Part B: Simulation Analysis (Due November 21st; 50 points)

Bin Xia

---

1. (20 points) Finding Dark Matter Halos and Computing their Mass Function

(a) Run the halo finder for your datasets at z = 0, 1, 2, 3.

In [ ]:
```python
import yt
from yt.extensions.astro_analysis.halo_analysis import HaloCatalog
import numpy as np
import matplotlib.pyplot as plt
# yt.set_log_level(40)
```

In [ ]:
```python
ts = yt.load("RD????/RedshiftOutput????")
idz_array = [-1, 8, 6, 5]
ds = {}
for i, idz in enumerate(idz_array):
    ds[i] = ts[idz]
    # ds[z] = yt.load(f"RD{z:04d}/RedshiftOutput{z:04d}")
    hc = HaloCatalog(data_ds=ds[i], finder_method="hop")
    hc.create()
```

```
yt : [INFO     ] 2023-11-20 17:45:09,205 Parameters: current_time              = 650.91854077973
yt : [INFO     ] 2023-11-20 17:45:09,206 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:09,206 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:09,207 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:09,208 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:09,208 Parameters: current_redshift          = -3.3306690738755e-16
yt : [INFO     ] 2023-11-20 17:45:09,208 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:09,209 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:09,209 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:09,209 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:09,212 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:09,431 Initializing HOP
Calling hop... 262144 1.600e+02
yt : [INFO     ] 2023-11-20 17:45:12,404 Parsing outputs
nSmooth = 65 kd->nActive = 262144
Calling regroup...
No minimum group size specified.  Assuming 10 particles.
ngroups = 699
Copying arrays for 262144 particles
Building Tree...
Finding Densities...
Finding Densest Neighbors...
Grouping...
Merging Groups...
Writing Output...
All Done!
yt : [INFO     ] 2023-11-20 17:45:15,484 Saving 92 halos: halo_catalogs/RedshiftOutput0014/RedshiftOutput0014.
0.h5.
yt : [INFO     ] 2023-11-20 17:45:15,591 Parameters: current_time              = 276.68070327028
yt : [INFO     ] 2023-11-20 17:45:15,591 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:15,592 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:15,592 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:15,593 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:15,593 Parameters: current_redshift          = 0.99999995433926
yt : [INFO     ] 2023-11-20 17:45:15,593 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:15,594 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:15,595 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:15,595 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:15,597 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:15,788 Initializing HOP
Calling hop... 262144 1.600e+02
yt : [INFO     ] 2023-11-20 17:45:18,622 Parsing outputs
nSmooth = 65 kd->nActive = 262144
Calling regroup...
No minimum group size specified.  Assuming 10 particles.
ngroups = 690
```

```
Copying arrays for 262144 particles
Building Tree...
Finding Densities...
Finding Densest Neighbors...
Grouping...
Merging Groups...
Writing Output...
All Done!
```

```
yt : [INFO     ] 2023-11-20 17:45:20,423 Saving 68 halos: halo_catalogs/RedshiftOutput0008/RedshiftOutput0008.0.h5.
yt : [INFO     ] 2023-11-20 17:45:20,528 Parameters: current_time              = 155.04474940309
yt : [INFO     ] 2023-11-20 17:45:20,528 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:20,529 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:20,529 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:20,530 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:20,530 Parameters: current_redshift          = 1.999998301463
yt : [INFO     ] 2023-11-20 17:45:20,530 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:20,531 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:20,531 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:20,531 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:20,533 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:20,722 Initializing HOP
yt : [INFO     ] 2023-11-20 17:45:23,581 Parsing outputs
```

```
Copying arrays for 262144 particles
Building Tree...
Finding Densities...
Finding Densest Neighbors...
Grouping...
Merging Groups...
Writing Output...
All Done!
```

```
Calling hop... 262144 1.600e+02
nSmooth = 65 kd->nActive = 262144
Calling regroup...
No minimum group size specified.  Assuming 10 particles.
ngroups = 634
yt : [INFO     ] 2023-11-20 17:45:24,582 Saving 43 halos: halo_catalogs/RedshiftOutput0006/RedshiftOutput0006.0.h5.
yt : [INFO     ] 2023-11-20 17:45:24,685 Parameters: current_time              = 101.47857431987
yt : [INFO     ] 2023-11-20 17:45:24,686 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:24,686 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:24,686 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:24,687 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:24,688 Parameters: current_redshift          = 2.9999997678668
yt : [INFO     ] 2023-11-20 17:45:24,688 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:24,688 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:24,689 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:24,689 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:24,690 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:24,875 Initializing HOP
Calling hop... 262144 1.600e+02
yt : [INFO     ] 2023-11-20 17:45:27,780 Parsing outputs
nSmooth = 65 kd->nActive = 262144
Calling regroup...
No minimum group size specified.  Assuming 10 particles.
ngroups = 588
```

```
Copying arrays for 262144 particles
Building Tree...
Finding Densities...
Finding Densest Neighbors...
Grouping...
Merging Groups...
Writing Output...
All Done!
```

```
yt : [INFO     ] 2023-11-20 17:45:28,268 Saving 26 halos: halo_catalogs/RedshiftOutput0005/RedshiftOutput0005.0.h5.
```

