Task 1

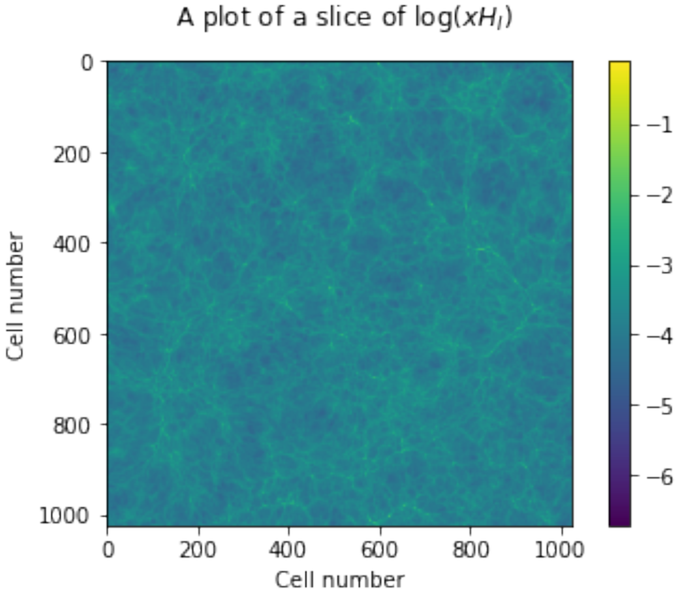
In [1]:

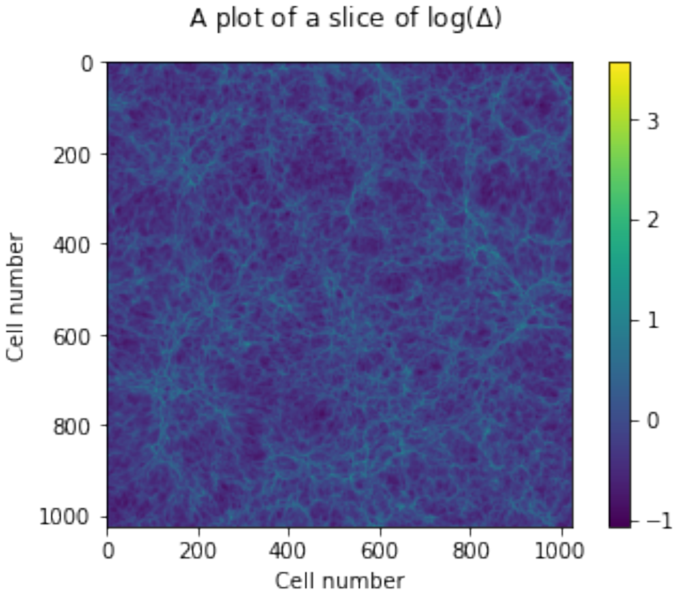
```
import array
import numpy as np
import matplotlib.pyplot as plt


binfpath="/fs/lustre/cita/hqchen/data/z-ifrit-a=0.1401.bin"
with open(binfpath,'rb') as f:
    binarydata=f.read()
lllen=8; flen=4; ilen=4

N=1024

firstfield=lllen*2+3*ilen

nextfield=firstfield

xHI=array.array('f',binarydata[nextfield+lllen:nextfield+lllen+N**3*flen])
xHI=np.reshape(xHI,(N,N,N),order='F')

nextfield+=lllen+N**3*flen
delta=array.array('f',binarydata[nextfield+lllen:nextfield+lllen+N**3*flen]
delta=np.reshape(delta,(N,N,N),order='F')

nextfield+=lllen+N**3*flen
T=array.array('f',binarydata[nextfield+lllen:nextfield+lllen+N**3*flen])
T=np.reshape(T,(N,N,N),order='F')
```

