Galactic Chemical Evolution Model

Name: Kunal Singh
UID: 219114

August 2022

1 Theoretical Background and Review of Relevant literature

The chemical composition of the gas changes as the gas evolves, The IGM is assumed to lead to formation of stars to due cooling and collapse of interstellar gas. Stars with mass less than 1 solar mass burns very slowly and just cut the mass of the gas from the rest of the circulation and lock it inside. The middle sized stars produce various elements and enrich the IGM through planetary nebulae and stellar winds. The white dwarfs formed temporarily locks up the gas mass but after accreting mass they explode as SNe I and prefentially produce Fe, thus further enriching the gas medium.Massive stars with mass greater than 9 solar mass results in core-collapse type 2 supernovae and enriches the galaxy with alpha type elements. To model the chemical evolution of galaxies one need following things:

1. Set initial conditions
2. Estimation of the amount of their mass stars emit throughout their evolution
3. The initial Mass Function which is an empirical function that describes the initial distribution of masses for a population of stars.
4. A model of SFR rate using the Schmidt-Kennicut Law
5. Assumptions about galactic properties like inflow and outflow rates, mixing of different phases of galaxies.

1.1 One Zone Closed model

The system is isolated with a constant total mass

$$M_g(t) + M_s(t) = M = constant \tag{1}$$

The system is well mixed at all times, i.e Z=Z(t) is the abundance of any element(s) in the gas The galaxy starts as a pure gas and no enrichment: $M_g(0) = M$ and Z(0)=0

1.1.1 Mass evolution

The total mass of the galaxy is:

$$M = M_g + M_s \tag{2}$$

M is also called Baryonic Mass. M_g is the Mass of gas and M_s is the Stellar Mass. In Closed Box approximation, there are no inflows and outflows therefore,

$$dM_g/dt = \mathcal{E} - \Psi(t) \tag{3}$$

$$dM_s/dt = \Psi(t) - \mathcal{E} \tag{4}$$

$$dM/dt = 0 \tag{5}$$

We will use Schmidth-Kennicutt relation for star formation rate Ψ which depends on M_g, $\mathcal{E}$ is the ejection rate of matter from stars. If a star with initial mass m has lifetime of $\tau(m)$ and leaves behind a remnant of mass m_{rem} then the mass returned in ISM at a time tau is $m - m_{rem}$. In our program we use the initial final mass relation given by [3] The number of stars of mass m formed at time $t - \tau$ is given by the product of star formation rate $Psi(t - \tau(m))$ and the initial mass function $phi(m)$. Hence we get,

$$\mathcal{E}(t) = \int_{m_{tau=t}}^{m_U} (m - m_{rem})\Psi[t - \tau(m)]\phi(m)dm \tag{6}$$

here the lower limit corresponds to stars that have a life time $t = \tau$
and the upper limit is assumed to be 100 $M_\odot$.

1.1.2 Chemical enrichment

The equation for the change in the quantity of metal present in gas(for a closed box model) is:

$$dM_z/dt = dM_g/dt = \mathcal{E}_z - Z\Psi \tag{7}$$

here, $\mathcal{E}$ is the rate at which metals are added to the gas phase after been produced in stars
$Z\Psi$ accounts for the metals that are being incorporated in new star via star formation rate We can use an equation similar to 6 for $\mathcal{E}$:

$$\mathcal{E}_z(t) = \int_{m_{tau=t}}^{m_U} (m - m_{rem})Z(t - \tau(m)) + mq_z(m)\Psi[t - \tau(m)]\phi(m)dm \tag{8}$$

Here, $q_z(m)$ is the mass fraction of metals in the gas from the star The enrichment of the gas due to Type I SNe can be simplified by assuming that only a fraction of all stars with mass less than $8M_\odot$ explode as a Type I SNe and all Type I SNe produce the same amount of enrichment. As we have already accounted for return of unenriched gas in 8, we write:

$$\mathcal{E}_{Z,I_a}(t) = \eta_{Ia} \int_{m_{\tau=t+t_{delay}}}^{8M_\odot} m_z\Psi(t - \tau(m) - t_{delay})\phi(m)dm \tag{9}$$

η is the efficiency parameter which specifies the fraction of stars below $8M_\odot$ that explode as Type Ia SNe. η is chosen somewhere between 0.05 and 0.09[4]

1.2 Review of Relevant literature:

The IFMR relation was used from *Lawlor & MacDonald (2006)*[3]. The Schmidth-Kennicutt relation,the table for Stellar lifetimes and the Salpeter and kroupa IMF was referred from [4].
Woosley and Weaver(1995)[5] was referred to get the table of yields for Type II SNe.