(b) Overplot circles corresponding to those halos on the x-axis dark matter projection at z = 0.

In [ ]:

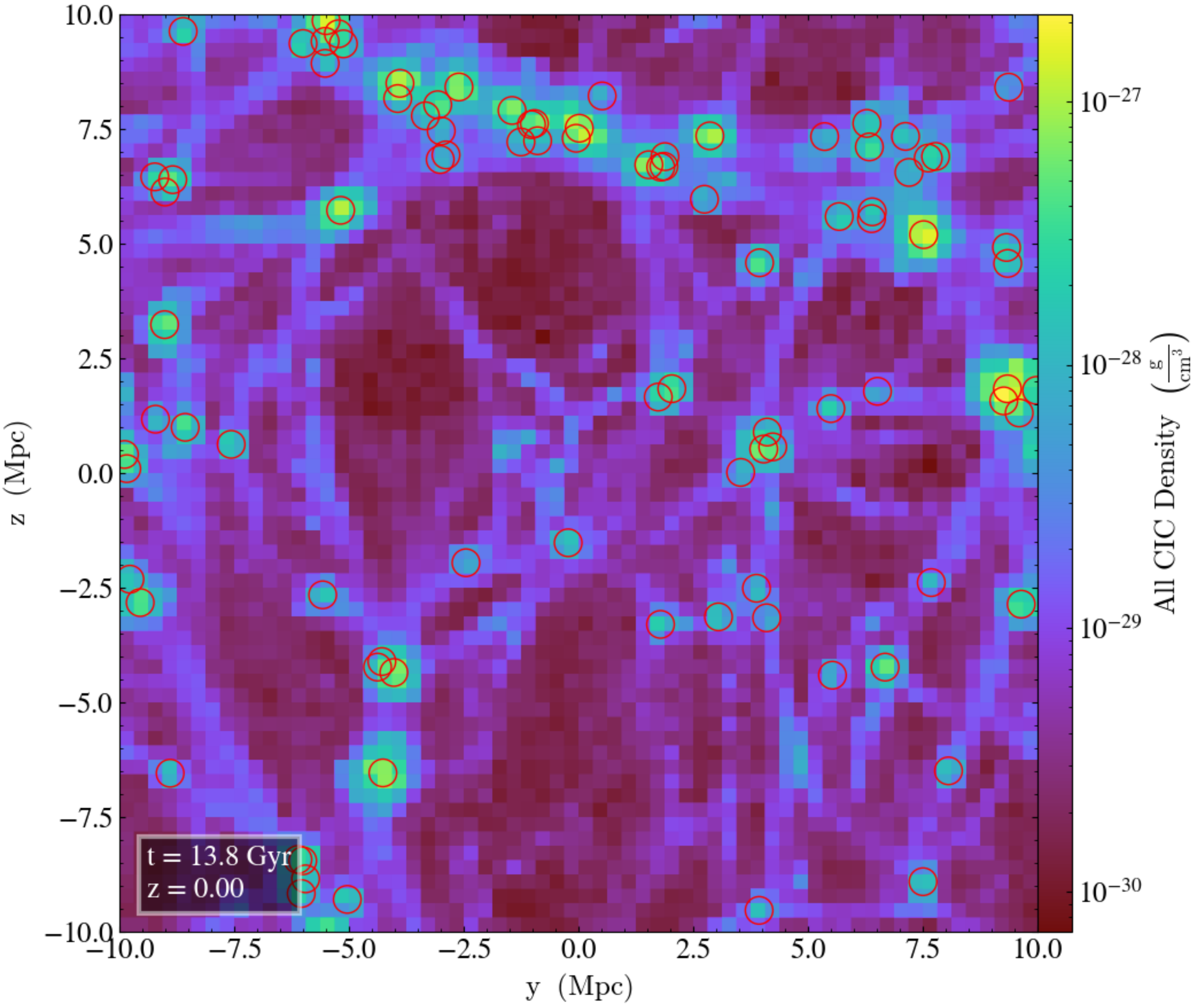
```python
hds_ts = yt.load("halo_catalogs/RedshiftOutput00??/RedshiftOutput00??.0.h5")
halos = {}
masses = {}
positions = {}
for z in range(4):
    halos[z] = hds_ts[-1-z].all_data()
    masses[z] = halos[z]["all", "particle_mass"].to("Msun")
    positions[z] = halos[z]["all", "particle_position"]
```

```
yt : [INFO     ] 2023-11-20 17:45:28,409 Parameters: current_time              = 650.91854077973 code_time
yt : [INFO     ] 2023-11-20 17:45:28,409 Parameters: domain_dimensions         = [1 1 1]
yt : [INFO     ] 2023-11-20 17:45:28,410 Parameters: domain_left_edge          = [0. 0. 0.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,410 Parameters: domain_right_edge         = [1. 1. 1.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,411 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:28,412 Parameters: current_redshift          = -3.3306690738755e-16
yt : [INFO     ] 2023-11-20 17:45:28,412 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:28,413 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:28,413 Parameters: omega_radiation           = 0.0
yt : [INFO     ] 2023-11-20 17:45:28,414 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:28,409 Parameters: domain_dimensions         = [1 1 1]
yt : [INFO     ] 2023-11-20 17:45:28,410 Parameters: domain_left_edge          = [0. 0. 0.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,410 Parameters: domain_right_edge         = [1. 1. 1.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,411 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:28,412 Parameters: current_redshift          = -3.3306690738755e-16
yt : [INFO     ] 2023-11-20 17:45:28,412 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:28,413 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:28,413 Parameters: omega_radiation           = 0.0
yt : [INFO     ] 2023-11-20 17:45:28,414 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:28,495 Allocating for 92 particles
yt : [INFO     ] 2023-11-20 17:45:28,723 Parameters: current_time              = 276.68070327028 code_time
yt : [INFO     ] 2023-11-20 17:45:28,723 Parameters: domain_dimensions         = [1 1 1]
yt : [INFO     ] 2023-11-20 17:45:28,724 Parameters: domain_left_edge          = [0. 0. 0.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,724 Parameters: domain_right_edge         = [1. 1. 1.] code_length
