# dark_compact_V1_MR_fix

August 13, 2020

[1]:
```
%load_ext autoreload
%autoreload 2

%config IPCompleter.greedy=True
```

[2]:
```
import matplotlib.pyplot as plt
import numpy as np
import h5py
from astropy import units
from pathlib import Path
import os
import time

import snapshot_obj
import dataset_compute
import curve_fit

import importlib
```

[3]:
```
importlib.reload(snapshot_obj)
importlib.reload(dataset_compute)
importlib.reload(curve_fit)
```

[3]:
```
<module 'curve_fit' from
'/home/kasper/Curvaton_Simulations/analysis/curve_fit.py'>
```

## 1 Massive compact non-luminous halos

Here, we inspect the properties of the clump of subhalos in the ΛCDM simulation with $v_{\rm max} \sim v_{\rm 1kpc} > 30 {\rm km/s}$.

### 1.1 Construct selection arrays

Read $v_{\rm max}$ and $v_{\rm 1kpc}$ and construct mask arrays for the peculiar dark subhalos, and for all dark subhalos in the same mass range.

[4]:
```
sim_id = "V1_MR_fix"
snap = snapshot_obj.Snapshot(sim_id, 127)
max_point = snap.get_subhalos("Max_Vcirc", "Extended")
vmax = max_point[:,0] * units.cm.to(units.km)
v1kpc = snap.get_subhalos("V1kpc", "Extended") * units.cm.to(units.km)
```

[5]:
```
masks_sat, mask_isol = dataset_compute.split_satellites_by_group_number(
    snap, (1,0), (2,0))
mask_dark = dataset_compute.split_luminous(snap)[1]
mask_pecul = np.logical_and(vmax > 30, np.abs(vmax-v1kpc)/vmax < 0.15)
mask_pecul = np.logical_and.reduce([mask_isol, mask_dark, mask_pecul])
mask_ref = np.logical_and(vmax > 25, vmax < 50)
mask_ref = np.logical_and.reduce([mask_isol, mask_dark, mask_ref])
```

Inspect the constituents of the peculiar subhalos:

[6]:
```
masstype = snap.get_subhalos("MassType") * units.g.to(units.Msun)
for m_arr in masstype[mask_pecul]:
    print(["{:.2E}".format(m) for m in m_arr])
```

```
['3.25E+08', '8.17E+09', '0.00E+00', '0.00E+00', '0.00E+00', '0.00E+00']
['1.07E+08', '7.06E+09', '0.00E+00', '0.00E+00', '0.00E+00', '0.00E+00']
['1.55E+08', '6.71E+09', '0.00E+00', '0.00E+00', '0.00E+00', '0.00E+00']
['1.76E+08', '6.26E+09', '0.00E+00', '0.00E+00', '0.00E+00', '0.00E+00']
['6.52E+07', '2.38E+09', '0.00E+00', '0.00E+00', '0.00E+00', '0.00E+00']
```

## 1.2 Plot the rotation curves

Let us first see how many there are of these peculiar halos, and compare that to the number of all halos in the mass range:

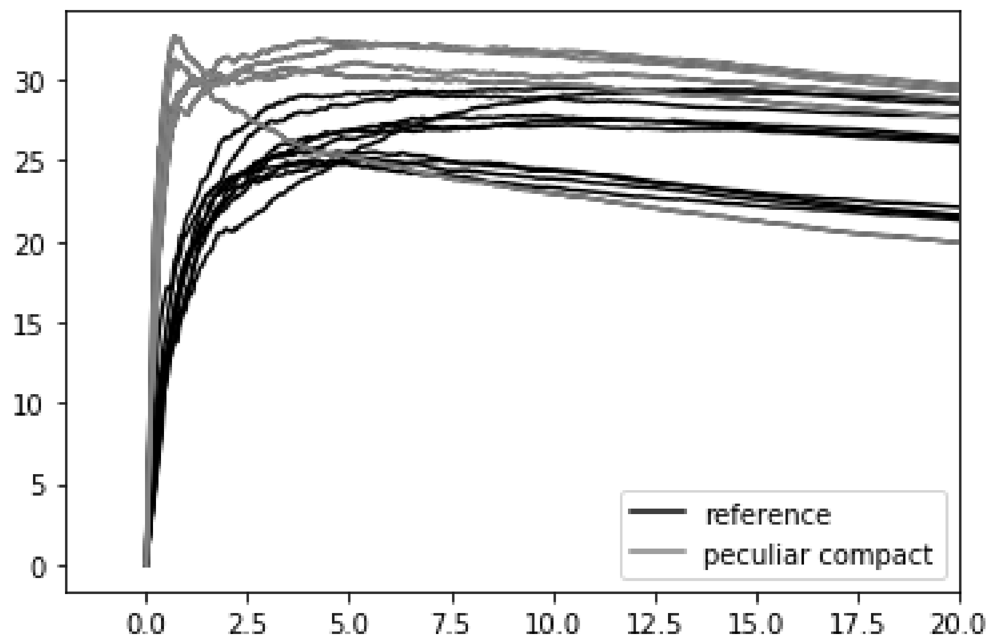
[7]:
```
print(np.sum(mask_pecul))
print(np.sum(mask_ref))
print(np.sum(np.logical_and(mask_pecul, mask_ref)))
```

```
5
15
5
```

We see that the rotation curves of the subhalos in the given mass range divide into two classes: very compact ones, which constitute the clump in the V1kpc plots, and more regular looking ones.

[8]:
```
rot_curves = snap.get_subhalos('Vcirc', group='Extended/RotationCurve/All')
sub_offset = snap.get_subhalos('SubOffset', group='Extended/RotationCurve/All')

fig, axes = plt.subplots()
axes.set_xlim(-2, 20)

idxs = np.arange(mask_pecul.size)[mask_ref]
```

```
for i,idx in enumerate(idxs):
    v_circ = rot_curves[sub_offset[idx]:sub_offset[idx+1],0] * units.cm.
 ↪to(units.km)
    radii = rot_curves[sub_offset[idx]:sub_offset[idx+1],1] * units.cm.to(units.
 ↪kpc)
    if i==0:
        axes.plot(radii, v_circ, c='black', label="reference")
    else:
        axes.plot(radii, v_circ, c='black')

idxs = np.arange(mask_pecul.size)[mask_pecul]
for i,idx in enumerate(idxs):
    v_circ = rot_curves[sub_offset[idx]:sub_offset[idx+1],0] * units.cm.
 ↪to(units.km)
    radii = rot_curves[sub_offset[idx]:sub_offset[idx+1],1] * units.cm.to(units.
 ↪kpc)
    if i==0:
        axes.plot(radii, v_circ, c='gray', label="peculiar compact")
    else:
        axes.plot(radii, v_circ, c='gray')

plt.legend()
plt.savefig("rotation_curves_peculiar_{}.png".format(sim_id), dpi=200)
```




Let us see, what is the constitution of the mass near the centre. We calculate the rotation curves with bins of three particles:

[9]:

```
pts = {'all': None, 'gas': [0], 'dm': [1]}
v_circ = {}
radii = {}

for key, pt in pts.items():
    if pt is None:
        v, r = dataset_compute.compute_rotation_curves(snap, n_soft=3)
    else:
        v, r = dataset_compute.compute_rotation_curves(snap, n_soft=3,␣
 ↪part_type=pt)

    v_circ[key] = v[mask_pecul] * units.cm.to(units.km)
    radii[key] = r[mask_pecul] * units.cm.to(units.kpc)
```

```
Computing subhalo rotation curves for V1_MR_fix…
Done.
Computing subhalo rotation curves for V1_MR_fix…
Done.
Computing subhalo rotation curves for V1_MR_fix…
Done.
```

[10]:

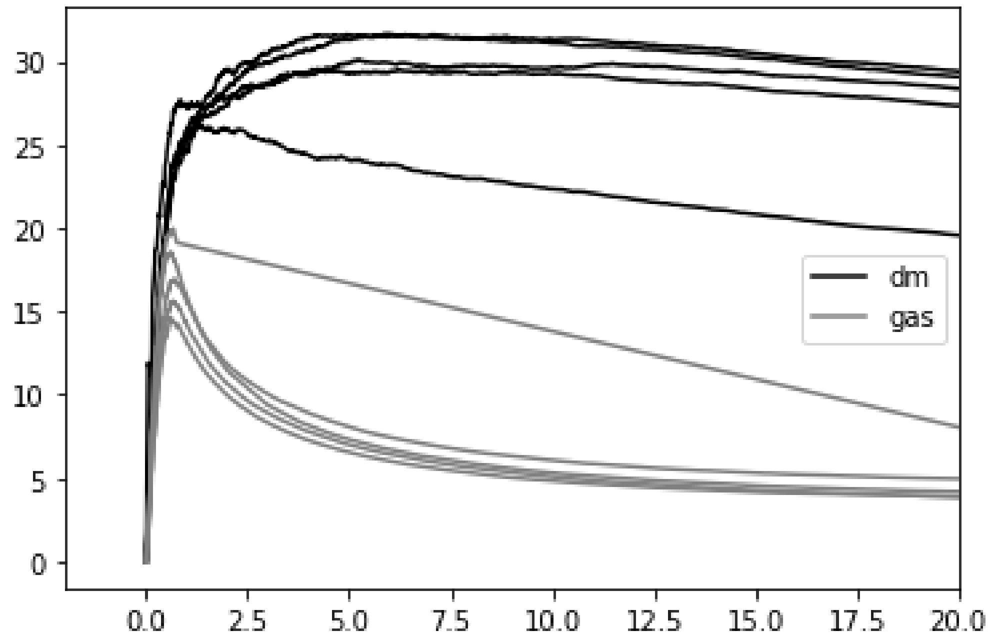
```
for pt in pts.keys():
    print("Particle type: {}\n".format(pt) + \
          "Number of particles in the peculiar subhalos within radii\n" +\
          "   1kpc   2kpc")
    for r in radii[pt]:
        print("  {:3d}    {:3d}".format(np.sum(r < 1), np.sum(r < 2)))
    print('')
```

```
Particle type: all
Number of particles in the peculiar subhalos within radii
   1kpc   2kpc
  240    374
  217    371
  190    336
  260    368
  265    353

Particle type: gas
Number of particles in the peculiar subhalos within radii
   1kpc   2kpc
  157    181
  131    147
  111    128
  163    169
  179    179
```

```
Particle type: dm
Number of particles in the peculiar subhalos within radii
   1kpc   2kpc
   84     194
   87     225
   79     208
   98     199
   87     174
```

[11]:
```python
fig, axes = plt.subplots()

axes.set_xlim(-2, 20)

for i, (v, r) in enumerate(zip(v_circ['dm'], radii['dm'])):
    if i==0:
        axes.plot(r, v, c='black', label="dm")
    else:
        axes.plot(r, v, c='black')

for i, (v, r) in enumerate(zip(v_circ['gas'], radii['gas'])):
    if i==0:
        axes.plot(r, v, c='grey', label="gas")
    else:
        axes.plot(r, v, c='grey')

plt.legend()
```

[11]:
```
<matplotlib.legend.Legend at 0x7fb0f6e1c710>
```




### 1.3 Further inspection

Let us see, how far from the LG centre the peculiar subhalos are located:

[12]:
```
LG = dataset_compute.compute_LG_centre(snap, (1,0), (2,0))
dist_to_lg = dataset_compute.distance_to_point(snap, LG) * units.cm.to(units.
 ↪kpc)
print(dist_to_lg[mask_pecul])
```

```
[5240.2538803  2294.92363458 4697.04628532 4404.5327097  1075.81554943]
```