Equivalent Circuits

Lee Farrugia

10th November 2022

Abstract

1 Introduction & Theoretical Background

2 Results & Discussion

3 Conclusion

4 References

5 Appendix

```
import numpy as np
import matplotlib.pyplot as plt
import pynbody
import pynbody.plot.sph as sph
from functools import reduce

#Question 1
s1 = pynbody.load('run708main.01000')

print(s1.families()) # Exploring groups in simulation
print(s1.properties) # Showing the properties of the simulation
print(f's1.s: {s1.s.loadable_keys()}') # Showing keys for stars
print(f's1.d: {s1.d.loadable_keys()}') # Showing keys for dark
   matter
print(f's1.g: {s1.g.loadable_keys()}') # Showing keys for gas

s2 = pynbody.load('run708mainDiff.01000')

print(s2.families()) # Exploring groups in simulation
print(s2.properties) # Showing the properties of the simulation
print(f's2.s: {s2.s.loadable_keys()}') # Showing keys for stars
print(f's2.d: {s2.d.loadable_keys()}') # Showing keys for dark
   matter
```

```
print(f's2.g: {s2.g.loadable_keys()}') # Showing keys for gas

s1.physical_units() # Changing units to correct units
pynbody.analysis.angmom.faceon(s1)


s2.physical_units() # Changing units to correct units
pynbody.analysis.angmom.faceon(s2)

#Question 2
mass_s1_s = s1.s['mass'] # Extracting the mass contained within
   the stellar component
mass_s1_d = s1.d['mass'] # Extracting the mass contained within
   the dark matter component
mass_s1_g = s1.g['mass'] # Extracting the mass contained within
   the gas component

mass_s2_s = s2.s['mass'] # Extracting the mass contained within
   the stellar component
mass_s2_d = s2.d['mass'] # Extracting the mass contained within
   the dark matter component
mass_s2_g = s2.g['mass'] # Extracting the mass contained within
   the gas component

Sum = 0
array = []

def mass_sum(array):
  Sum = reduce(lambda a, b: a+b, array)
  return(Sum)

# Calculating the total mass contained within the stellar, dark
   matter, and gas components
total_mass_s1 = mass_sum(mass_s1_s) + mass_sum(mass_s1_d) +
   mass_sum(mass_s1_g)

Sum = 0
array = []

# Calculating the total mass contained within the stellar, dark
   matter, and gas components
total_mass_s2 = mass_sum(mass_s2_s) + mass_sum(mass_s2_d) +
   mass_sum(mass_s2_g)

print(f'Total mass s1: {total_mass_s1}')
print(f'Total mass s2: {total_mass_s2}')

#Question 3
```

```
# Rendering a density heat map and optical image of the face-on
   image of the stellar component
plt.figure(figsize=(7.5,10.5))
pynbody.plot.image(s1.s, threaded=False)
plt.title('Denisty Heat map of system 1, face on view')
plt.savefig(f'Plots/Figure 1.png', dpi=800)
plt.clf

plt.figure(figsize=(7.5,10.5))
pynbody.plot.stars.render(s1.s)
plt.title('Optical image of system 1, face on view')
plt.savefig(f'Plots/Figure 2.png', dpi=800)
plt.clf

# Rendering a density heat map and optical image of the face-on
   image of the stellar component
plt.figure(figsize=(7.5,10.5))
pynbody.plot.image(s2.s, threaded=False)
plt.title('Denisty Heat map of system 2, face on view')
plt.savefig(f'Plots/Figure 3.png', dpi=800)
plt.clf

plt.figure(figsize=(7.5,10.5))
pynbody.plot.stars.render(s2.s)
plt.title('Optical image of system 2, face on view')
plt.savefig(f'Plots/Figure 4.png', dpi=800)
plt.clf


#Question 4
# Generating a face-on image of the gaseous component
plt.figure(figsize=(7.5,10.5))
pynbody.plot.image(s1.g, threaded=False)
plt.title('Denisty Heat map of gaseous component of system 1,
   face on view')
plt.savefig(f'Plots/Figure 5.png', dpi=800)
plt.clf