2 Statement of Problem

Analysing Metal abundancies using closed box one-zone Galactic chemical Evolution Model

3 Details of Study

3.1 Important formulae and functions

3.1.1 SFR

A function to calculate the SFR was made using the Schmidt-Kennicut relation from [1]

$$\Sigma_{SFR} = 2.5 * 10^{-4}(\frac{\Sigma_{gas}}{1M_\odot pc^{-2}})^{1.4} M_\odot yr^{-1}kpc^{-2} \tag{10}$$

where Σ_{SFR} and Σ_{gas} are the SFR and gas surface densities, respectively and Σ_{gas} is in $1M_\odot pc^{-2}$

3.1.2 IMF and mass remnant

A function to calculate IMF has been defined by referring to [2]

$$\psi(m) = m^{-\alpha} \tag{11}$$

where,
$\alpha = 0.3$for $m < 0.08$
$\alpha = 1.3$for $0.08 < m < 0.5$
$\alpha = 2.3$for $m > 0.5$ A function to calculate Stellar mass remnant is made by referring to *Iben and Tutukov 1984b*:

$$m_{rem} = \begin{cases} 0.11m + 0.45, & m \leq 9.5 \\ 1.5, & m > 9.5 \end{cases} \tag{12}$$

3.1.3 Initial final mass relation(IFMR)

The IFMR table used from [3] is listed below. A function was made by interpolating the data (with scipy) to predict the final mass for the given initial mass.

initial mass	Final mass
0.8	0.54
0.85	0.54
0.9	0.54
0.95	0.54
1.0	0.541
1.2	0.557
1.5	0.563
2.0	0.595
2.5	0.632
3.0	0.678
4.0	0.778
5.0	0.874
6.5	0.973
8.0	1.08
10.0	1.11

Table 1: IFMR

3.1.4 Type II SNe yield

The yields table used from [5] is listed below. A function was made by interpolating the data (with scipy) to predict the yield of required metal for the given Stellar mass.

Stellar mass	16O yield in Solar Mass	Fe yield in Solar Mass
11.0	0.136	0.0804
12.0	0.21	0.0554
13.0	0.272	0.1458
15.0	0.68	0.1297
18.0	1.13	0.0828
19.0	1.43	0.1177
20.0	1.94	0.1063
22.0	2.38	0.2247
25.0	3.25	0.1509
30.0	3.65	0.318
35.0	3.07	0.0918
40.0	2.36	0.051

Table 2: Yields

3.1.5 Assumptions and numerical integration:

The initial surface gas density M_g0 is 10 $M_{\odot}pc^{-2}$ and initial stellar surface density M_s0 is 0 $M_{\odot}pc^{-2}$. The initial metallicity in the gas is zero The model will run for 10^10yr We first numerically integrate the ejection rates from eqn 8 and 6 using Euler's Method and then use them for mass evolution and metal enrichment equations viz. 3, 4 and 7 Hence we obtain the time evolution of Mass and Metal enrichment in the Interstellar gas.