yt : [INFO     ] 2023-11-20 17:45:28,725 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:28,725 Parameters: current_redshift          = 0.99999995433926
yt : [INFO     ] 2023-11-20 17:45:28,725 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:28,726 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:28,726 Parameters: omega_radiation           = 0.0
yt : [INFO     ] 2023-11-20 17:45:28,726 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:28,809 Allocating for 68 particles
yt : [INFO     ] 2023-11-20 17:45:29,041 Parameters: current_time              = 155.04474940309 code_time
yt : [INFO     ] 2023-11-20 17:45:29,041 Parameters: domain_dimensions         = [1 1 1]
yt : [INFO     ] 2023-11-20 17:45:29,042 Parameters: domain_left_edge          = [0. 0. 0.] code_length
yt : [INFO     ] 2023-11-20 17:45:29,042 Parameters: domain_right_edge         = [1. 1. 1.] code_length
yt : [INFO     ] 2023-11-20 17:45:29,043 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:29,044 Parameters: current_redshift          = 1.999998301463
yt : [INFO     ] 2023-11-20 17:45:29,044 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:29,044 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:29,045 Parameters: omega_radiation           = 0.0
yt : [INFO     ] 2023-11-20 17:45:29,045 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:29,126 Allocating for 43 particles
yt : [INFO     ] 2023-11-20 17:45:29,356 Parameters: current_time              = 101.47857431987 code_time
yt : [INFO     ] 2023-11-20 17:45:29,357 Parameters: domain_dimensions         = [1 1 1]
yt : [INFO     ] 2023-11-20 17:45:29,357 Parameters: domain_left_edge          = [0. 0. 0.] code_length
yt : [INFO     ] 2023-11-20 17:45:29,358 Parameters: domain_right_edge         = [1. 1. 1.] code_length
yt : [INFO     ] 2023-11-20 17:45:29,358 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:29,359 Parameters: current_redshift          = 2.9999997678668
yt : [INFO     ] 2023-11-20 17:45:29,359 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:29,360 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:29,360 Parameters: omega_radiation           = 0.0
yt : [INFO     ] 2023-11-20 17:45:29,361 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:29,441 Allocating for 26 particles
```

