Nikhil Patten
25 April, 2022
Dr. Pierce
ASTR5465

As we discussed in class, I would like you to make your own simulations of the "Stochastic, self-propagating, star formation model. The relevant references are:

Gerola & Seiden 1978, ApJ, 223, 129
Seiden & Gerola 1979, ApJ, 233, 56

Do the simulation by:

1. **make a fine grid in polar coordinates,**
2. **use a random number generator to assign star formation randomly with (say) 5-10% probability in each time step.**
3. **Assign a fading time scale for the star forming region of (say) 50 million years, and a "dead-period" during which that point cannot form stars.**
4. **For each time step (t 5-10 million years) rotate your grid so that the galaxy rotates differentially with a constant velocity (flat rotation curve). You can just propagate any star forming region the random number generator "creates" to the appropriate grid point on your galaxy for each time step.**
5. **During this time step assign a second probability that any stochastic star forming event (from part 2 above) from the previous time step propagates to adjoining cells.**
6. **Repeat steps 2-5 as your model galaxy rotates and plot symbols for your star formation sites in polar coordinates for several time steps. Your plots don't have to be adjacent time steps since we want to see the system evolve but they should represent some reasonable time interval, say every ten time steps.**

You might need to iterate a bit to find reasonable values for the time intervals, propagation probabilities and rotational velocity and the fading times to get nice looking plots.

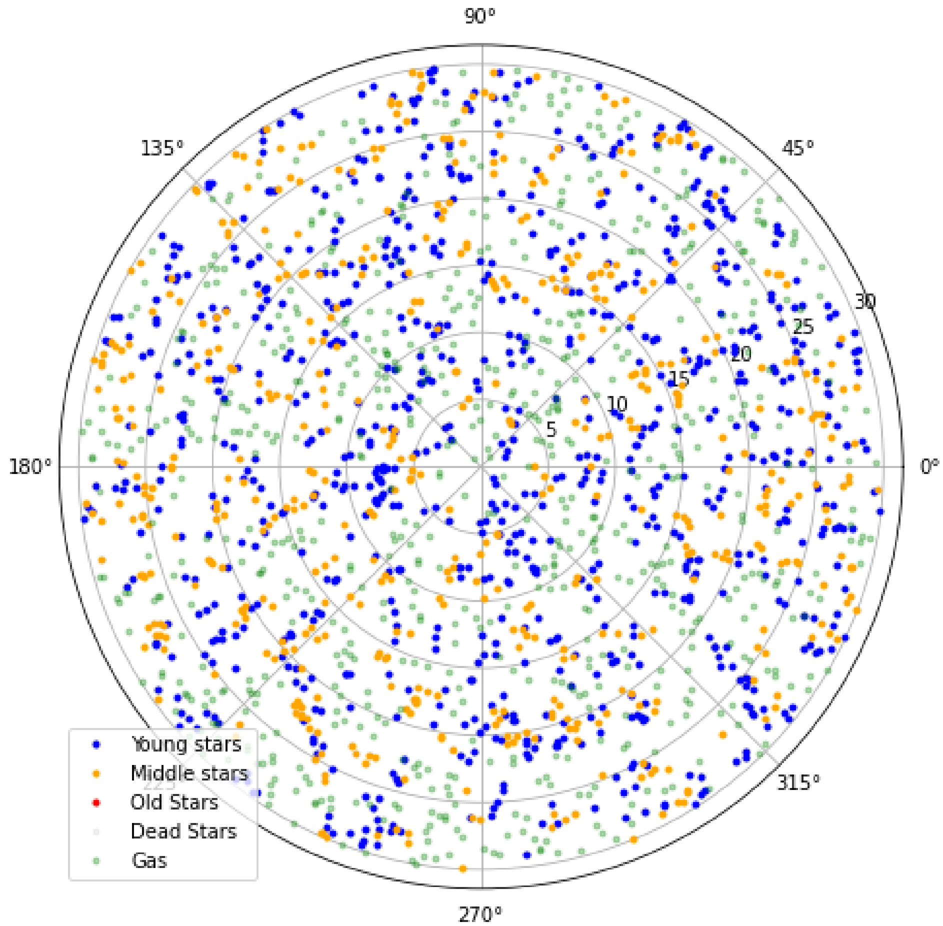
Turn in both the plots and your computer code. This is a simple but fun calculation so you can use anything you wish: python, even Excel (cool!). Just be sure that you use something that has libraries or capabilities for making polar plots to make your life easier. Let me know if you have any questions.