3.2 Main Program

```
import numpy as np
import scipy.interpolate as ip

minit = [0.80, 0.85, 0.90, 0.95, 1.000, 1.200, 1.500, 2.000, 2.500, 3.000, 4.000,\
         5.000, 6.500, 8.000, 10.0]
mfin  = [0.54, 0.54, 0.54, 0.54, 0.541, 0.557, 0.563, 0.595, 0.632, 0.678, 0.778,\
         0.874, 0.973, 1.080, 1.11]
fmfin = ip.interp1d(minit, mfin) #interpolating to form a function to find final mass

#Yields from type 2 Super Novae in terms of Solar Mass
msn2 = [11.0, 12.0, 13.0, 15.0, 18.0, 19.0, 20.0, 22.0, 25.0, 30.0, 35.0, 40.0]
m16O = [1.36e-01, 2.10e-01, 2.72e-01, 6.80e-01, 1.13, 1.43, 1.94, 2.38, 3.25,\
        3.65, 3.07, 2.36]#yield for 16O
mFe=[1.11e-02,1.22e-02,1.28e-02,1.47e-02,1.71e-02,1.77e-02,1.83e-02,1.97e-02,\
     2.19e-02,2.50e-02,2.77e-02,2.83e-02] #yield for Fe

f16O = ip.interp1d(msn2,m16O)#interpolating to form a function to find the yield for 16O
fFe  = ip.interp1d(msn2,mFe)#interpolating to form a function to find the yield for Fe



def calcsfr(mg):
#Calculating SFR using Schmidt-Kennicutt
    SFrate = 1e-10*2.5*mg**1.4
    return SFrate

def mrem(ms):
#function computes mass remnant for initial stellar mass m
   if ms < 9.5:
      mlast = 0.11*ms + 0.45 #lower mass stars
   else:
      mlast= 1.5
   return mlast

def fq16O(ms):

    if ((ms > min(msn2)) and (ms < max(sn2))):
        return f16O(m)/m
    else:
        return 0.

def fqFe(ms):

    if min(msn2)<ms<max(msn2):
        return fFe(ms)/ms
    else:
        return 0.

#Function finds the IMF based on kroupa(2001)
def IMF(mg):

  mmin = 0.15
  mmax = 100
  if mg<0.08:
    alpha = -0.3
  elif mg>0.08 and mg<0.5:
    alpha=-1.3
  elif m>0.5:
```

```
    alpha=-2.3
  return mg**alpha * -(alpha+2) / (mmin**(alpha+2) - mmax**(alpha+2))




def mej(ti, at, aSFR):
# Calculate the ejected mass, from stellar evolution, at time ti.
# at and aSFR are arrays containing the time and star formation rate.
# Stellar masses and lifetimes are read from the file t_m.txt

    logt, m = np.loadtxt('/content/drive/MyDrive/GCE/t_m.txt',usecols=(0,1), unpack=True)
    t = 10**logt
    fm = ip.interp1d(t, m)
    ft = ip.interp1d(np.flipud(m), np.flipud(t))

    if (ti < min(t)):        # No stars have evolved off MS yet
        ee = 0.
    elif (len(at) < 2):
        ee = 0.
    else:
        fSFR = ip.interp1d(at, aSFR)      # Interpolate in (t, SFR) arrays
        mt = fm(ti)                       # Lower limit of the integral
        mU = max(m)                       # Upper limit
        mm = mt
        ee = 0.
        while (mm < mU):
            dm = mm / 25.
            tSFR = max(ti - ft(mm),0)
                                              # m was born
            if (tSFR > max(at)): tSFR = max(at)
            de = (mm - mrem(mm)) * fSFR(tSFR) * IMF(mm) * dm  # ejected mass
            ee = ee + de
            mm = mm + dm

    return ee


def mejZ_SNIa(ti, at, aSFR, mZ):

# mZ is the mass of the element produced by one SN explosion

    mSNIamax = 8.0
    fSNIa    = 0.02
    delay = 1e9       # Time delay, after endpoint of normal stellar evolution