In [ ]:

```
field = ("deposit", "all_cic")
redshift = 0
p = yt.ProjectionPlot(ds[redshift], "x", fields=field, weight_field=field)
for i in range(positions[redshift].shape[0]):
    p.annotate_sphere(positions[redshift][i], radius=(0.3, "Mpc"), circle_args={"color":"red"})
p.annotate_timestamp(redshift=True, draw_inset_box=True)
p.show()
```

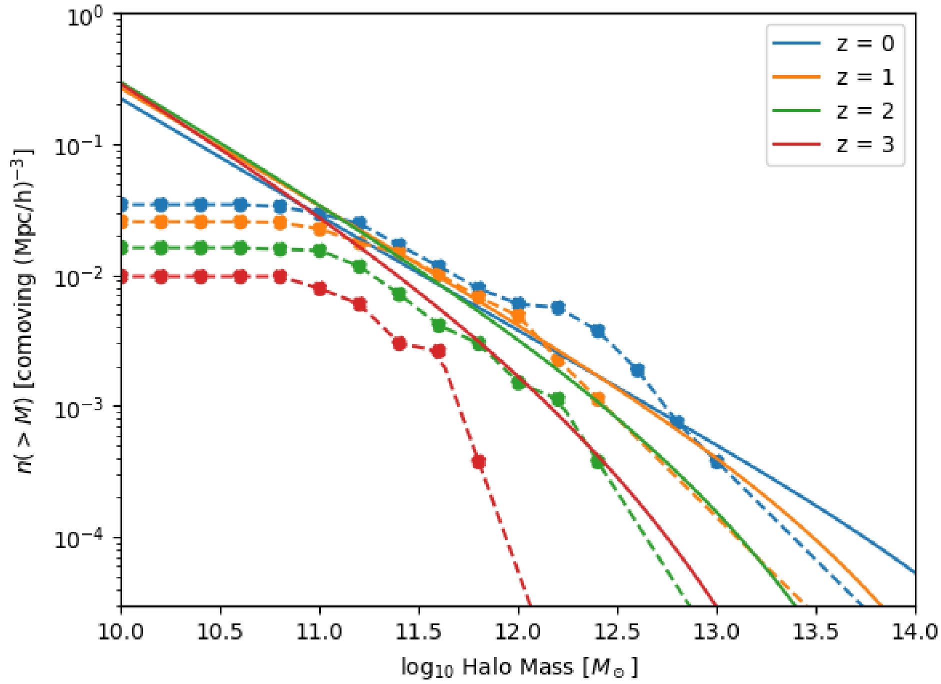
```
yt : [INFO     ] 2023-11-20 17:45:29,757 Projection completed
yt : [INFO     ] 2023-11-20 17:45:29,759 xlim = 0.000000 1.000000
yt : [INFO     ] 2023-11-20 17:45:29,759 ylim = 0.000000 1.000000
yt : [INFO     ] 2023-11-20 17:45:29,761 xlim = 0.000000 1.000000
yt : [INFO     ] 2023-11-20 17:45:29,761 ylim = 0.000000 1.000000
yt : [INFO     ] 2023-11-20 17:45:29,763 Making a fixed resolution buffer of (('deposit', 'all_cic')) 800 by 8
00
```



(c) Compute the halo mass function n(< M) at z = 0, 1, 2, 3 and compare this with the mass function calculated from Press-Schetcher formalism that considers an ellipsoidal collapse model.

In [ ]:
```
from astropy.cosmology import WMAP9 as cosmo
from scipy.interpolate import interp1d

littleh = cosmo.H(0).value/100

n = {}
fn = {}
logM_array = {}
for z in range(4):
    logM = np.log10(masses[z])
    logM_array[z] = np.arange(0, logM.max(), 0.2)
    n[z] = np.array([np.sum(logM>lm) for lm in logM_array[z]], dtype=np.float64) # count number of halo with m
    n[z] /= (20*littleh)**3 # convert from number to number density
    fn[z] = interp1d(logM_array[z], np.log10(n[z]), kind="linear", fill_value="extrapolate")
```