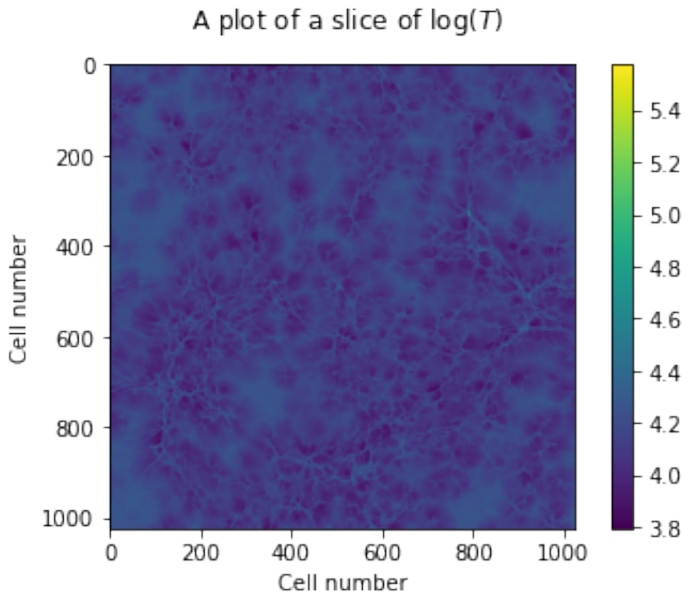
In [2]:

```
fig1, ax1 = plt.subplots()
im1 = ax1.imshow(np.log10(xHI[0]))
ax1.set(xlabel = "Cell number", ylabel = "Cell number")
fig1.colorbar(im1)
fig1.suptitle("A plot of a slice of $\log(xH_I)$")

fig2, ax2 = plt.subplots()
im2  = ax2.imshow(np.log10(delta[0]))
fig2.colorbar(im2)
ax2.set(xlabel = "Cell number", ylabel = "Cell number")
fig2.suptitle("A plot of a slice of $\log(\Delta)$")

fig3, ax3 = plt.subplots()
im3 = ax3.imshow(np.log10(T[0]))
fig3.colorbar(im3)
ax3.set(xlabel = "Cell number", ylabel = "Cell number")
fig3.suptitle("A plot of a slice of $\log(T)$")
```

```
/tmp/ipykernel_2004116/591149048.py:8: RuntimeWarning: divide by zero encoun
tered in log10
  im2  = ax2.imshow(np.log10(delta[0]))
/tmp/ipykernel_2004116/591149048.py:14: RuntimeWarning: divide by zero encou
ntered in log10
  im3 = ax3.imshow(np.log10(T[0]))
Out[2]: Text(0.5, 0.98, 'A plot of a slice of $\\log(T)$')
```







Task 2

In [3]:

```
import astropy
```

In [4]:

```
from astropy.cosmology import WMAP9 as cosmo
from astropy import units as u
crt_dns = cosmo.critical_density(6)
mean_gas_density = crt_dns*cosmo.Ob(6)
print(mean_gas_density)
```

```
1.432778841086186e-28 g / cm3
```

Task 3

In [5]:

```
from astropy.constants import m_p
p = m_p.to(u.g)
```

In [ ]:

In [6]:

```
sigma  = 4.48e-18
nH =  (0.76*mean_gas_density/p)
nHI = xHI*nH
nHI = nHI.astype('float128')
cml = (40*0.68)/1024*u.Mpc
ds = cml*(1/(1+6))
ds = ds.to(u.cm)
dT = sigma * nHI * ds
print(dT.max())
```

```
2.910085329003365 1 / cm2
```

