13 - Calculating Kappa Co

September 11, 2022

[1]:
```
import h5py
import matplotlib.pyplot as plt
from os import listdir
import numpy as np
import matplotlib.ticker as ticker
import re
import pandas as pd

axisScale = 0.03
datasets = ['organic', 'gm_early', 'gm_late']
colours = ['green', 'blue', 'red']

datasetNo = 0

# define storage arrays and reshape to hold all three galaxy type data
redshifts = np.array(range(3*24), dtype=float)
redshifts.shape = (3, 24)
kco_s = np.array(range(3*24), dtype=float)
kco_s.shape = (3, 24)

# Read data for the redshift->time lookup table
df_o = pd.read_csv('halo_catalogue_organic.txt', delimiter='\t')
df_r2t = df_o[["time [Gyr]"]].copy()
df_r2t = df_r2t.drop(df_r2t.index[range(0,3)])
times = df_r2t.to_numpy()

debug = False

for colour, dataset in enumerate(datasets):
    print('-----------------------------------------')
    print(dataset)
    print('-----------------------------------------')

    files = listdir('data/' + dataset)

    count = 0
```

```
    for file in files:

        # get redshift from the filename
        m = re.search('(z[0-9])\w+', file)
        s = m.group(0).replace('z', '')
        s = s.replace('p', '.')
        z = float(s)

        print ('z=', z)
        print ('t= %s Gyr' % times[count])

        # load data for a particular galaxy at a particular redshift
        f = h5py.File('data/' + dataset + '/' + file,'r')

        # extract data from the file
        ds_c = f['Coordinates']
        ds_v = f['Velocity']
        ds_m = f['Mass']

        # Calculate the resultant angular momentum vectors angmom
        # angmom contains the AM vectors for each particle per redshift era
        mv = np.multiply(ds_m, np.transpose(ds_v))
        angMom = np.cross(ds_c, np.transpose(mv))
        if debug: print ("shape angmom", np.shape(angMom))

        # Calculate the total angular momentum vector by summing the vectors␣
↪(per redshift epoch)
        angMomTot = np.sum(angMom, axis=0)
        if debug: print ("shape angMomTot", np.shape(angMomTot))

        # Calculate the magnitude of the total angular momentum vector for each␣
↪redshift epoch
        # We use this to normalise the angular momentum to a unit vector for␣
↪scaling during the transform
        magnitude = np.linalg.norm(angMomTot)

        # Alternative way of calculating the magnitude
        # magnitude2 = np.sqrt(angMomTot[0]**2 + angMomTot[1]**2 +␣
↪angMomTot[2]**2)

        # Convert the angular momentum vectors to a unit size - the other␣
↪vectors will be based on this to prevent distortion
        unitVect_z = angMomTot / magnitude
        if debug: print('angMomTot: ', angMomTot)
        if debug: print('magnitude: ', magnitude)
        if debug: print ("unitVect_z", unitVect_z)
```

```
    if debug: print ("np.linalg.norm(unitVect_z)", np.linalg.
↪norm(unitVect_z))


    # the angular momentum's vector's (unitVect_z) direction is directly␣
↪out of the plane of the galaxy
    # unitVect_z = k, but j = [-k2/k1, 1, 0], so

    k = unitVect_z
    if debug: print ("k: ", k)

    j = [-k[1]/k[0], 1, 0]
    j = j/np.linalg.norm(j)