In [ ]:
```
data = {}
for z in range(4):
    data[z] = np.loadtxt(f"mass-fn-z{z}.dat")
    plt.plot(data[z].T[0], data[z].T[1], label=f"z = {z}", c=f"C{z}", linestyle="-")
    # plt.plot(logM_array[z], fn[z](logM_array[z]), c=f"C{z}", linestyle="--")
    # plt.scatter(logM_array[z], 10**fn[z](logM_array[z]), c=f"C{z}", linestyle="--")
    plt.scatter(logM_array[z], n[z], c=f"C{z}", linestyle="--")
    plt.plot(data[z].T[0], 10**fn[z](data[z].T[0]), c=f"C{z}", linestyle="--")
plt.xlabel("log$_{10}$ Halo Mass $[M_\odot]$")
plt.ylabel("$n(>M)$ [comoving (Mpc/h)$^{-3}$]")
plt.xlim(10,14)
plt.ylim(3e-5, 1)
plt.legend()
plt.yscale("log")
```



Caption: redshift = [0,1,2,3] correpond to curves in blue, orange, green, and red, respectively. Solid curves are from Press-Schetcher formalism, while dashed curves are interpolated from my numerical halo (big dots in color).

From the figure, we can see numerical halos tend to have smaller density for $log_{10}(M/M_{\odot}) < 11$ than that of Press-Schetcher formalism. I think this is because HOP method didn't count the halos with low mass.

For $log_{10}(M/M_{\odot}) > 11$, the halo mass functions from the numerical halos tend to be power-law shape, though they are still lower than the solid curves.

2. (20 points) Calculating the Two-Point Correlation Function of Halos

In [ ]:
```
import treecorr
```

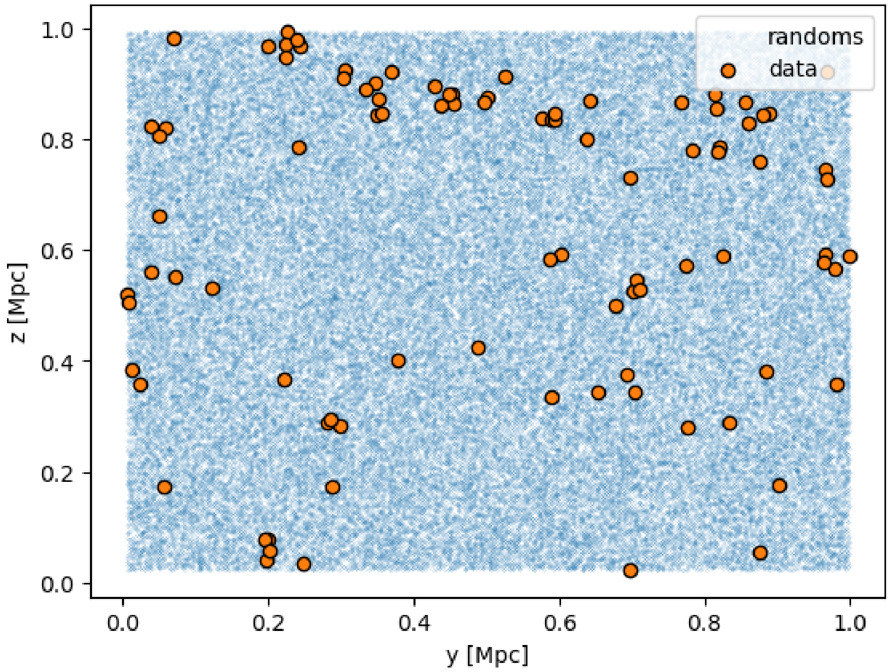
In [ ]:
```
print(positions[0].T.shape)
print(positions[0].T.max())
print(positions[0].T.min())
print(positions[0].T.units)
# print(positions[0].T[0])
```

```
(3, 92)
0.9989284252932232 code_length
0.005510675127156528 code_length
code_length
```

In [ ]:
```
min_sep=0.01; max_sep=1; nbins=20
# Calculate count pairs for data-data pairs
cat = treecorr.Catalog(x=positions[0].T[0], y=positions[0].T[1], z=positions[0].T[2])
dd = treecorr.NNCorrelation(min_sep=min_sep, max_sep=max_sep, nbins=nbins)
dd.process(cat)
print(dd)
```

```
NNCorrelation(config={'min_sep': 0.01, 'max_sep': 1.0, 'nbins': 20, 'brute': False, 'verbose': 1, 'split_metho
d': 'mean', 'max_top': 10, 'precision': 4, 'pairwise': False, 'm2_uform': 'Crittenden', 'metric': 'Euclidean',
'bin_type': 'Log', 'var_method': 'shot', 'num_bootstrap': 500})
```