Code:

```
import time
import numpy as np
import matplotlib.pyplot as plt
from numpy import random
from IPython.display import clear_output
# NP Necessary imports


def starformingprob(radii, theta, formingstars):
”’Function for determining the star formation rates based on probability

Inputs:
radii: array. An array of radial distances for each star. Float.
theta: array. An array of the angular distances for each star. Float.
formingstars: array. An array of indicating wheter each point is forming
stars. Boolean.

Outputs:
null. Updates the formingstars array.”’
sfr = random.random(2000)
# NP Generating 1000 numbers randomly
sto = np.array(np.repeat(0, 2000), dtype = float)
for i in range(len(sfr)-1):
        ii = ((radii[i]*np.cos(theta[i]) -radii*np.cos(theta))**2 \
                +(radii[i]*np.sin(theta[i]) -radii*np.sin(theta))**2)\
                **0.5 ¡ 1
        # NP Finding stars that are closer than 2 Mpc from each star
        if(formingstars[ii].any()):
                sto[i] = random.random()
                # NP If there is a nearby star that is actively undergoing star
                # NP formation, then there is a chance that this star will form stars
i2 = ((sfr >= 0.95) — (sto > 0.5)) & (ages < 50)
formingstars[i2] = True
# NP 5% chance that the star will randomly start forming stars and 5% chance
# NP that the star will for stars if there are nearby forming stars if the
# NP star cluster is less inactive
i3 = (ages >= 50) & (ages < 100)
formingstars[i3] = False
# NP Once the star is old enough, it dies and is unable to form new stars for a
# NP while
i4 = (ages >= 100)
ages[i4] = 0
# NP After 100 Myrs, then the star cluster restarts and can form stars again

t = 0
# NP Starting at time 0 years
timestep = 5
# NP Every loop is 5 million years
formingstars = np.repeat(False, 2000)
theta = random.random(2000) *2 *np.pi
```

```
radii = (random.random(2000)**(1/2)) *30
ages = np.repeat(0, 2000)
# NP Creating all starting information
while(t < 1000):
        young = formingstars & (ages <= 16)
        middle = formingstars & (ages < 33) & (ages > 16)
        old = formingstars & (ages >= 33) & (ages < 50)
        dead = ages >= 50
        # NP Defining classification of stars based on age
        ages[young — middle — old — dead] += timestep
        # NP Increasing age of all star groups
        clear_output(wait=True)
        # NP Clearing any previous graphics
        f = plt.figure()
        f.set_figwidth(8)
        f.set_figheight(8)
        # NP Making figure larger
        plt.polar(theta[young], radii[young], '.b', alpha = 1, label = \
        'Young stars')
        plt.polar(theta[middle], radii[middle], '.b', alpha = .6,\
        label = 'Middle stars')
        plt.polar(theta[old], radii[old], '.b', alpha = .3, label = 'Old Stars')
        plt.polar(theta[dead], radii[dead], '.k', alpha = .05, label = \
        'Dead Stars')
        plt.polar(theta[ formingstars &  dead], radii[ formingstars &  dead],\
        '.g', alpha = .3, label = 'Gas')
        # NP Plotting all stars
        plt.legend(loc = 'lower left')
        # NP Displaying legend
        plt.grid(True)
        plt.show()
        # NP Displaying graph
        t += timestep
        # NP Increasing time
        theta = theta +(1/(radii))
        # NP Increasing angular position with constant velocity
        print(str(len(theta[young])) +' young stars')
        print(str(len(theta[middle])) +' middle stars')
        print(str(len(theta[old])) +' old stars')
        print(str(len(theta[dead])) +' dead stars')
        print(str(len(theta[ formingstars &  dead])) +' nebulae')
        print(str(t)+' million years old')
        # NP Displaying relevant information for the simulation
        starformingprob(radii, theta, formingstars)
        # NP Running star formation function
        time.sleep(.001)
```