    if debug: print ("j: ", j)

    i = np.cross(j, k)
    i = i/np.linalg.norm(i)
    if debug: print ("i: %s" %  i)

    #Orthogonal tests
    if debug:
        test_i_result = np.dot(i,j)
        test_j_result = np.dot(i,k)
        test_k_result = np.dot(j,k)
        if abs(test_i_result) < 1e-5:
            test_i = "Pass"
        else:
            test_i = "FAIL"
        if abs(test_j_result) < 1e-5:
            test_j = "Pass"
        else:
            test_j = "FAIL"
        if abs(test_k_result) < 1e-5:
            test_k = "Pass"
        else:
            test_k = "FAIL"
        print('i.j: %s %s' % (test_i_result, test_i))
        print('i.k: %s %s' % (test_j_result, test_j))
        print('j.k: %s %s' % (test_k_result, test_k))


    # transform co-ordinates
    dsc_x_trsfrm = np.dot(ds_c, i)
    dsc_y_trsfrm = np.dot(ds_c, j)
    dsc_z_trsfrm = np.dot(ds_c, k)
    dsc_trans = np.transpose(np.array([dsc_x_trsfrm, dsc_y_trsfrm,␣
↪dsc_z_trsfrm]))
```

```
        #transform velocities
        dsv_x_trsfrm = np.dot(ds_v, i)
        dsv_y_trsfrm = np.dot(ds_v, j)
        dsv_z_trsfrm = np.dot(ds_v, k)
        dsv_trans = np.transpose(np.array([dsv_x_trsfrm, dsv_y_trsfrm,␣
↪dsv_z_trsfrm]))

        # Calculate KE of transformed particles
        # Get magnitudes of the vectors
        vel_magnitude = np.linalg.norm(dsv_trans, axis=1)
        if debug: print ('shape vel_magnitude: ', np.shape(vel_magnitude))
        if debug: print ('shape ds_m: ', np.shape(ds_m))

        # Calculate kinetic energy for all star particles
        K_tot = np.sum(0.5 * np.array(ds_m) * np.square(vel_magnitude))
        if debug: print ('K_tot: ', K_tot)

        # Calculate R, the distance from the centre in the x-y plane
        R = np.sqrt(np.square(dsc_x_trsfrm) + np.square(dsc_y_trsfrm))

        # Calculate momentum
        mv = np.multiply(ds_m, np.transpose(dsv_trans))
        angMom = np.cross(dsc_trans, np.transpose(mv))

        # Extract L_z
        L_z = np.array(angMom[:,2])

        r0 = 0
        Krot_co = 0
        for n in range(0, len(ds_m)):
            if L_z[n] < 0:
                continue
            if R[n] == 0:
                r0 = r0 + 1
                continue
            Krot_co = Krot_co + (0.5 * ds_m[n] * np.square(L_z[n] /␣
↪(ds_m[n]*R[n]) ))

        print('R=0 count:',r0)
        print('Krot_co:',Krot_co)
        print('Krot_tot:',K_tot)

        K_co = Krot_co / K_tot
        print('K_co:',K_co)

        redshifts[datasetNo,count] = z
```

```
        kco_s[datasetNo,count] = K_co

        count = count + 1
        print ('--------------------------------------------------')

    # Next galaxy
    datasetNo = datasetNo + 1
```

```
-----------------------------------------
organic
-----------------------------------------
z= 7.05
t= [0.76] Gyr
R=0 count: 0
Krot_co: 5524279878.3322735
Krot_tot: 36702494430.64071
K_co: 0.15051510705278887
-------------------------------------------------
z= 5.971
t= [0.942] Gyr
R=0 count: 0
Krot_co: 37546370059.415985
Krot_tot: 119774170054.30975
K_co: 0.3134763533939844
-------------------------------------------------