    logt, m = np.loadtxt('/content/drive/MyDrive/GCE/t_m.txt',usecols=(0,1), unpack=True)
    t = 10**logt + 4*delay      # Remember to add the delay
    fm = ip.interp1d(t, m)
    ft = ip.interp1d(np.flipud(m), np.flipud(t))
    w = np.where(m < mSNIamax)

    if (ti < min(t[w])):
        ee = 0.
    elif (len(at) < 2):
        ee = 0.
    else:
        fSFR = ip.interp1d(at, aSFR)
        mt = fm(ti)
        mm = mt
        ee = 0.
```

```
        while (mm < mSNIamax):
            dm = mm / 25.
            tSFR = max(ti - ft(mm),0)
            if (tSFR > max(at)): tSFR = max(at)
            de = fSNIa * mZ * fSFR(tSFR) * IMF(mm) * dm
            ee = ee + de
            mm = mm + dm

    return ee


def mejZ(ti, at, aSFR, aZ, fqZ):

    logt, m = np.loadtxt('/content/drive/MyDrive/GCE/t_m.txt',usecols=(0,1), unpack=True)
    t = 10**logt
    fm = ip.interp1d(t, m)
    ft = ip.interp1d(np.flipud(m), np.flipud(t))

    if (ti < min(t)):
        ee = 0.
    elif (len(at) < 2):
        ee = 0.
    else:
        fSFR = ip.interp1d(at, aSFR)
        fZ = ip.interp1d(at, aZ)
        mt = fm(ti)
        mU = max(m)
        mm = mt
        ee = 0.
        while (mm < mU):
            dm = mm / 50.
            tSFR = max(ti - ft(mm),0)
            if (tSFR > max(at)): tSFR = max(at)
            de = ((mm - mrem(mm))*fZ(tSFR) + mm*fqZ(mm)) * fSFR(tSFR) * IMF(mm) * dm
            ee = ee + de
            mm = mm + dm

    return ee

Tarr=[]
def gce_model(tmax=1.0e10, Mg0=10., Ms0=0.):
    t = 0.
    Mg = Mg0
    Ms = Ms0
    M16O = 0.
    MFe  =  0


# Arrays to store the enrichment history and related variables
    at, aMg, aMs, aSFR, aZ16O, aZFe = [], [], [], [], [], []

    while (t < tmax):

        dt = 1e6 + t/1e2      # Time steps: minimum 1e6 yr>>1% of age

        SFR = calcsfr(Mg)
        et  = mej(t, at, aSFR)

        e16O = mejZ(t, at, aSFR, aZ16O, fq16O) # O from Type II SNe
        Z16O = M16O/Mg
```

```
        eFeI = mejZ_SNIa(t, at, aSFR, 0.61)     # Fe from Type Ia SNe
        eFeII = mejZ(t, at, aSFR, aZFe, fqFe)  # Fe from Type II SNe
        eFe = eFeI + eFeII
        ZFe = MFe/Mg




        Z16Oej = Z16O       # Assume that ejected gas has same composition
                            # as current gas composition
        ZFeej = ZFe
        # rwind = 0  # No Winds
#numerically integrating using Euler Method
# Update variables
        dMg = (et - SFR) * dt
        dMs = (SFR - et) * dt
        dM16O = (e16O - Z16O*SFR ) * dt
        dMFe  = (eFe  -  ZFe*SFR) * dt
        Mg = Mg + dMg
        Ms = Ms + dMs
        M16O = M16O + dM16O
        MFe  = MFe  + dMFe

        at.append(t)
        aSFR.append(SFR)
        aMg.append(Mg)
        aMs.append(Ms)
        aZ16O.append(M16O/Mg)
        aZFe.append(MFe/Mg)

        table=[t, Mg, Ms, SFR, et, e16O, Z16O, eFeI, eFeII, ZFe]
        Tarr.append(table)
        t = t + dt
    return Tarr,aZFe,aZ16O,aMg,aSFR,at,aMs

#run the function
import matplotlib.pyplot as plt

A,aZFe,aZ16O,aMg,aSFR,at,aMs=gce_model()
A=np.array(A)
```