# Generating a face-on image of the gaseous component
plt.figure(figsize=(7.5,10.5))
pynbody.plot.image(s2.g, threaded=False)
plt.title('Denisty Heat map of gaseous component of system 2,
   face on view')
plt.savefig(f'Plots/Figure 6.png', dpi=800)
plt.clf

#Question 6
# Generating the stellar radial density profile using a
```

```
   logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'

plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

p1 = pynbody.analysis.profile.Profile(s1.s, max=30, nbins=200,
   ndim=3)
plt.plot(p1['rbins'], p1['rho'], color='k')
plt.ylabel(r'$\log{\rho}$')
plt.semilogy()
plt.xlabel('Stellar Radii')
plt.title('Stellar radial density profile of system 1')
plt.savefig(f'Plots/Figure 7.png', dpi=800)
plt.clf

# Generating the stellar radial density profile using a
   logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'

plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

p2 = pynbody.analysis.profile.Profile(s2.s, max=30, nbins=200,
   ndim=3)
plt.plot(p2['rbins'], p2['rho'], color='k')
plt.ylabel(r'$\log{\rho}$')
plt.semilogy()
plt.xlabel('Stellar Radii')
plt.title('Stellar radial density profile of system 2')
plt.savefig(f'Plots/Figure 8.png', dpi=800)
plt.clf

#Task2

#Question 1
```

```
# Rotating the barred galaxy so that its bar is aligned with
   the x-axis
pynbody.analysis.angmom.sideon(s1)
pynbody.plot.image(s1.s, threaded=False)
plt.title('Denisty Heat map of System 1, side-on view')
plt.savefig(f'Plots/Figure 9.png', dpi=800)
plt.clf

#Question 2
radius = 4 # Defining the radius to be considered
centre = (0,0,0) # Defining the center of the galaxy

sphere1 = s1.s[pynbody.filt.Sphere(radius, centre)] # Filtering
   the stars according to radii
pynbody.analysis.angmom.sideon(sphere1) # Aligning simulation
   to appear side-on

# Rendering a density heat map of the side-on image of the
   filtered stars
sph.image(s1.s, width='8 kpc')
plt.title('Denisty Heat map of filtered system 1, side-on view')
plt.savefig(f'Plots/Figure 10.png', dpi=800)
plt.clf

pynbody.plot.stars.render(sphere1, width='8 kpc')
plt.title('Optical image of filtered system 1, side-on view')
plt.savefig(f'Plots/Figure 11.png', dpi=800)
plt.clf

pynbody.analysis.angmom.sideon(s2)

sphere2 = s2.s[pynbody.filt.Sphere(radius, centre)] # Filtering
   the stars according to radii
pynbody.analysis.angmom.sideon(sphere2) # Aligning simulation
   to appear side-on

# Rendering a density heat map of the side-on image of the
   filtered stars
sph.image(s2.s, width='8 kpc')
plt.title('Denisty Heat map of filtered system 2, side-on view')
plt.savefig(f'Plots/Figure 12.png', dpi=800)
plt.clf

pynbody.plot.stars.render(sphere2, width='8 kpc')
plt.title('Optical image of filtered system 2, side-on view')
plt.savefig(f'Plots/Figure 13.png', dpi=800)
plt.clf
```

```
#Question 3
pynbody.analysis.angmom.sideon(s1) # Viewing the galaxy from
    the side

# Rendering a density heat map and optical image of the side-on
    image of the galaxy
sph.image(s1.s, width='1.5 kpc')
plt.title('Denisty Heat map of filtered system 1, side-on view')
plt.savefig(f'Plots/Figure 14.png', dpi=800)
plt.clf

pynbody.plot.stars.render(s1.s, width='1.5 kpc')
plt.title('Optical image of filtered system 1, side-on view')
plt.savefig(f'Plots/Figure 15.png', dpi=800)
plt.clf

pynbody.analysis.angmom.sideon(s2) # Viewing the galaxy from
    the side

# Rendering a density heat map and optical image of the side-on
    image of the galaxy
sph.image(s2.s, width='1.5 kpc')
plt.title('Denisty Heat map of filtered system 2, side-on view')