In [ ]:
```
rand_x = np.random.uniform(cat.x.min(), cat.x.max(), 10**5) * positions[0].T.units
rand_y = np.random.uniform(cat.y.min(), cat.y.max(), 10**5) * positions[0].T.units
rand_z = np.random.uniform(cat.z.min(), cat.z.max(), 10**5) * positions[0].T.units
plt.scatter(rand_y, rand_z, s=0.01, label="randoms")
plt.scatter(positions[0].T[1], positions[0].T[2], label="data", edgecolors="k")
plt.legend()
plt.xlabel("y [Mpc]")
plt.ylabel("z [Mpc]")
plt.show()
```



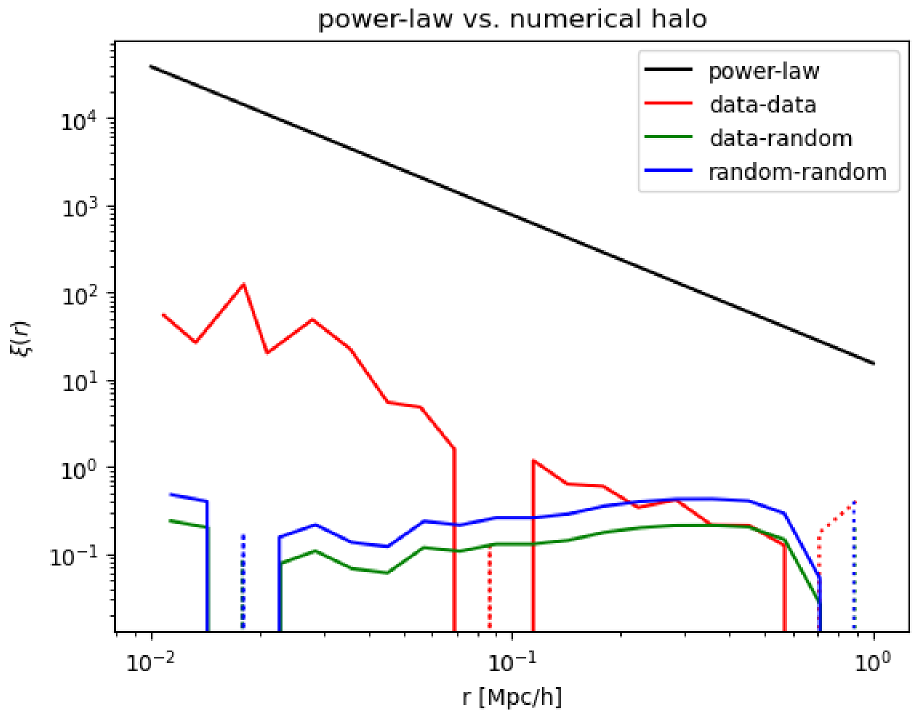
In [ ]:
```
# Calculate count pairs for the random-random pairs.
rand = treecorr.Catalog(x=rand_x, y=rand_y, z=rand_z)
rr = treecorr.NNCorrelation(min_sep=min_sep, max_sep=max_sep, nbins=nbins)
rr.process(rand)
print(rr)
```

```
NNCorrelation(config={'min_sep': 0.01, 'max_sep': 1.0, 'nbins': 20, 'brute': False, 'verbose': 1, 'split_metho
d': 'mean', 'max_top': 10, 'precision': 4, 'pairwise': False, 'm2_uform': 'Crittenden', 'metric': 'Euclidean',
'bin_type': 'Log', 'var_method': 'shot', 'num_bootstrap': 500})
```

In [ ]:
```
# Calculate count pairs for the data-random pairs.
dr = treecorr.NNCorrelation(min_sep=min_sep, max_sep=max_sep, nbins=nbins)
dr.process(cat, rand)
print(dr)
```

```
NNCorrelation(config={'min_sep': 0.01, 'max_sep': 1.0, 'nbins': 20, 'brute': False, 'verbose': 1, 'split_metho
d': 'mean', 'max_top': 10, 'precision': 4, 'pairwise': False, 'm2_uform': 'Crittenden', 'metric': 'Euclidean',
'bin_type': 'Log', 'var_method': 'shot', 'num_bootstrap': 500})
```

