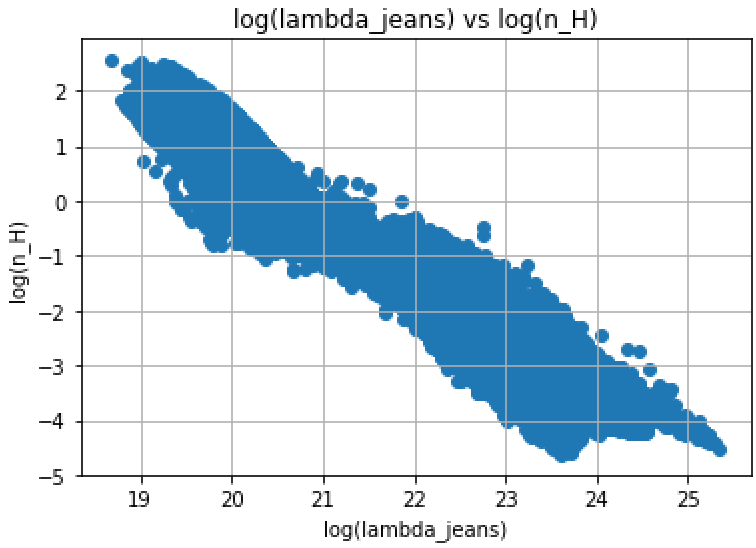
In [64]:

```
plt.scatter(np.log10(n_H), X_CO)
plt.xlabel('log(n_H)')
plt.ylabel('X_CO')
plt.grid(b=True, which='both', axis='both')
plt.title('log(n_H) vs X_CO')
#plt.savefig('log(n_H)vsX_CO.png'.format())
plt.show()
```



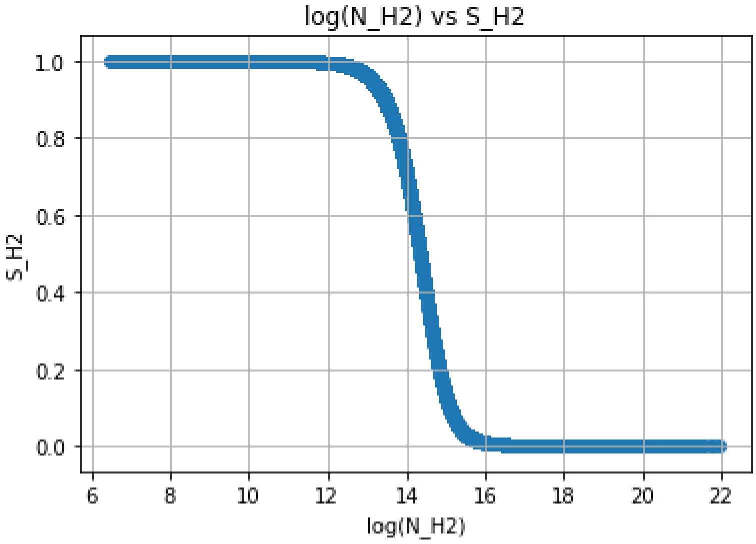
In [65]:

```
plt.scatter(np.log10(lambda_jeans), np.log10(n_H))
plt.xlabel('log(lambda_jeans)')
plt.ylabel('log(n_H)')
plt.grid(b=True, which='both', axis='both')
plt.title('log(lambda_jeans) vs log(n_H)')
#plt.savefig('log(lambda_jeans)vslog(n_H).png'.format())
plt.show()
```



In [66]:

```
plt.scatter(np.log10(N_H2), S_H2)
plt.xlabel('log(N_H2)')
plt.ylabel('S_H2')
plt.grid(b=True, which='both', axis='both')
plt.title("log(N_H2) vs S_H2")
#plt.savefig('log(N_H2)vsS_H2.png'.format())
plt.show()
```