z= 5.487
t= [1.049] Gyr
R=0 count: 0
Krot_co: 42588177690.39492
Krot_tot: 160746946946.8226
K_co: 0.2649392632289539
-------------------------------------------------
z= 5.037
t= [1.168] Gyr
R=0 count: 0
Krot_co: 26944582706.502235
Krot_tot: 170171808213.33142
K_co: 0.1583375236438921
-------------------------------------------------
z= 4.485
t= [1.348] Gyr
R=0 count: 0
Krot_co: 77690905306.52808
Krot_tot: 276901259257.15857
K_co: 0.2805725965817166
-------------------------------------------------
```

```
z= 3.984
t= [1.556] Gyr
R=0 count: 0
Krot_co: 72713941492.38333
Krot_tot: 483293447277.48175
K_co: 0.1504550535539018
-------------------------------------------------
z= 3.528
t= [1.795] Gyr
R=0 count: 1
Krot_co: 130707010706.00284
Krot_tot: 496683807328.24457
K_co: 0.2631593959325962
-------------------------------------------------
z= 3.017
t= [2.144] Gyr
R=0 count: 0
Krot_co: 489740617018.32104
Krot_tot: 1995984484664.5852
K_co: 0.24536293782895782
-------------------------------------------------
z= 2.478
t= [2.653] Gyr
R=0 count: 1
Krot_co: 307053237732.97534
Krot_tot: 1654283268270.0288
K_co: 0.18561103991221353
-------------------------------------------------
z= 2.237
t= [2.949] Gyr
R=0 count: 1
Krot_co: 1392427871606.0225
Krot_tot: 4451823355310.673
K_co: 0.3127769815810338
-------------------------------------------------
z= 2.012
t= [3.276] Gyr
R=0 count: 0
Krot_co: 11419314401341.814
Krot_tot: 39357742880603.41
K_co: 0.29014149607063905
-------------------------------------------------
z= 1.737
t= [3.767] Gyr
R=0 count: 0
Krot_co: 14100978048156.979
Krot_tot: 60571084930892.984
K_co: 0.23280048663888264
```

```
-------------------------------------------------
z= 1.487
t= [4.325] Gyr
R=0 count: 1
Krot_co: 78451878778586.6
Krot_tot: 179291924354592.7
K_co: 0.43756504405312324
-------------------------------------------------
z= 1.259
t= [4.958] Gyr
R=0 count: 1
Krot_co: 420656089292013.7
Krot_tot: 644922883216845.2
K_co: 0.6522579679508358
-------------------------------------------------
z= 1.004
t= [5.864] Gyr
R=0 count: 1
Krot_co: 578313287564592.5
Krot_tot: 839886629218849.5
K_co: 0.6885611312832333
-------------------------------------------------
z= 0.865
t= [6.472] Gyr
R=0 count: 1
Krot_co: 629301572898908.8
Krot_tot: 907798821596240.2
K_co: 0.6932169969028688
-------------------------------------------------
z= 0.736
t= [7.131] Gyr
R=0 count: 1
Krot_co: 660658038428588.8
Krot_tot: 1227208782659834.2
K_co: 0.5383420064813162
-------------------------------------------------
z= 0.615
t= [7.841] Gyr
R=0 count: 1
Krot_co: 736403732823198.8
Krot_tot: 1404883458356614.8
K_co: 0.5241742497876791
-------------------------------------------------
z= 0.503
t= [8.603] Gyr
R=0 count: 1
Krot_co: 788894197278415.5
Krot_tot: 1541633063829232.2
```

```
K_co: 0.5117263088006796
--------------------------------------------------
z= 0.366
t= [9.698] Gyr
R=0 count: 1
Krot_co: 729198217534447.8
Krot_tot: 1541911824800068.5
K_co: 0.4729182342375505
--------------------------------------------------
z= 0.271
t= [10.577] Gyr
R=0 count: 1
Krot_co: 655907139424084.2