In [ ]:
```
# plot power-law
r_0 = 5#*littleh
r_array = np.logspace(-2,0,10)
plt.plot(r_array, (r_array/r_0)**(-1.7), label="power-law", c='k')

# plot data-data
xi, varxi = dd.calculateXi(rr=rr, dr=dr)
r = np.exp(dd.meanlogr)
# sig = np.sqrt(varxi)
plt.plot(r, xi, ls="-", label="data-data", c='r')
plt.plot(r, -xi, ls=":", c='r')

# plot data-random
xi, varxi = dr.calculateXi(rr=rr, dr=dr)
r = np.exp(dr.meanlogr)
plt.plot(r, xi, ls="-", label="data-random", c='g')
plt.plot(r, -xi, ls=":", c='g')

# plot random-random
xi, varxi = rr.calculateXi(rr=rr, dr=dr)
r = np.exp(rr.meanlogr)
plt.plot(r, xi, ls="-", label="random-random", c='b')
plt.plot(r, -xi, ls=":", c='b')

plt.xscale("log")
plt.yscale("log")
plt.xlabel("r [Mpc/h]")
plt.ylabel(r"$\xi(r)$")
plt.title("power-law vs. numerical halo")
plt.legend()
plt.show()
```



We can see that the data-data correlation function fits a power-law shape as expected. However, the data-random and random-random curves increase gradually, which makes sense since the *random* galaxies at smaller scales are as clustered as at greater distances.

3. (10 points) Calculating the Evolution of the Matter Power Spectrum

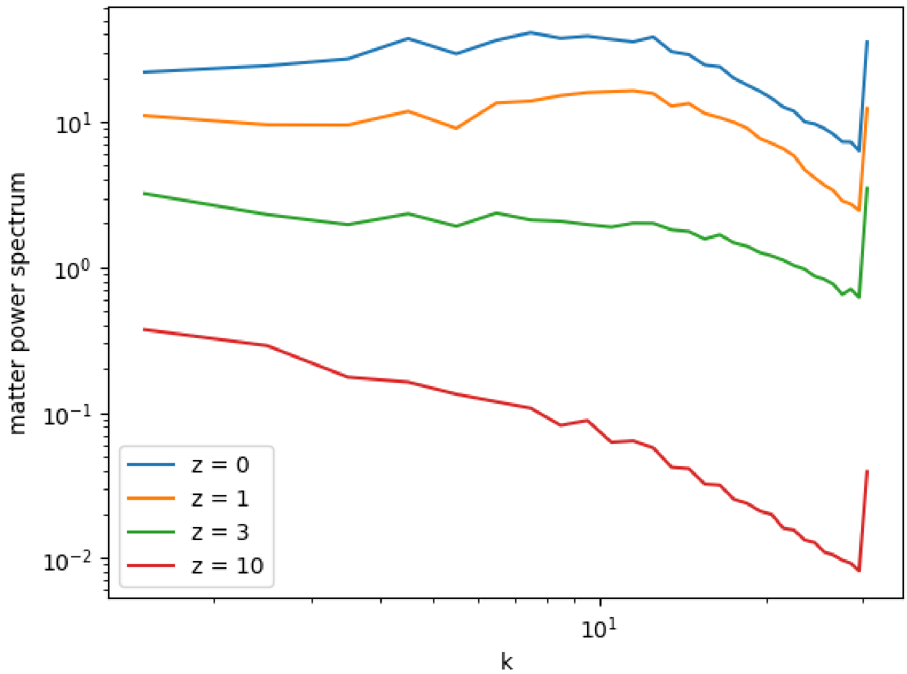
In [ ]:
```
z_array = [0,1,3,10]
idz_array = [-1, 8, 5, 1]
ds = {}
rho_dm = {}
for i, idz in enumerate(idz_array):
    ds[z_array[i]] = ts[idz]
    cg = ds[z_array[i]].covering_grid(0, [0]*3, [64]*3)
    rho_dm[z_array[i]] = cg[("deposit", "all_cic")].in_units("code_density").v
```

```
yt : [INFO     ] 2023-11-20 17:45:35,197 Parameters: current_time              = 650.91854077973
yt : [INFO     ] 2023-11-20 17:45:35,198 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: current_redshift          = -3.3306690738755e-16
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:35,201 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:35,201 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:35,204 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:35,198 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:35,199 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: current_redshift          = -3.3306690738755e-16
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:35,200 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:35,201 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:35,201 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:35,204 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:35,481 Parameters: current_time              = 276.68070327028
yt : [INFO     ] 2023-11-20 17:45:35,482 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:35,482 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:35,482 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:35,483 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:35,483 Parameters: current_redshift          = 0.99999995433926
yt : [INFO     ] 2023-11-20 17:45:35,483 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:35,484 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:35,484 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:35,484 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:35,487 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:35,761 Parameters: current_time              = 101.47857431987
yt : [INFO     ] 2023-11-20 17:45:35,762 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:35,762 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:35,763 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:35,763 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:35,764 Parameters: current_redshift          = 2.9999997678668
yt : [INFO     ] 2023-11-20 17:45:35,764 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:35,764 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:35,764 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:35,765 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:35,767 Gathering a field list (this may take a moment.)
yt : [INFO     ] 2023-11-20 17:45:36,044 Parameters: current_time              = 22.374497863705
yt : [INFO     ] 2023-11-20 17:45:36,044 Parameters: domain_dimensions         = [64 64 64]
yt : [INFO     ] 2023-11-20 17:45:36,044 Parameters: domain_left_edge          = [0. 0. 0.]
yt : [INFO     ] 2023-11-20 17:45:36,045 Parameters: domain_right_edge         = [1. 1. 1.]
yt : [INFO     ] 2023-11-20 17:45:36,045 Parameters: cosmological_simulation   = 1
yt : [INFO     ] 2023-11-20 17:45:36,046 Parameters: current_redshift          = 9.9999870047908
yt : [INFO     ] 2023-11-20 17:45:36,046 Parameters: omega_lambda              = 0.6911
yt : [INFO     ] 2023-11-20 17:45:36,047 Parameters: omega_matter              = 0.3089
yt : [INFO     ] 2023-11-20 17:45:36,047 Parameters: omega_radiation           = 0
yt : [INFO     ] 2023-11-20 17:45:36,047 Parameters: hubble_constant           = 0.6774
yt : [INFO     ] 2023-11-20 17:45:36,049 Gathering a field list (this may take a moment.)
```