plt.savefig(f'Plots/Figure 16.png', dpi=800)
plt.clf

pynbody.plot.stars.render(s2.s, width='1.5 kpc')
plt.title('Optical image of filtered system 2, side-on view')
plt.savefig(f'Plots/Figure 17.png', dpi=800)
plt.clf

#Question 4
# Creating filters for the different radii
radius_filter_0 = pynbody.filt.BandPass('pos', '0 kpc', '0.25
    kpc')
radius_filter_1 = pynbody.filt.BandPass('pos', '0.25 kpc', '0.5
    kpc')
radius_filter_2 = pynbody.filt.BandPass('pos', '0.5 kpc', '0.75
    kpc')
radius_filter_3 = pynbody.filt.BandPass('pos', '0.75 kpc', '1
    kpc')
radius_filter_4 = pynbody.filt.BandPass('pos', '1 kpc', '1.25
    kpc')
radius_filter_5 = pynbody.filt.BandPass('pos', '1.25 kpc', '1.5
    kpc')

# Filtering stars according to their radii, changing radius in
    steps of 0.25 from 0 to 1.5
```

```
s1_1_filtered = s1.s[radius_filter_0]
s1_2_filtered = s1.s[radius_filter_1]
s1_3_filtered = s1.s[radius_filter_2]
s1_4_filtered = s1.s[radius_filter_3]
s1_5_filtered = s1.s[radius_filter_4]
s1_6_filtered = s1.s[radius_filter_5]

s2_1_filtered = s2.s[radius_filter_0]
s2_2_filtered = s2.s[radius_filter_1]
s2_3_filtered = s2.s[radius_filter_2]
s2_4_filtered = s2.s[radius_filter_3]
s2_5_filtered = s2.s[radius_filter_4]
s2_6_filtered = s2.s[radius_filter_5]

# Generating profiles for each filter
rho_s1_1 = pynbody.analysis.profile.Profile(s1_1_filtered,
   ndim=3)
rho_s1_2 = pynbody.analysis.profile.Profile(s1_2_filtered,
   ndim=3)
rho_s1_3 = pynbody.analysis.profile.Profile(s1_3_filtered,
   ndim=3)
rho_s1_4 = pynbody.analysis.profile.Profile(s1_4_filtered,
   ndim=3)
rho_s1_5 = pynbody.analysis.profile.Profile(s1_5_filtered,
   ndim=3)
rho_s1_6 = pynbody.analysis.profile.Profile(s1_6_filtered,
   ndim=3)

# Plotting the generated stellar radial density profile using a
   logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'

plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

plt.plot(rho_s1_1['rbins'], rho_s1_1['rho'], label='0 to 0.25')
plt.plot(rho_s1_2['rbins'], rho_s1_2['rho'], label='0.25 to
   0.50')
plt.plot(rho_s1_3['rbins'], rho_s1_3['rho'], label='0.50 to
   0.75')
plt.plot(rho_s1_4['rbins'], rho_s1_4['rho'], label='0.75 to
   1.00')
```

```
plt.plot(rho_s1_5['rbins'], rho_s1_5['rho'], label='1.00 to
    1.25')
plt.plot(rho_s1_6['rbins'], rho_s1_6['rho'], label='1.25 to
    1.50')
plt.semilogy()
plt.ylabel(r'$\log{\rho}$')
plt.xlabel('Stellar Radii')
plt.title('Stellar radial density profile of system 1')
plt.legend()
plt.savefig(f'Plots/Figure 18.png', dpi=800)
plt.clf

# Generating profiles for each filter
rho_s2_1 = pynbody.analysis.profile.Profile(s2_1_filtered,
    ndim=3)
rho_s2_2 = pynbody.analysis.profile.Profile(s2_2_filtered,
    ndim=3)
rho_s2_3 = pynbody.analysis.profile.Profile(s2_3_filtered,
    ndim=3)
rho_s2_4 = pynbody.analysis.profile.Profile(s2_4_filtered,
    ndim=3)
rho_s2_5 = pynbody.analysis.profile.Profile(s2_5_filtered,
    ndim=3)
rho_s2_6 = pynbody.analysis.profile.Profile(s2_6_filtered,
    ndim=3)

# Plotting the generated stellar radial density profile using a
    logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'

plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

plt.plot(rho_s2_1['rbins'], rho_s2_1['rho'], label='0 to 0.25')
plt.plot(rho_s2_2['rbins'], rho_s2_2['rho'], label='0.25 to
    0.50')
plt.plot(rho_s2_3['rbins'], rho_s2_3['rho'], label='0.50 to
    0.75')
plt.plot(rho_s2_4['rbins'], rho_s2_4['rho'], label='0.75 to
    1.00')
plt.plot(rho_s2_5['rbins'], rho_s2_5['rho'], label='1.00 to
    1.25')
```

```