Krot_tot: 1510538822647821.5
K_co: 0.4342206433823035
--------------------------------------------------
z= 0.183
t= [11.501] Gyr
R=0 count: 1
Krot_co: 625061546302555.9
Krot_tot: 1500461714235284.5
K_co: 0.4165794704206236
--------------------------------------------------
z= 0.101
t= [12.469] Gyr
R=0 count: 1
Krot_co: 613680078270110.4
Krot_tot: 1465439071999007.2
K_co: 0.4187687430996286
--------------------------------------------------
z= 0.0
t= [13.821] Gyr
R=0 count: 1
Krot_co: 616638956016225.1
Krot_tot: 1446793967528535.5
K_co: 0.4262106214539929
--------------------------------------------------
-----------------------------------------
gm_early
-----------------------------------------
z= 7.05
t= [0.76] Gyr
R=0 count: 0
Krot_co: 5434859618.363703
Krot_tot: 23066505006.794815
K_co: 0.2356169526663328
--------------------------------------------------
z= 5.971
```

```
t= [0.942] Gyr
R=0 count: 1
Krot_co: 9918261465.565002
Krot_tot: 70320167853.50604
K_co: 0.1410443371839946
-------------------------------------------------
z= 5.487
t= [1.049] Gyr
R=0 count: 0
Krot_co: 17161616918.635698
Krot_tot: 61561241205.997925
K_co: 0.2787730816084267
-------------------------------------------------
z= 5.037
t= [1.168] Gyr
R=0 count: 1
Krot_co: 289357462810.42163
Krot_tot: 1304289947559.075
K_co: 0.22185056578251042
-------------------------------------------------
z= 4.485
t= [1.348] Gyr
R=0 count: 0
Krot_co: 452142220368.8009
Krot_tot: 2151001436197.9253
K_co: 0.21020079892089708
-------------------------------------------------
z= 3.984
t= [1.556] Gyr
R=0 count: 1
Krot_co: 265937194524.9409
Krot_tot: 1239580122163.0303
K_co: 0.21453812445853715
-------------------------------------------------
z= 3.528
t= [1.795] Gyr
R=0 count: 1
Krot_co: 950530663486.0294
Krot_tot: 3309892525500.173
K_co: 0.2871787093275454
-------------------------------------------------
z= 3.017
t= [2.144] Gyr
R=0 count: 1
Krot_co: 5437603701796.342
Krot_tot: 18472819653951.766
K_co: 0.2943569960438125
-------------------------------------------------
```

```
z= 2.478
t= [2.653] Gyr
R=0 count: 1
Krot_co: 133236331424859.73
Krot_tot: 284916810499416.44
K_co: 0.4676323983527558
--------------------------------------------------
z= 2.237
t= [2.949] Gyr
R=0 count: 1
Krot_co: 374554759895582.4
Krot_tot: 598733507776185.5
K_co: 0.6255784168264654
--------------------------------------------------
z= 2.012
t= [3.276] Gyr
R=0 count: 1
Krot_co: 696040690408514.5
Krot_tot: 1043188159288270.0
K_co: 0.6672244927352302
--------------------------------------------------
z= 1.737
t= [3.767] Gyr
R=0 count: 1
Krot_co: 604403938711711.6
Krot_tot: 941729586263992.1
K_co: 0.641802007208342
--------------------------------------------------
z= 1.487
t= [4.325] Gyr
R=0 count: 1
Krot_co: 503519095823553.2
Krot_tot: 856020075996005.9
K_co: 0.5882094473516793
--------------------------------------------------
z= 1.259
t= [4.958] Gyr
R=0 count: 1
Krot_co: 443290310530203.44
Krot_tot: 848841590801263.1
K_co: 0.5222297249970516
--------------------------------------------------
z= 1.004
t= [5.864] Gyr
R=0 count: 1