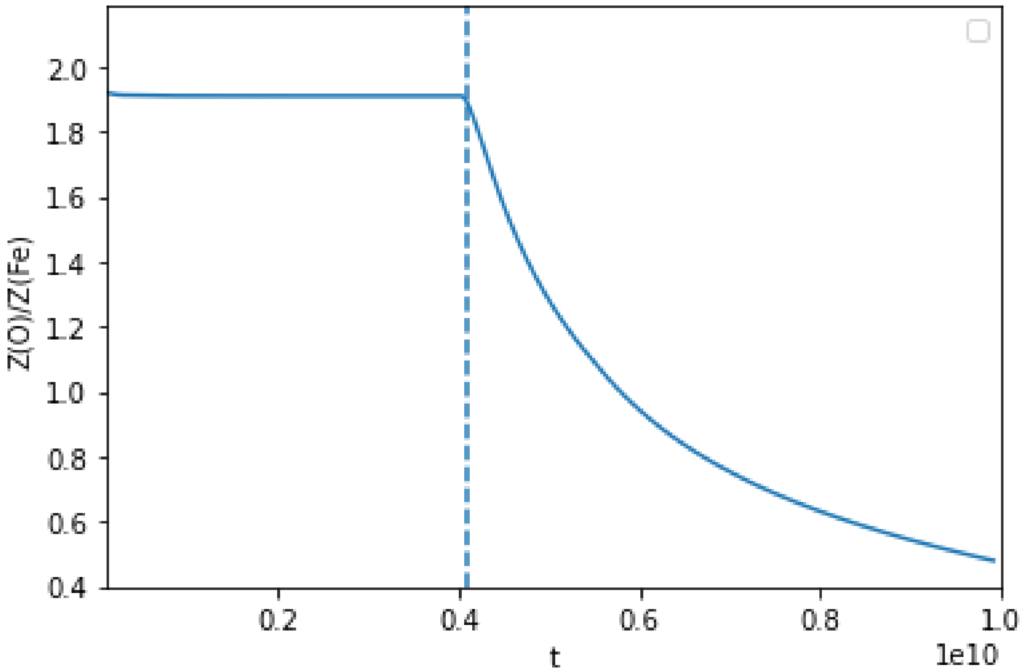
Program for plots:

```
#Plots
plt.plot(at,np.log10(np.array(np.array(aZFe))),lw=2,label='Fe')
plt.plot(at,np.log10(np.array(aZ16O)),lw=2,label='$ ^{16} O$')
plt.xlabel('Z')
plt.ylabel('t')
plt.axvline(x=0.41*1e10,linestyle='--')
plt.legend()
plt.figure()
plt.plot(at,np.log10(np.array(aZ16O)/np.array(aZFe)))
plt.ylabel('Z(O)/Z(Fe)')
plt.xlabel('np.log10(Z(Fe))')
plt.xlim(0.01*1e10,1.0*1e10)
plt.axvline(x=0.41*1e10,linestyle='--')
plt.legend()
plt.plot(at,aSFR)
```