In [67]:

```
region.gas["lambda_jeans"] = lambda_jeans
```

In [68]:

```
region.gas["n_H"] = n_H
```

In [69]:

```
region.gas["X_H2"] = X_H2
```

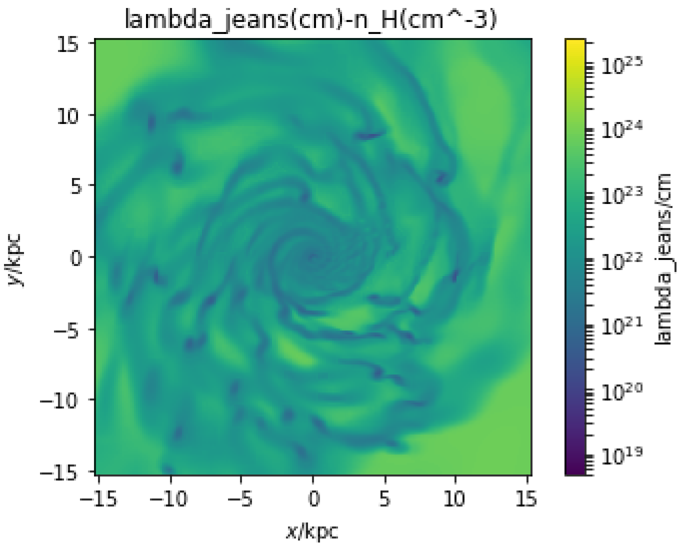
In [70]:

```
region.gas["X_CO"] = X_CO
```

In [71]:

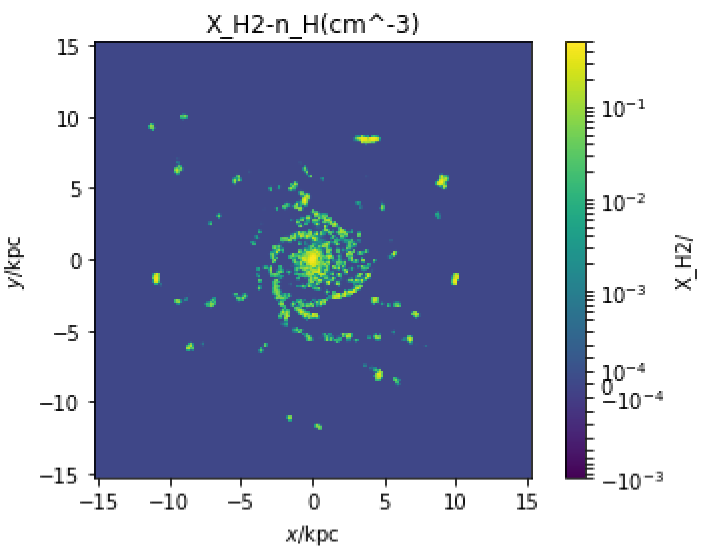
```
pynbody.plot.image(region.g, qty="lambda_jeans", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="n_H",
                   vmin=4.9e+18, vmax=2.3e+25)
plt.title("lambda_jeans(cm)-n_H(cm^-3)")
#plt.savefig('lambda_jeans-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```

```
/home/cluster/mkrets/anaconda2/lib/python2.7/site-packages/pynbody-
0.47-py2.7-linux-x86_64.egg/pynbody/snapshot/__init__.py:1443: Runti
meWarning: Conjoining derived and non-derived arrays. Assuming resul
t is non-derived, so no further updates will be made.
  "Conjoining derived and non-derived arrays. Assuming result is non
-derived, so no further updates will be made.", RuntimeWarning)
```



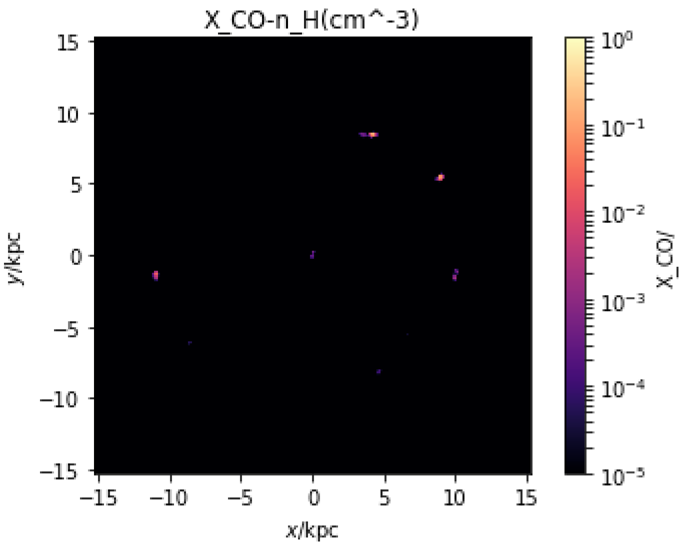
In [72]:

```
pynbody.plot.image(region.g, qty="X_H2", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="n_H",
                   vmin=-1e-3, vmax=5e-1)
plt.title("X_H2-n_H(cm^-3)")
#plt.savefig('X_H2-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```