plt.plot(rho_s2_6['rbins'], rho_s2_6['rho'], label='1.25 to
    1.50')
plt.semilogy()
plt.ylabel(r'$\log{\rho}$')
plt.xlabel('Stellar Radii')
plt.title('Stellar radial density profile of system 2')
plt.legend()
plt.savefig(f'Plots/Figure 19.png', dpi=800)
plt.clf

#Question 5
# Creating filters for the different heights
age_filter_1 = pynbody.filt.BandPass('pos', '0.75 kpc', '0.8
    kpc')
age_filter_2 = pynbody.filt.BandPass('pos', '0.8 kpc', '0.85
    kpc')
age_filter_3 = pynbody.filt.BandPass('pos', '0.85 kpc', '0.9
    kpc')
age_filter_4 = pynbody.filt.BandPass('pos', '0.9 kpc', '0.95
    kpc')
age_filter_5 = pynbody.filt.BandPass('pos', '0.95 kpc', '1 kpc')

# Filtering stars according to their radii, changing radius in
    steps of 0.05 from 0.75 to 1.00
s1_1_filtered = s1.s[age_filter_1]
s1_2_filtered = s1.s[age_filter_2]
s1_3_filtered = s1.s[age_filter_3]
s1_4_filtered = s1.s[age_filter_4]
s1_5_filtered = s1.s[age_filter_5]

s2_1_filtered = s2.s[age_filter_1]
s2_2_filtered = s2.s[age_filter_2]
s2_3_filtered = s2.s[age_filter_3]
s2_4_filtered = s2.s[age_filter_4]
s2_5_filtered = s2.s[age_filter_5]

# Generating profiles for each filter
s1_p_1 = pynbody.analysis.profile.Profile(s1_1_filtered,
    nbins=200)
s1_p_2 = pynbody.analysis.profile.Profile(s1_2_filtered,
    nbins=200)
s1_p_3 = pynbody.analysis.profile.Profile(s1_3_filtered,
    nbins=200)
s1_p_4 = pynbody.analysis.profile.Profile(s1_4_filtered,
    nbins=200)
s1_p_5 = pynbody.analysis.profile.Profile(s1_5_filtered,
    nbins=200)
```

```
# Plotting the generated stellar age density profile using a
   logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'

plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

plt.plot(s1_p_1['rbins'], s1_p_1['rho'], label='0.75 to 0.80')
plt.plot(s1_p_2['rbins'], s1_p_2['rho'], label='0.80 to 0.85')
plt.plot(s1_p_3['rbins'], s1_p_3['rho'], label='0.85 to 0.90')
plt.plot(s1_p_4['rbins'], s1_p_4['rho'], label='0.90 to 0.95')
plt.plot(s1_p_5['rbins'], s1_p_5['rho'], label='0.95 to 1.00')

plt.semilogy()
plt.ylabel(r'$\log{\rho}$')
plt.xlabel('Stellar Ages')
plt.title('Stellar age density profile of system 1')
plt.legend()
plt.savefig(f'Plots/Figure 20.png', dpi=800)
plt.clf

# Generating profiles for each filter
s2_p_1 = pynbody.analysis.profile.Profile(s2_1_filtered,
   nbins=200)
s2_p_2 = pynbody.analysis.profile.Profile(s2_2_filtered,
   nbins=200)
s2_p_3 = pynbody.analysis.profile.Profile(s2_3_filtered,
   nbins=200)
s2_p_4 = pynbody.analysis.profile.Profile(s2_4_filtered,
   nbins=200)
s2_p_5 = pynbody.analysis.profile.Profile(s2_5_filtered,
   nbins=200)

# Plotting the generated stellar age density profile using a
   logarithmic scale on the y-axis
plt.figure(figsize=(7.5,10.5))

plt.rcParams['font.family'] = 'STIXGeneral'
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams['font.size'] = 12
plt.rcParams['font.weight'] = 'normal'
```

```
plt.minorticks_on()
plt.grid(visible=True, which='major', linestyle='-')
plt.grid(visible=True, which='minor', linestyle='--')

plt.plot(s2_p_1['rbins'], s2_p_1['rho'], label='0.75 to 0.80')
plt.plot(s2_p_2['rbins'], s2_p_2['rho'], label='0.80 to 0.85')
plt.plot(s2_p_3['rbins'], s2_p_3['rho'], label='0.85 to 0.90')
plt.plot(s2_p_4['rbins'], s2_p_4['rho'], label='0.90 to 0.95')
plt.plot(s2_p_5['rbins'], s2_p_5['rho'], label='0.95 to 1.00')

plt.semilogy()
plt.ylabel(r'$\log{\rho}$')
plt.xlabel('Stellar Ages')
plt.title('Stellar age density profile of system 2')
plt.legend()
plt.savefig(f'Plots/Figure 21.png', dpi=800)
plt.clf
```