Krot_co: 315585053009394.5
Krot_tot: 783756437264932.6
K_co: 0.40265704752702086
```

```
-------------------------------------------------
z= 0.865
t= [6.472] Gyr
R=0 count: 1
Krot_co: 259507785638879.0
Krot_tot: 742026176622576.4
K_co: 0.34972861310642794
-------------------------------------------------
z= 0.736
t= [7.131] Gyr
R=0 count: 1
Krot_co: 416637200618539.25
Krot_tot: 1070985588061629.9
K_co: 0.3890222289289703
-------------------------------------------------
z= 0.615
t= [7.841] Gyr
R=0 count: 1
Krot_co: 207972572393450.4
Krot_tot: 694407286552043.9
K_co: 0.2994965295167067
-------------------------------------------------
z= 0.503
t= [8.603] Gyr
R=0 count: 1
Krot_co: 208774914331322.84
Krot_tot: 711102390450357.4
K_co: 0.2935933237393576
-------------------------------------------------
z= 0.366
t= [9.698] Gyr
R=0 count: 1
Krot_co: 195377369888197.2
Krot_tot: 719368814068828.8
K_co: 0.2715955516379997
-------------------------------------------------
z= 0.271
t= [10.577] Gyr
R=0 count: 1
Krot_co: 180515692218865.53
Krot_tot: 693028080756410.5
K_co: 0.26047384980683663
-------------------------------------------------
z= 0.183
t= [11.501] Gyr
R=0 count: 1
Krot_co: 173914369633645.94
Krot_tot: 700030832556549.0
```

```
K_co: 0.24843815664304617
-------------------------------------------------
z= 0.101
t= [12.469] Gyr
R=0 count: 1
Krot_co: 171421126870378.72
Krot_tot: 697350880078298.0
K_co: 0.2458176102841251
-------------------------------------------------
z= 0.0
t= [13.821] Gyr
R=0 count: 1
Krot_co: 172695521275575.1
Krot_tot: 710559917561270.6
K_co: 0.2430414621025732
-------------------------------------------------
-----------------------------------------
gm_late
-----------------------------------------
z= 7.05
t= [0.76] Gyr
R=0 count: 0
Krot_co: 5613497010.9829
Krot_tot: 17027921328.29594
K_co: 0.32966425571010477
-------------------------------------------------
z= 5.971
t= [0.942] Gyr
R=0 count: 0
Krot_co: 14141587793.740345
Krot_tot: 100246965110.20807
K_co: 0.14106749045413564
-------------------------------------------------
z= 5.487
t= [1.049] Gyr
R=0 count: 1
Krot_co: 23277643864.75515
Krot_tot: 92253840136.04788
K_co: 0.25232167929733135
-------------------------------------------------
z= 5.037
t= [1.168] Gyr
R=0 count: 0
Krot_co: 32101885548.100502
Krot_tot: 131972968214.76765
K_co: 0.2432459160565302
-------------------------------------------------
z= 4.485
```

```
t= [1.348] Gyr
R=0 count: 0
Krot_co: 7527474763.802637
Krot_tot: 32293551558.347992
K_co: 0.23309528994362838
-------------------------------------------------
z= 3.984
t= [1.556] Gyr
R=0 count: 0
Krot_co: 45467757206.55435
Krot_tot: 368938648667.5443
K_co: 0.12323934445676894
-------------------------------------------------
z= 3.528
t= [1.795] Gyr
R=0 count: 0
Krot_co: 97336961099.47969
Krot_tot: 635228988264.0859
K_co: 0.15323129595435506
-------------------------------------------------
z= 3.017
t= [2.144] Gyr
R=0 count: 0
Krot_co: 558365694407.6399
Krot_tot: 2106516254446.5386
K_co: 0.26506593207102674
-------------------------------------------------
z= 2.478
t= [2.653] Gyr