In [7]:

```
print(nHI.max())
```

```
5.547614273382351e-05 1 / cm3
```

Task 4

In [8]:

```
from numpy import float128
print(nHI)
```

```
[[[6.74571377e-09 6.38827302e-09 5.75865577e-09 ... 1.07600648e-08
   8.21360135e-09 7.04414260e-09]
  [7.67662289e-09 6.92509960e-09 6.13529849e-09 ... 1.13985843e-08
   9.96054794e-09 8.55018278e-09]
  [9.01550923e-09 7.51253726e-09 6.50632348e-09 ... 1.14072840e-08
   1.12177583e-08 1.05493712e-08]
  ...
  [5.40711653e-09 5.72221870e-09 6.19928953e-09 ... 6.03759931e-09
   5.55259971e-09 5.45984546e-09]
  [5.71314507e-09 5.72291681e-09 5.85148463e-09 ... 7.31123873e-09
   6.16871665e-09 5.86614579e-09]
  [6.06233908e-09 5.96980865e-09 5.72571146e-09 ... 9.11899800e-09
   6.97984026e-09 6.31654506e-09]]

 [[5.48221868e-09 5.28250510e-09 5.17780352e-09 ... 8.64833005e-09
   7.15157933e-09 6.03492856e-09]
  [6.61523991e-09 5.69851810e-09 5.04244824e-09 ... 9.86796511e-09
   9.04403130e-09 7.89581023e-09]
  [8.48688053e-09 6.67445255e-09 5.43424594e-09 ... 9.99739669e-09
   1.04928599e-08 1.01232711e-08]
  ...
  [5.35914690e-09 5.90746518e-09 6.73172940e-09 ... 5.94980287e-09
   5.42529799e-09 5.36209166e-09]
  [5.29865085e-09 5.49629009e-09 6.00415806e-09 ... 6.46318110e-09
   5.62216762e-09 5.38439249e-09]
  [5.24459631e-09 5.30382804e-09 5.45351719e-09 ... 7.39922790e-09
   6.04202244e-09 5.43723200e-09]]
```

```
[[5.65969582e-09 5.40447687e-09 5.32718669e-09 ... 8.78586093e-09
  7.66012498e-09 6.51277920e-09]
 [6.74983269e-09 5.43955547e-09 4.94069496e-09 ... 1.01115383e-08
  9.74585923e-09 8.50905479e-09]
 [8.27511659e-09 5.94726091e-09 4.65099470e-09 ... 1.02272972e-08
  1.10958682e-08 1.05951843e-08]
 ...
 [5.74321213e-09 6.75065648e-09 8.07553402e-09 ... 6.12818774e-09
  5.51611823e-09 5.61805180e-09]
 [5.43424594e-09 5.91756999e-09 6.90652691e-09 ... 6.49798881e-09
  5.70652325e-09 5.57161073e-09]
 [5.37224709e-09 5.50367529e-09 5.88012572e-09 ... 7.37938377e-09
  6.22773211e-09 5.55938140e-09]]

...

[[1.19268533e-08 1.15134302e-08 1.07251497e-08 ... 1.40352689e-08
  1.33990881e-08 1.25596511e-08]
 [1.25206165e-08 1.14769465e-08 1.03723554e-08 ... 1.24436701e-08
  1.33240592e-08 1.31913547e-08]
 [1.21977752e-08 1.05859739e-08 8.71948558e-09 ... 1.04326725e-08
  1.20314079e-08 1.29868418e-08]
 ...
 [6.00654060e-09 6.70937395e-09 7.05845338e-09 ... 7.74368214e-09
  5.96088867e-09 5.85237769e-09]
 [7.55000151e-09 7.81013476e-09 7.81824294e-09 ... 1.09410241e-08
  8.33111624e-09 7.68858133e-09]
 [9.84165016e-09 9.77997416e-09 9.55050616e-09 ... 1.34060398e-08
  1.13394298e-08 1.01863371e-08]]

[[1.24550663e-08 1.11333485e-08 9.92762139e-09 ... 1.59916471e-08
  1.53827706e-08 1.38836107e-08]
 [1.25435635e-08 1.07303881e-08 9.55896251e-09 ... 1.55223567e-08
  1.58943401e-08 1.49971093e-08]
 [1.19662289e-08 9.15804232e-09 7.71655273e-09 ... 1.23242456e-08
  1.45179655e-08 1.47393902e-08]
 ...
 [5.99372241e-09 6.77149625e-09 7.40306660e-09 ... 6.24009866e-09
  5.38948841e-09 5.57586333e-09]
 [7.96806532e-09 8.31029379e-09 8.14707146e-09 ... 9.95279947e-09
  7.95883093e-09 7.80632181e-09]
 [1.04770210e-08 1.01515791e-08 9.23015087e-09 ... 1.40485517e-08
  1.17774750e-08 1.08903908e-08]]

[[9.89857885e-09 9.12902198e-09 7.99851918e-09 ... 1.50374380e-08
  1.27527651e-08 1.10960370e-08]
 [1.03382263e-08 8.87411211e-09 7.83998555e-09 ... 1.50668384e-08
  1.42406380e-08 1.26288322e-08]
 [1.03281339e-08 8.21122015e-09 7.27207450e-09 ... 1.27465389e-08
  1.41608973e-08 1.31094859e-08]
 ...
 [5.77009107e-09 6.13660900e-09 6.66895561e-09 ... 6.04331385e-09
  5.52572210e-09 5.72868286e-09]
 [7.17705095e-09 7.16960757e-09 6.77356216e-09 ... 8.78170692e-09
  7.27096516e-09 7.18669479e-09]
 [8.72241124e-09 8.30928037e-09 7.34680095e-09 ... 1.21840085e-08
  9.77877868e-09 8.99050789e-09]]] 1 / cm3
```

In [9]:

```
from numpy import float128

new_tau = dT.max().value
print(new_tau)
percent = float128(10)
percent = float128(np.exp(-new_tau)*100)
print(percent)
```

```
2.9100853290033651996
5.4471081707766847687
```