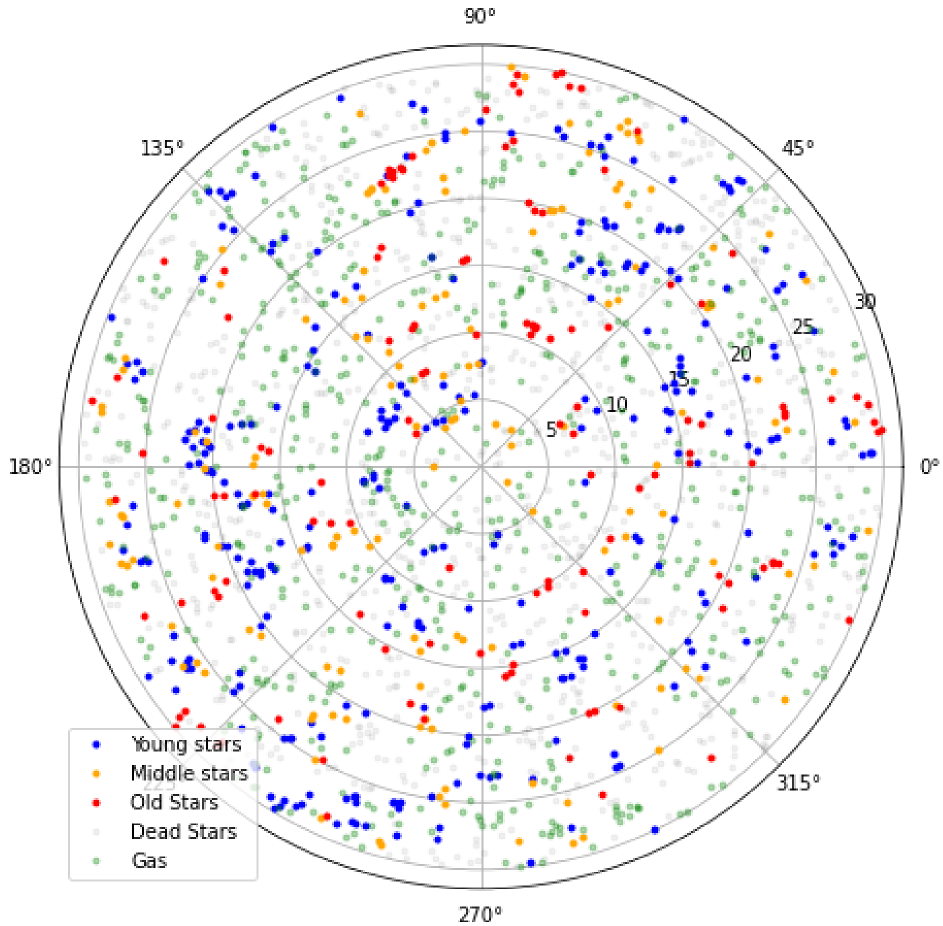
Simulation:




```
807 young stars
523 middle stars
0 old stars
0 dead stars
670 nebulae
35 million years old
```

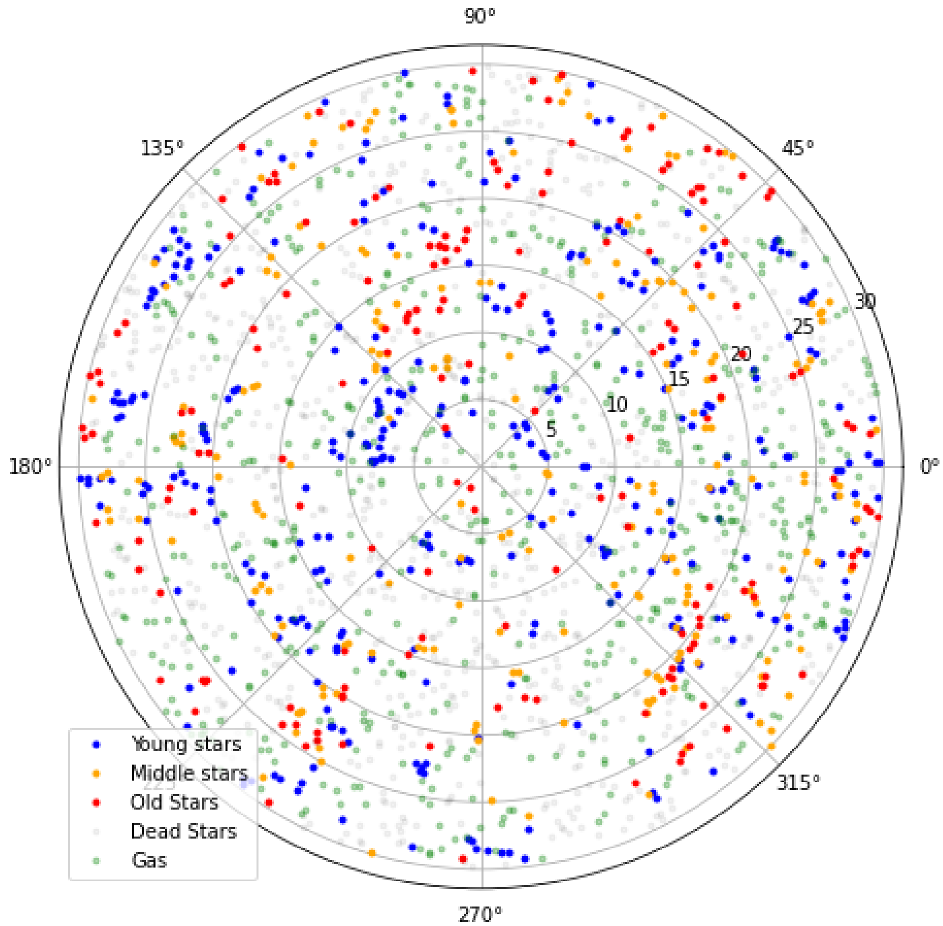
This simulation ran with 2000 stars distributed evenly over a circular plane ranging from 0 to 30 Mpc. There is a 5% chance that a cloud of gas will randomly start star formation and a 50% chance that a cloud of gas will begin star formation if there is a region of active star formation less than 1 Mpc away. The simulation runs in iterations of 5 million years up to 1 billion years (200 iterations). Each star is moving at a constant velocity and its angular position therefore changes as a function of the inverse of the radius from the center. Each star has a lifetime of 50 million years, at which point it dies and is unable to form new stars for an additional 50 million years.