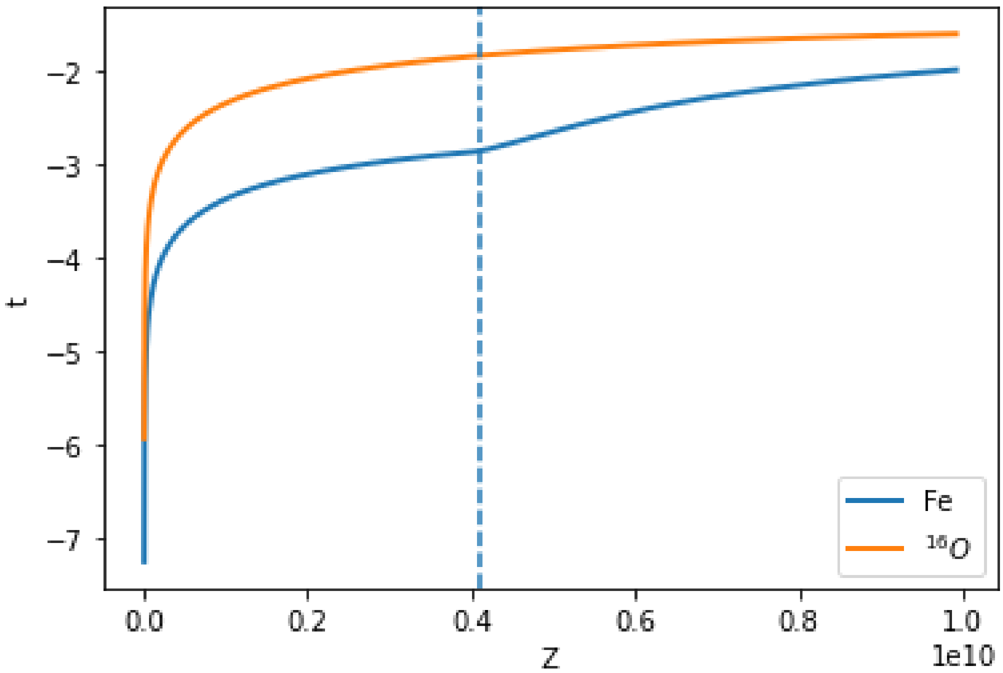
```
plt.xlabel('SFR')
plt.ylabel('time')

plt.title('Change in star formation rate')
plt.figure()
plt.plot(at,aMg,label='Gas mass surface density')
plt.title('Change in Stellar mass and gas Mass surface density')
plt.plot(at,np.array(aMg)+np.array(aMs),label='Baryonic Mass')
plt.plot(at,aMs,label='Stellar mass surface density')
plt.xlabel('mass in Msun')
plt.ylabel('time')
plt.legend()
```