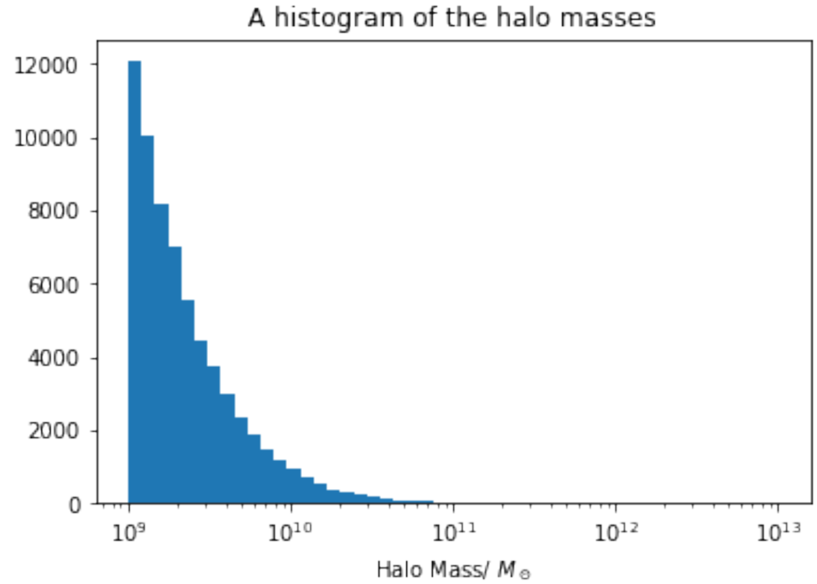
Task 5

In [10]:

```
halofpath="/fs/lustre/cita/hqchen/data/hlist_0.14005.list"
halodata = np.loadtxt(halofpath, skiprows = 63)

halomass = halodata[:, 10]





#mass = []
#for i in range(len(halodata)):
#    mass = np.append(mass, float128(halodata[i][83:94]))
```

In [11]:

```
plt.hist(halomass, bins = np.logspace(start =np.log10(10**9), stop = np.log
plt.gca().set_xscale("log")
plt.title("A histogram of the halo masses")
plt.xlabel("Halo Mass/ $M_\odot$")
plt.show()
```


A histogram of the halo masses
12000
10000
8000
6000
4000
2000
0
10^9
10^{10}
10^{11}
10^{12}
10^{13}
Halo Mass/ $M_\odot$

Task 6

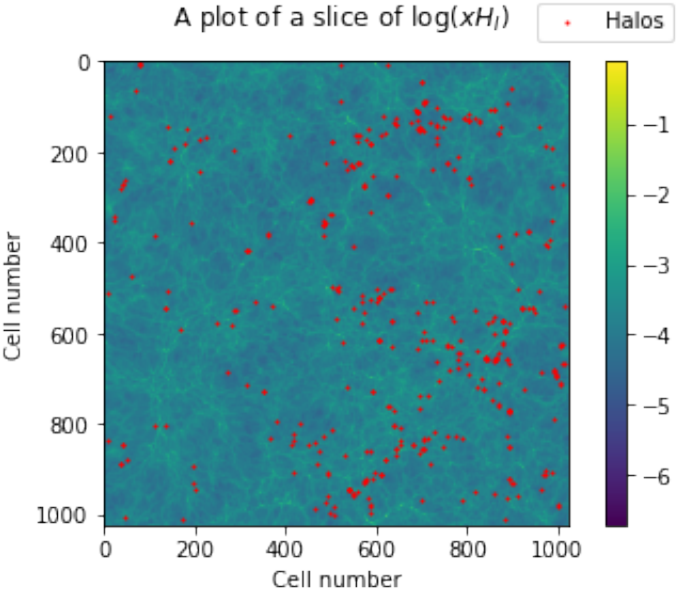
In [52]:

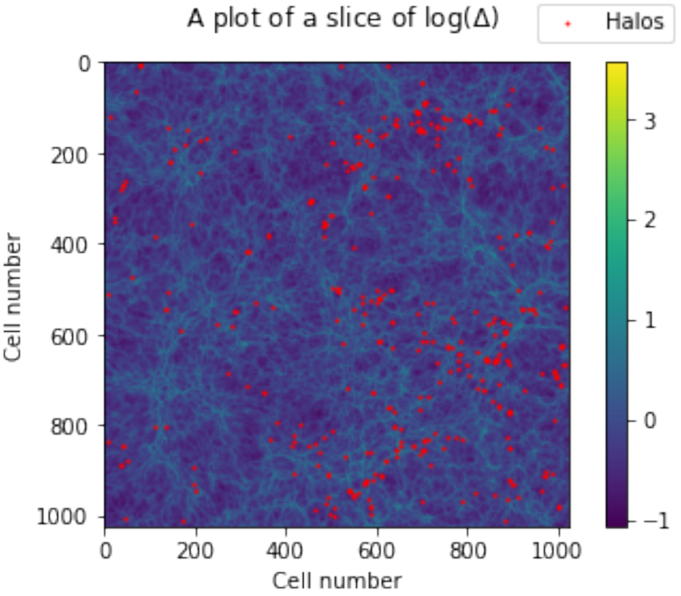
```
x, y, z = halodata[:, 17], halodata[:, 18], halodata[:, 19]
m = 1024*0.989391/40
x, y, z = m*x, m*y, m*z




x_int = x.astype(int)
indices = np.where(x_int ==0)
y_int = y[indices]
z_int = z[indices]


fig1, ax1 = plt.subplots()
im1 = ax1.imshow(np.log10(xHI[0]))
ax1.set(xlabel = "Cell number", ylabel = "Cell number")
ax1.scatter(y_int, z_int, color='red', s=1, label= 'Halos')
ax1.legend(loc = 'upper right', bbox_to_anchor = (1.25, 1.15))

fig1.colorbar(im1)
fig1.suptitle("A plot of a slice of $\log(xH_I)$")

fig2, ax2 = plt.subplots()
im2  = ax2.imshow(np.log10(delta[0]))
ax2.scatter(y_int, z_int, color='red', s=1, label= 'Halos')
ax2.legend(loc = 'upper right', bbox_to_anchor = (1.25, 1.15))
fig2.colorbar(im2)
ax2.set(xlabel = "Cell number", ylabel = "Cell number")
fig2.suptitle("A plot of a slice of $\log(\Delta)$")

fig3, ax3 = plt.subplots()
im3 = ax3.imshow(np.log10(T[0]))
ax3.scatter(y_int, z_int, color='red', s=1, label= 'Halos')
ax3.legend(loc = 'upper right', bbox_to_anchor = (1.25, 1.15))
fig3.colorbar(im3)
ax3.set(xlabel = "Cell number", ylabel = "Cell number")
fig3.suptitle("A plot of a slice of $\log(T)$")
```

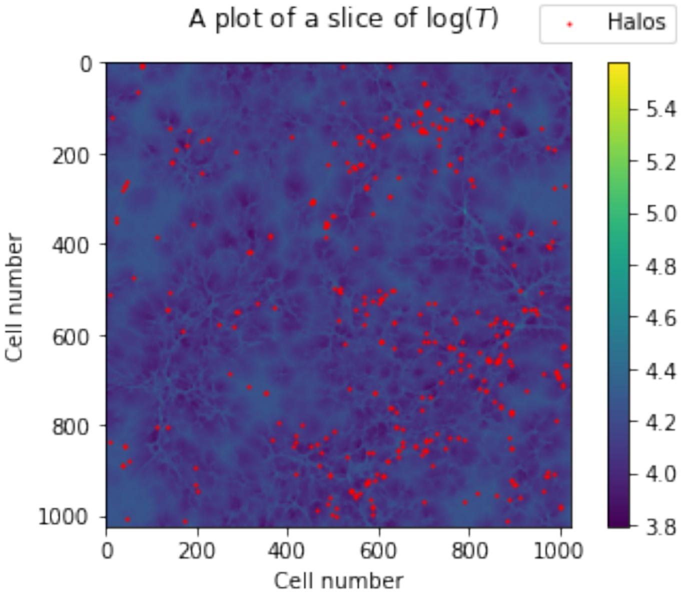
```
/tmp/ipykernel_2004116/3369886743.py:24: RuntimeWarning: divide by zero enco
untered in log10
  im2  = ax2.imshow(np.log10(delta[0]))
/tmp/ipykernel_2004116/3369886743.py:32: RuntimeWarning: divide by zero enco
untered in log10
  im3 = ax3.imshow(np.log10(T[0]))
```

Out[52]:

```
Text(0.5, 0.98, 'A plot of a slice of $\\log(T)$')
```


A plot of a slice of log(xH_I)
Halos
Cell number
Cell number
0
200
400
600
800
1000
0
200
400
600
800
1000
−1
−2
−3
−4
−5
−6


A plot of a slice of log(Δ)
Halos
Cell number
Cell number
0
200
400
600
800
1000
0
200
400
600
800
1000
3
2
1
0
−1


A plot of a slice of log(T)
Halos
Cell number
Cell number
0
200
400
600
800
1000
0
200
400
600
800
1000
5.4
5.2
5.0
4.8
4.6
4.4
4.2
4.0
3.8

Task 7

```
In [ ]:
```