In [ ]:

```python
def fft_comp(rho):
    # print(rho.shape)
    nx, ny, nz = rho.shape
    ru = np.fft.fftn(rho)[0:nx//2+1, 0:ny//2+1, 0:nz//2+1]
    ru = 8*ru/(nx*ny*nz)
    return np.abs(ru) ** 2

for z in z_array:
    L = (ds[z].domain_right_edge - ds[z].domain_left_edge).d
    # max_level = ds[z].index.max_level
    # ref = int(np.prod(ds[z].ref_factors[0:max_level]))
    low = ds[z].domain_left_edge
    dims = ds[z].domain_dimensions# * ref
    # print(dims)
    nx, ny, nz = dims

    kx = np.fft.rfftfreq(nx) * nx / L[0]
    ky = np.fft.rfftfreq(ny) * ny / L[1]
    kz = np.fft.rfftfreq(nz) * nz / L[2]

    Kk = np.zeros((nx // 2 + 1, ny // 2 + 1, nz // 2 + 1))
    Kk += fft_comp(rho_dm[z])

    # physical limits to the wavenumbers
    kmin = np.min(1.0 / L)
    kmax = np.min(0.5 * dims / L)

    kbins = np.arange(kmin, kmax, kmin)
```

```python
    N = len(kbins)

    # bin the Fourier KE into radial kbins
    kx3d, ky3d, kz3d = np.meshgrid(kx, ky, kz, indexing="ij")
    k = np.sqrt(kx3d**2 + ky3d**2 + kz3d**2)

    whichbin = np.digitize(k.flat, kbins)
    ncount = np.bincount(whichbin)

    matterps = np.zeros(len(ncount) - 1)

    for n in range(1, len(ncount)):
        matterps[n - 1] = np.sum(Kk.flat[whichbin == n])

    k = 0.5 * (kbins[0 : N - 1] + kbins[1:N])
    matterps = matterps[1:N]

    plt.plot(k, matterps, label=f"z = {z}")
plt.xscale('log')
plt.yscale('log')
plt.xlabel("k")
plt.ylabel("matter power spectrum")
plt.legend()
plt.show()
```



According to this tutorial, the spike at high wavenumbers is due to non-periodicity. Thus we can see that the $z = 10$ case fits the power-law pretty well. However, as time goes on, the power-spectrum increased so much at high wavenumbers which indicates that structure formation has gone non-linear.

The power spectrum provides information about the amplitudes of different spatial frequencies, and the correlation function gives information about how these amplitudes contribute to the clustering at different spatial scales. Mathematically, the Fourier transform of the correlation function is the power spectrum, and the inverse Fourier transform of the power spectrum is the correlation function.