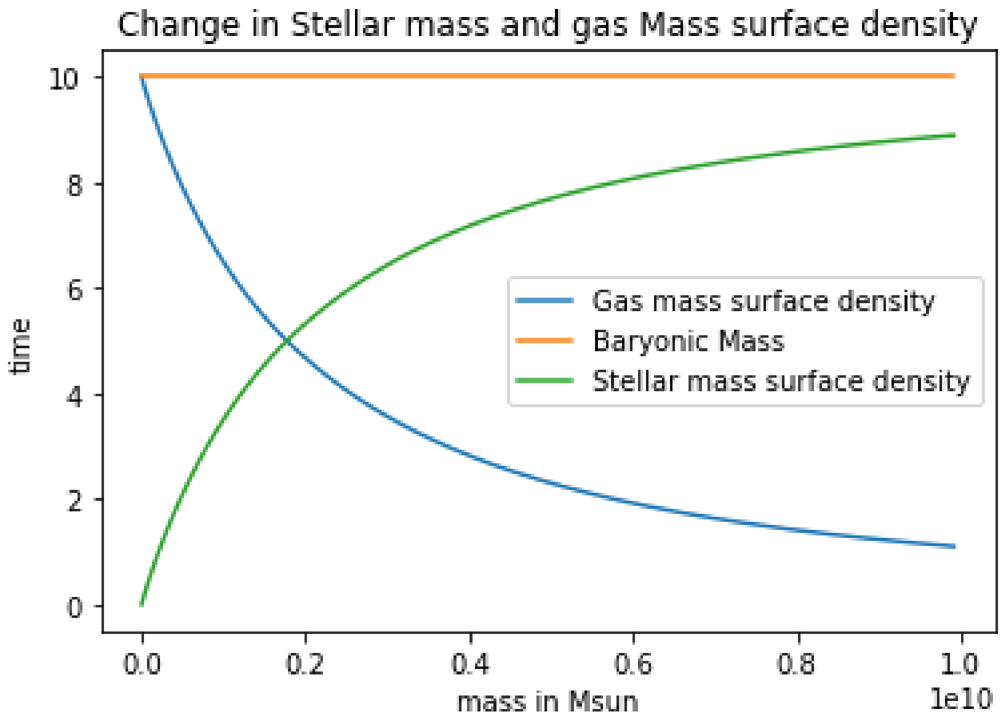
4 Results



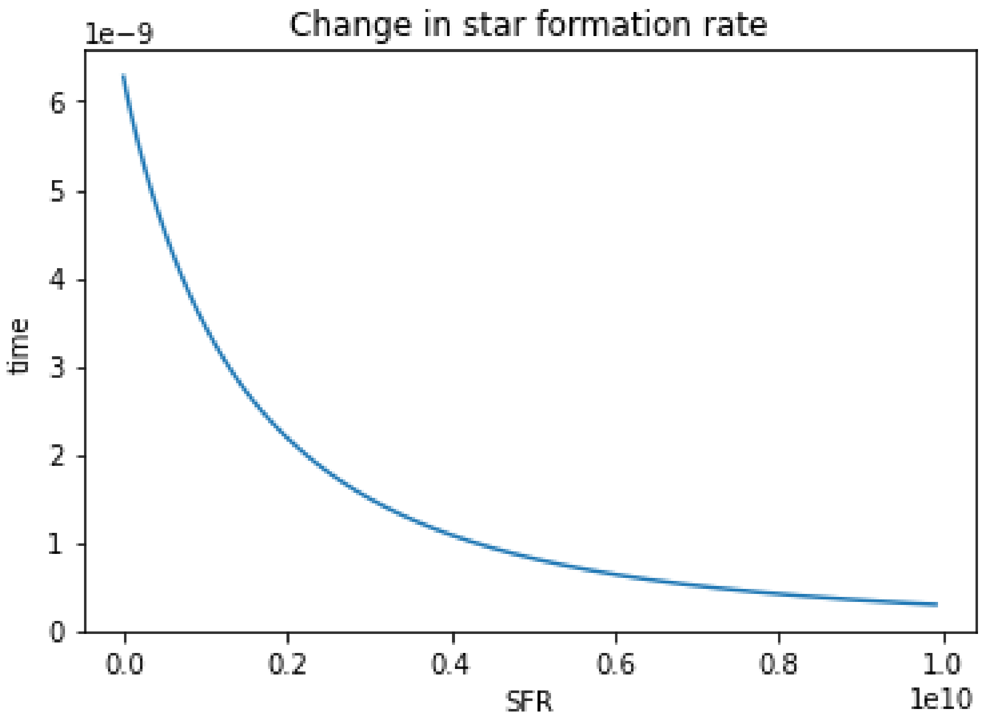
The “knee” in this plot corresponds to the level of iron enrichment that had been reached when Type Ia SNe started to contribute. The ISM of the early universe was enriched primarily by type II SNe but the present ISM has Fe abundance due to both SNe type I and II, as Type I SNe start after a delay



Initially,Type II SNe produce Alpha-elements (and some iron). After t=0.4e10 years Type Ia SNe kick in and produce extra iron



As the stars starting forming, the gas mass starts decreasing and the stellar mass starts increasing. As this is a closed model, the Baryonic mass which is the sum of stellar mass and gas mass remains constant.



We can notice that the star formation rate keeps on decreasing as the galaxy ages. It fits the observations that as the mass of gas decreases as the galaxy evolves the star formation rate decreases.

5 Limitations

1. The galaxies accrete and lose mass throughout their evolution, in our closed box model we have neglected that part entirely accounting for inflow and outflow of mass will give a more accurate model.

2. We have considered the one zone model which assumes that the entire galaxy is in the same phase, but in reality galaxies can be consists different zones.

6 Relevance of the Study

1. Even though approximations are made in the model for simplifications it still captures a realistic trends of the galactic evolution.

2. By studying such models and comparing it with different models and observational data we can gain a better understanding of the Galaxy evolution.

References

[1] Jr. Kennicutt, Robert C. The Global Schmidt Law in Star-forming Galaxies. , 498(2):541–552, May 1998.

[2] Pavel Kroupa. On the variation of the initial mass function. , 322(2):231–246, April 2001.

[3] T. M. Lawlor and J. MacDonald. The mass of helium in white dwarf stars and the formation and evolution of hydrogen-deficient post-AGB stars. *Monthly Notices of the Royal Astronomical Society*, 371(1):263–282, 08 2006.

[4] Bernard E. J. Pagel. *Nucleosynthesis and Chemical Evolution of Galaxies.* Cambridge University Press, 2 edition, 2009.

[5] S. E. Woosley and Thomas A. Weaver. The Evolution and Explosion of Massive Stars. II. Explosive Hydrodynamics and Nucleosynthesis. , 101:181, November 1995.