In the beginning of the simulation, the stars form rapidly everywhere.



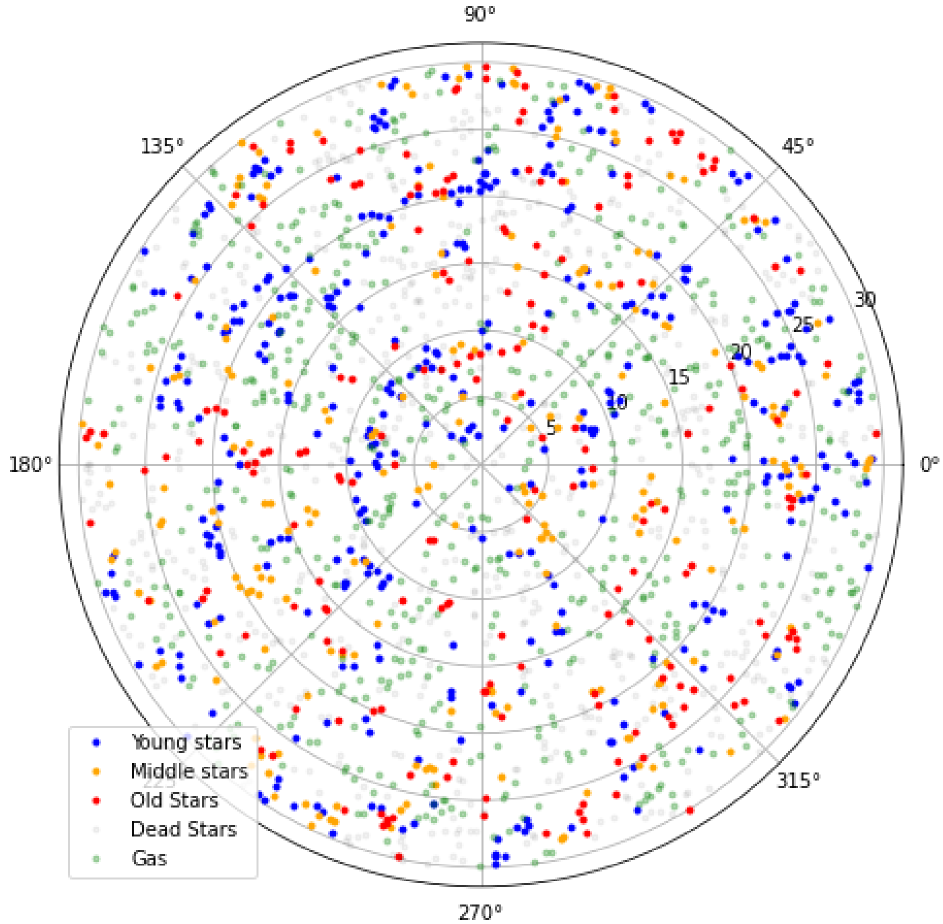

```
297 young stars
153 middle stars
128 old stars
805 dead stars
617 nebulae
500 million years old
```

After approximately 50 iterations, the simulation shows very clumpy star formation.




```
330 young stars
202 middle stars
177 old stars
761 dead stars
530 nebulae
750 million years old
```

The star formation rate in the galaxy waxes and wanes, but as time goes on, the star formation becomes more and more out of sync and star formation reaches a constant rate throughout the galaxy.




```
365 young stars
213 middle stars
197 old stars
641 dead stars
584 nebulae
1000 million years old
```

At 200 iterations, the star formation is now concentrated in distinct clumps. The star formation is evenly distributed throughout the galaxy and it is very clumpy as shown above.