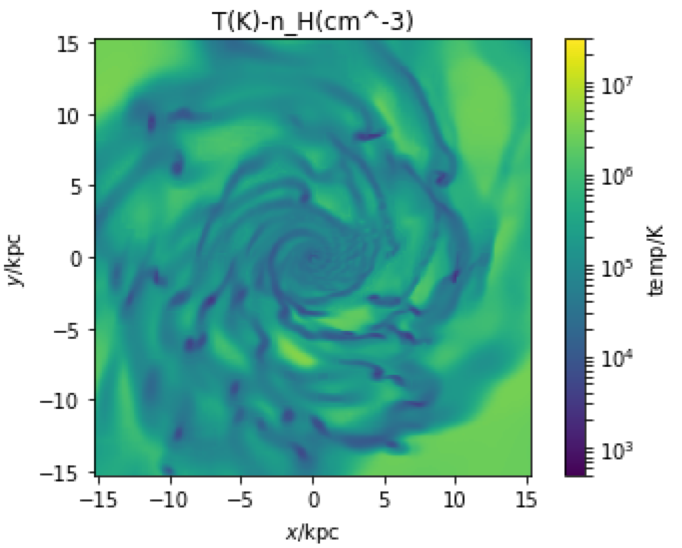
In [73]:

```
pynbody.plot.image(region.g, qty="X_CO", width="%f kpc"%(2.*r_e),
                   log=True, resolution=1000, cmap="magma", av_z="n_H",
                   vmin=1e-5, vmax=1)
plt.title("X_CO-n_H(cm^-3)")
#plt.savefig('X_CO-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```



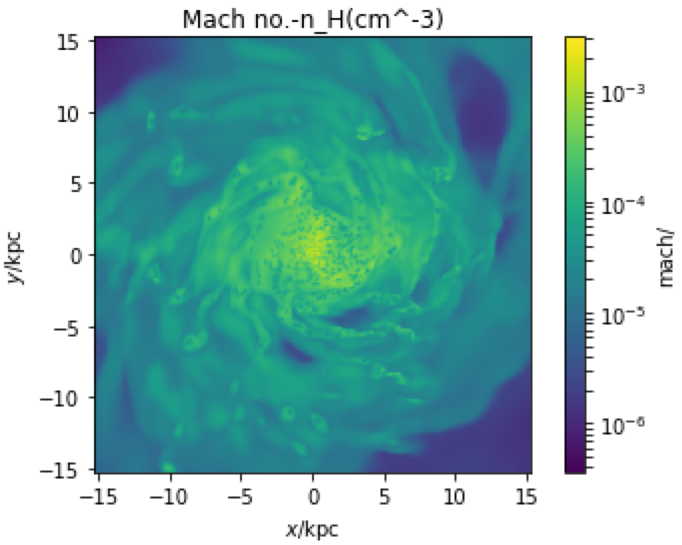
In [74]:

```
pynbody.plot.image(region.g, qty="temp", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="n_H",
                   vmin=0.5e3, vmax=0.3e8)
plt.title("T(K)-n_H(cm^-3)")
#plt.savefig('T-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```



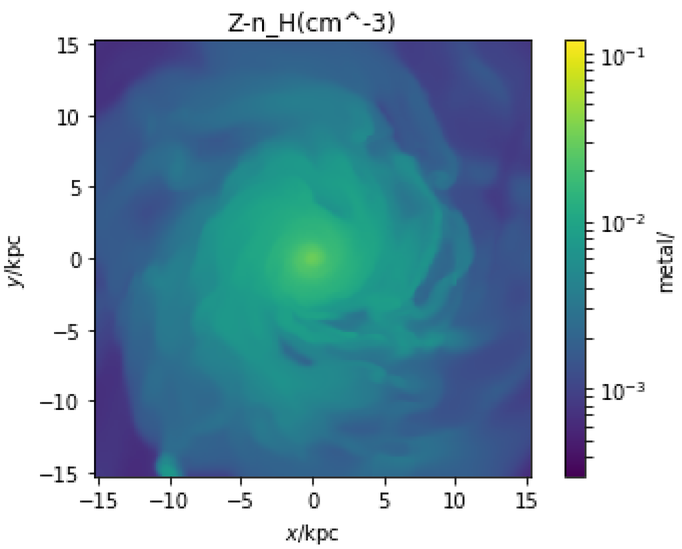
In [75]:

```
pynbody.plot.image(region.g, qty="mach", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="n_H",
                   vmin=3.6e-07, vmax=3.1e-3)
plt.title("Mach no.-n_H(cm^-3)")
#plt.savefig('Machno-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```



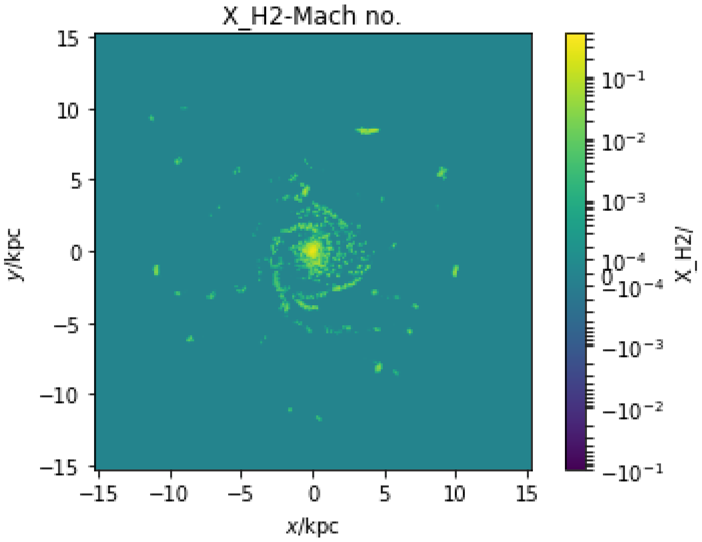
In [76]:

```
pynbody.plot.image(region.g, qty="metal", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="n_H",
                   vmin=3.03e-4, vmax=1.2e-1)
plt.title("Z-n_H(cm^-3)")
#plt.savefig('Z-n_H.png', dpi=300, bbox_inches='tight')
plt.show()
```



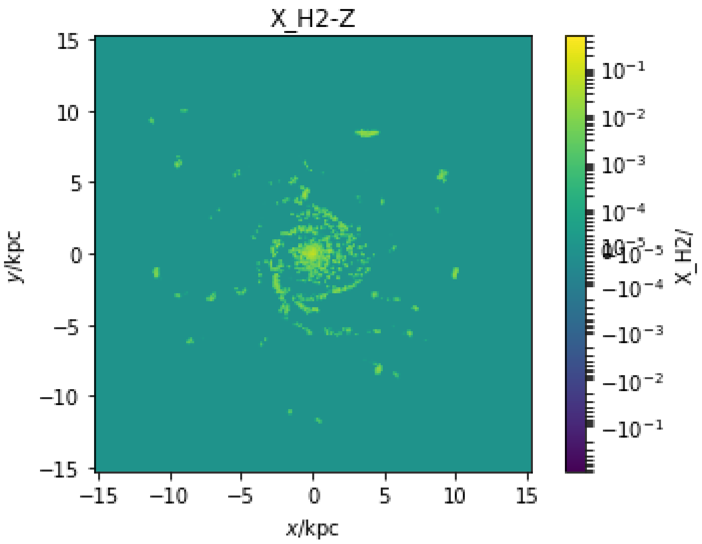
In [77]:

```
pynbody.plot.image(region.g, qty="X_H2", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="mach",
                   vmin=-0.1, vmax=5e-1)
plt.title("X_H2-Mach no.")
#plt.savefig('X_H2-Machno.png', dpi=300, bbox_inches='tight')
plt.show()
```



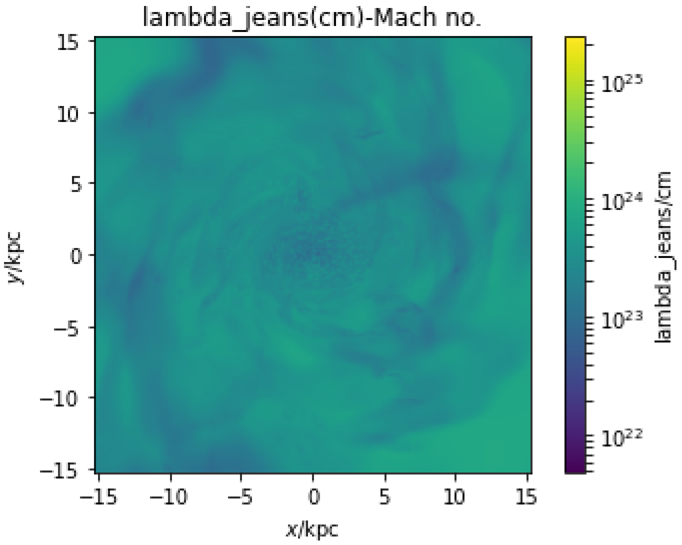
In [78]:

```
pynbody.plot.image(region.g, qty="X_H2", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="metal",
                   vmin=-0.9, vmax=5e-1)
plt.title("X_H2-Z")
#plt.savefig('X_H2-Z.png', dpi=300, bbox_inches='tight')
plt.show()
```



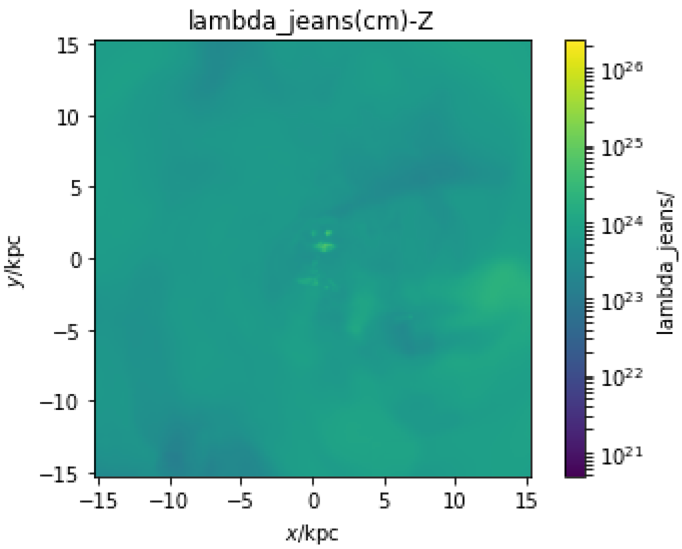
In [79]:

```
pynbody.plot.image(region.g, qty="lambda_jeans", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="mach",
                   vmin=4.9e+21, vmax=2.3e+25)
plt.title("lambda_jeans(cm)-Mach no.")
#plt.savefig('lambda_jeans-Machno.png', dpi=300, bbox_inches='tight')
plt.show()
```



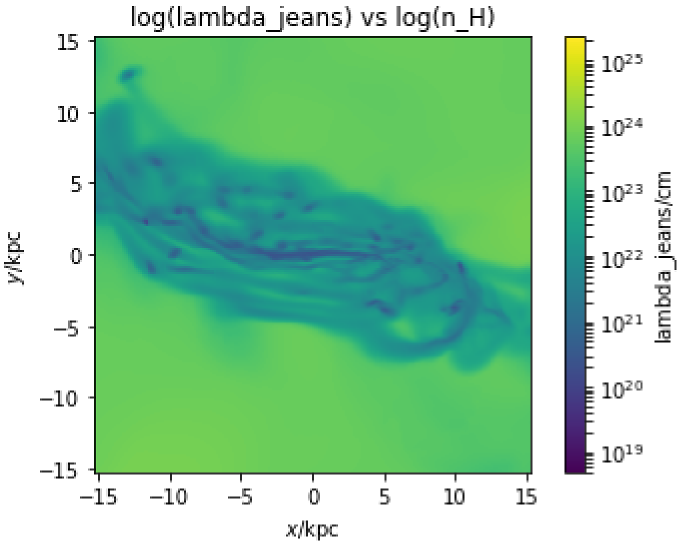
In [80]:

```
pynbody.plot.image(region.g, qty="lambda_jeans", width="%f kpc"%(2.*r_e),
                   log=True, resolution=500, cmap="viridis", av_z="metal",
                   vmin=4.9e+20, vmax=2.3e+26)

plt.title("lambda_jeans(cm)-Z")
#plt.savefig('lambda_jeans-Z.png', dpi=300, bbox_inches='tight')
plt.show()
```



In [81]:

```
with pynbody.analysis.angmom.sideon(region):
    pynbody.plot.image(region.g, qty="lambda_jeans", width="%f kpc"%(2.*r_e),
                       log=True, resolution=500, cmap="viridis", av_z="n_H",
                       vmin=4.9e+18, vmax=2.3e+25)
    plt.title('log(lambda_jeans) vs log(n_H)')
    plt.savefig('log(lambda_jeans)vslog(n_H)_sideon.png')
    plt.show()
```



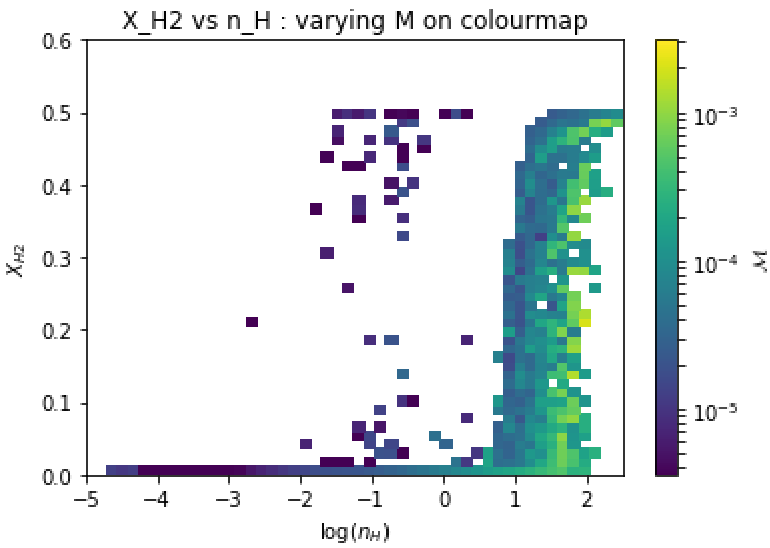
In [82]:

```
histX_H2_M_mass, yedges, xedges = np.histogram2d(region.gas["X_H2"],
                                   np.log10(region.gas["n_H"].in_units("cm**-3")),
                                   weights=region.gas["mach"] * region.gas["mass"],
bins=50 , range=[[0,0.6],[-5,2.5]])

histX_H2_mass, yedges, xedges = np.histogram2d(region.gas["X_H2"],
                                   np.log10(region.gas["n_H"].in_units("cm**-3")),
                                   weights=region.gas["mass"], bins=50 , range=[[0,
0.6],[-5,2.5]])

yX_H2_M, xX_H2_M = yedges, xedges
```

In [83]:

```
#Mach number - mass weighted - X_H2 vs n_H
plt.pcolormesh(xedges, yedges, histX_H2_M_mass/histX_H2_mass, norm=LogNorm(), vm
in=3.6e-6, vmax=3.1e-3)

plt.xlabel(r"$\log(n_H)$")
plt.ylabel(r"$X_{H2}$")
plt.colorbar(label=r"$\mathcal{M}$")
plt.title("X_H2 vs n_H : varying M on colourmap")
#plt.savefig('001_log(n_H)vsX_H2--M.png')
plt.show()
```

```
/net/cephfs/home/mkrets/anaconda2/lib/python2.7/site-packages/ipyker
nel_launcher.py:2: RuntimeWarning: invalid value encountered in divi
de

/home/cluster/mkrets/anaconda2/lib/python2.7/site-packages/matplotli
b/colors.py:1031: RuntimeWarning: invalid value encountered in less_
equal
  mask |= resdat <= 0
```



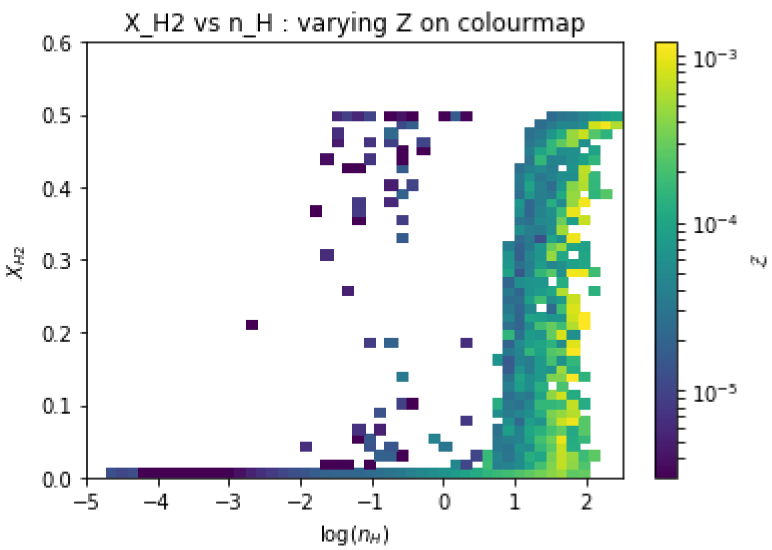
In [84]:

```
histX_H2_mass_Z, yedges, xedges = np.histogram2d(region.gas["X_H2"],
                                np.log10(region.gas["n_H"].in_units("cm**-3")),
                                weights=region.gas["metal"] * region.gas["mass"],
bins=50 , range=[[0,0.6],[-5,2.5]])

histX_H2_mass, yedges, xedges = np.histogram2d(region.gas["X_H2"],
                                np.log10(region.gas["n_H"].in_units("cm**-3")),
                                weights=region.gas["mass"], bins=50 , range=[[0,
0.6],[-5,2.5]])

yX_H2_Z, xX_H2_Z = yedges, xedges
```

In [85]:

```
#Z - mass weighted - X_H2 vs n_H
plt.pcolormesh(xedges, yedges, histX_H2_M_mass/histX_H2_mass, norm=LogNorm(), vm
in=3.03e-6, vmax=1.2e-3)

plt.xlabel(r"$\log(n_H)$")
plt.ylabel(r"$X_{H2}$")
plt.colorbar(label=r"$\mathcal{Z}$")
plt.title("X_H2 vs n_H : varying Z on colourmap")
#plt.savefig('002_log(n_H)vsX_H2--Z.png')
plt.show()
```

```
/net/cephfs/home/mkrets/anaconda2/lib/python2.7/site-packages/ipyker
nel_launcher.py:2: RuntimeWarning: invalid value encountered in divi
de
```



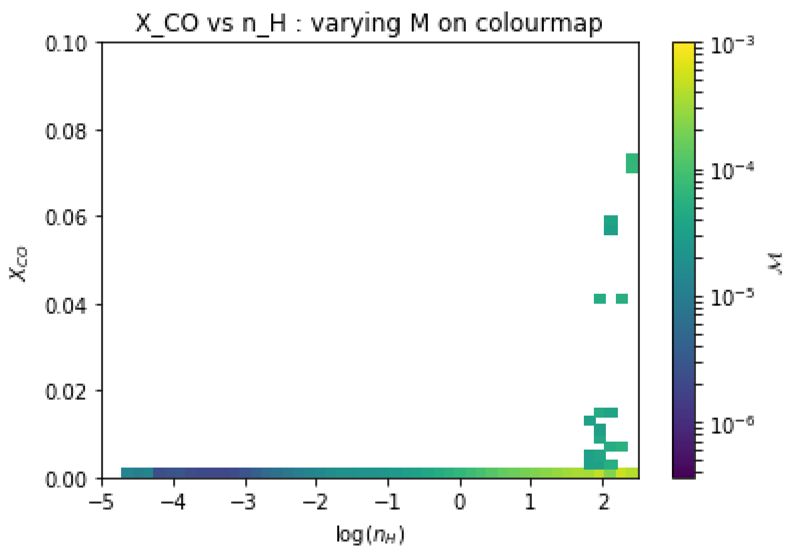
In [86]:

```
histX_CO_M_mass, yedges, xedges = np.histogram2d(region.gas["X_CO"],
                                  np.log10(region.gas["n_H"].in_units("cm**-3")),
                                  weights=region.gas["mach"] * region.gas["mass"],
bins=50 , range=[[0,0.1],[-5,2.5]])

histX_CO_mass, yedges, xedges = np.histogram2d(region.gas["X_CO"],
                                  np.log10(region.gas["n_H"].in_units("cm**-3")),
                                  weights=region.gas["mass"], bins=50 , range=[[0,
0.1],[-5,2.5]])

yX_CO_M, xX_CO_M = yedges, xedges
```

In [87]:

```
#M - mass weighted - X_CO vs n_H
plt.pcolormesh(xedges, yedges, histX_CO_M_mass/histX_CO_mass, norm=LogNorm(), vm
in=3.6e-7, vmax=1e-3)

plt.xlabel(r"$\log(n_H)$")
plt.ylabel(r"$X_{CO}$")
plt.colorbar(label=r"$\mathcal{M}$")
plt.title("X_CO vs n_H : varying M on colourmap")
#plt.savefig('003_log(n_H)vsX_CO--M.png')

plt.show()
```

```
/net/cephfs/home/mkrets/anaconda2/lib/python2.7/site-packages/ipyker
nel_launcher.py:2: RuntimeWarning: invalid value encountered in divi
de
```



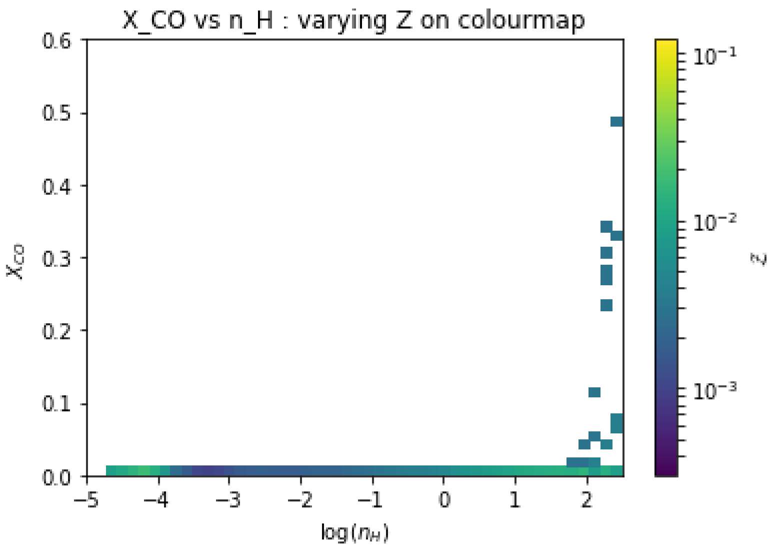
In [88]:

```
histX_CO_mass_Z, yedges, xedges = np.histogram2d(region.gas["X_CO"],
                               np.log10(region.gas["n_H"].in_units("cm**-3")),
                               weights=region.gas["metal"] * region.gas["mass"],
bins=50 , range=[[0,0.6],[-5,2.5]])

histX_CO_mass, yedges, xedges = np.histogram2d(region.gas["X_CO"],
                               np.log10(region.gas["n_H"].in_units("cm**-3")),
                               weights=region.gas["mass"], bins=50 , range=[[0,
0.6],[-5,2.5]])

yX_CO_Z, xX_CO_Z = yedges, xedges
```

In [89]:

```
#Z - mass weighted - X_CO vs n_H
plt.pcolormesh(xedges, yedges, histX_CO_mass_Z/histX_CO_mass, norm=LogNorm(), vm
in=3.03e-4, vmax=1.2e-1)

plt.xlabel(r"$\log(n_H)$")
plt.ylabel(r"$X_{CO}$")
plt.colorbar(label=r"$\mathcal{Z}$")
plt.title("X_CO vs n_H : varying Z on colourmap")
#plt.savefig('004_log(n_H)vsX_CO--Z.png')

plt.show()
```

```
/net/cephfs/home/mkrets/anaconda2/lib/python2.7/site-packages/ipyker
nel_launcher.py:2: RuntimeWarning: invalid value encountered in divi
de
```



In [90]:

```
np.min(Z)
```

Out[90]:

```
SimArray(0.000267)
```

In [91]:

```
np.max(Z)
```

Out[91]:

```
SimArray(0.11394659)
```

In [92]:

```
region.properties
```

Out[92]:

```
{'a': 0.600005205267563,
 'boxsize': Unit("2.21e+04 kpc"),
 'h': 0.6773999786376951,
 'omegaL0': 0.691100001335144,
 'omegaM0': 0.308899998664856,
 'time': Unit("7.68e+00 kpc s km**-1")}
```

In [93]:

```
region.gas.all_keys()
```

Out[93]:

```
['tform',
 'mass',
 'temp',
 'r_mag',
 'i_mag',
 'K_lum_den',
 'vtheta',
 'U_lum_den',
 'vcxy',
 'j2',
 'I_lum_den',
 'u_mag',
 'vr',
 'vt',
 'H_lum_den',
 'V_lum_den',
 'alt',
 'i_lum_den',
 'zeldovich_offset',
 'az',
 'I_mag',
 'vrxy',
 'u_lum_den',
 'J_lum_den',
 'k_mag',
 'v_mag',
 'U_mag',
 'v_mean',
 'h_lum_den',
 'v_lum_den',
 'theta',
 'b_mag',
 'j_mag',
 'B_lum_den',
 'jz',
 'j_lum_den',
 'K_mag',
 'V_mag',
 'v2',
 'J_mag',
 'rho',
 'H_mag',
 'h_mag',
 'B_mag',
 'aform',
 'k_lum_den',
 'te',
 'b_lum_den',
 'ke',
 'age',
 'j',
 'smooth',
 'rxy',
 'r',
 'R_lum_den',
 'vphi',
 'r_lum_den',
 'tform',
 'R_mag',
```

```
 'v_disp',
 'mask',
 'smooth',
 'metal',
 'pos',
 'p',
 'rho',
 'vel',
 'turb']
```

In [94]:

```
np.shape(M)
```

Out[94]:

```
(1291634,)
```

In [95]:

```
np.shape(Z)
```

Out[95]:

```
(1291634,)
```

In [96]:

```
np.shape(n_H)
```

Out[96]:

```
(1291634,)
```

In [97]:

```
np.shape(rho)
```

Out[97]:

```
(1291634,)
```

In [98]:

```
np.shape(X_H2)
```

Out[98]:

```
(1291634,)
```

In [99]:

```
len(n_H)
```

Out[99]:

```
1291634
```

In [100]:

```
region.gas.loadable_keys()
```

Out[100]:

```
['mask', 'smooth', 'metal', 'pos', 'p', 'rho', 'vel', 'turb']
```

In [101]:

```
c_s
```

Out[101]:

```
SimArray([ 3276673.11939746, 12738195.66072513,  9202021.83758316,
...,
          14469186.19848063, 13460450.29314729, 14119970.07822348],
'cm s**-1')
```