R=0 count: 1
Krot_co: 278490601397.074
Krot_tot: 1949846065236.5684
K_co: 0.14282696791414948
-------------------------------------------------
z= 2.237
t= [2.949] Gyr
R=0 count: 0
Krot_co: 128918287553.62604
Krot_tot: 904945188757.6228
K_co: 0.14245977453133357
-------------------------------------------------
z= 2.012
t= [3.276] Gyr
R=0 count: 1
Krot_co: 419324846073.4511
Krot_tot: 2715832769395.873
K_co: 0.1544000981204481
-------------------------------------------------
```

```
z= 1.737
t= [3.767] Gyr
R=0 count: 0
Krot_co: 950947909102.4252
Krot_tot: 4463212775891.092
K_co: 0.21306353894646352
-------------------------------------------------
z= 1.487
t= [4.325] Gyr
R=0 count: 1
Krot_co: 720069619772.307
Krot_tot: 3354255334724.364
K_co: 0.21467346636313206
-------------------------------------------------
z= 1.259
t= [4.958] Gyr
R=0 count: 1
Krot_co: 534924066568.0968
Krot_tot: 1770152025077.2424
K_co: 0.3021910316119627
-------------------------------------------------
z= 1.004
t= [5.864] Gyr
R=0 count: 0
Krot_co: 33763757890.949078
Krot_tot: 89931016633.9431
K_co: 0.3754406338847665
-------------------------------------------------
z= 0.865
t= [6.472] Gyr
R=0 count: 0
Krot_co: 588772822172.5251
Krot_tot: 1305541780452.1318
K_co: 0.45097968597268706
-------------------------------------------------
z= 0.736
t= [7.131] Gyr
R=0 count: 0
Krot_co: 1539124043706.0867
Krot_tot: 5656562121601.304
K_co: 0.2720953134817477
-------------------------------------------------
z= 0.615
t= [7.841] Gyr
R=0 count: 1
Krot_co: 4267340699977.1133
Krot_tot: 12161308211293.09
K_co: 0.35089487297217137
```

```
--------------------------------------------------
z= 0.503
t= [8.603] Gyr
R=0 count: 0
Krot_co: 6860368606604.737
Krot_tot: 15200896171437.27
K_co: 0.45131343107885186
--------------------------------------------------
z= 0.366
t= [9.698] Gyr
R=0 count: 1
Krot_co: 8072787732864.815
Krot_tot: 18831936036446.086
K_co: 0.4286754010443363
--------------------------------------------------
z= 0.271
t= [10.577] Gyr
R=0 count: 1
Krot_co: 9638804413452.809
Krot_tot: 20599576889648.26
K_co: 0.46791273748425966
--------------------------------------------------
z= 0.183
t= [11.501] Gyr
R=0 count: 1
Krot_co: 9890304161701.756
Krot_tot: 20523756605165.45
K_co: 0.4818954128121238
--------------------------------------------------
z= 0.101
t= [12.469] Gyr
R=0 count: 1
Krot_co: 10590940415134.719
Krot_tot: 22796719730851.395
K_co: 0.4645817705431419
--------------------------------------------------
z= 0.0
t= [13.821] Gyr
R=0 count: 1
Krot_co: 9061113173161.084
Krot_tot: 20793542069571.695
K_co: 0.4357657364408682
--------------------------------------------------
```

[11]:
```
plt.figure(figsize = (12,10))
plt.title('$\kappa_{co}$ vs. time for the three galaxy scenarios', pad=10)
plt.xlabel('$t [Gyr]$')
```

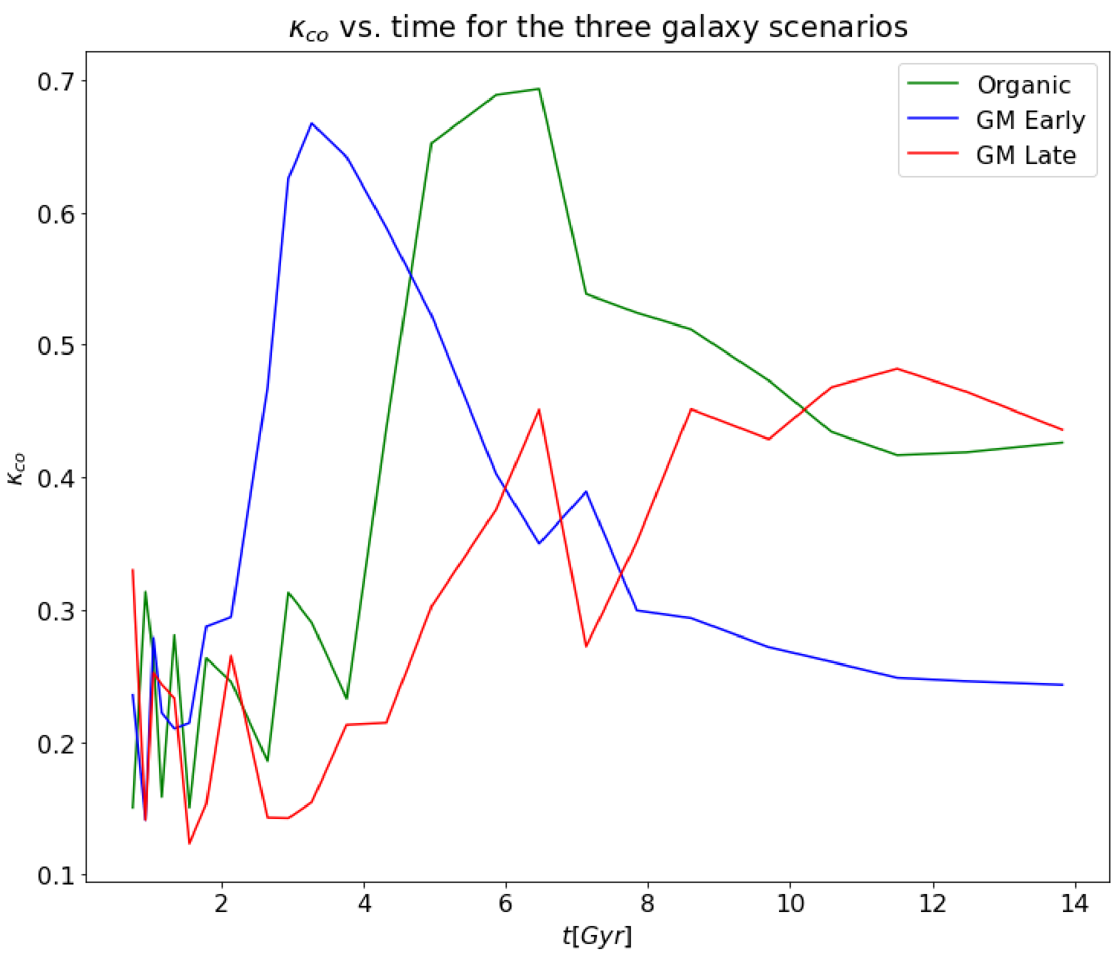
```
plt.ylabel('$\kappa_{co}$')

for i in range(0,3):
    plt.rcParams['font.size'] = '16'
    plt.plot(times, kco_s[i,0:], color=colours[i])

plt.legend(['Organic', 'GM Early', 'GM Late'])

#plt.xlim(0.3, 20)

plt.show()
```




```
[ ]:
```