

```
In [1]: import numpy as np
from scipy.integrate import odeint
%matplotlib inline
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
import sys
```

```
In [2]: t = np.linspace(0,50395, num=10080)
lightdata = np.transpose(np.delete(np.genfromtxt('AllLightnosmooth.csv', delimiter=','),

def func(t):
    z0 = [1,0,0,0,0,0]
    #mCherry2 = np.empty((len(lightdata[:,0]), len(t)))
    out = np.empty((len(lightdata[:,0]),len(t),6))
    arrayvalues = np.asarray([])
    #for i in range(42):
    for i in range(len(lightdata[:,0])):
        def I(t):
            tindex = t/5
            if tindex > 10079:
                tindex = 10079
            return lightdata[i][int(tindex)]

        def model(z,t):
            d1 = 0.01844
            k1 = 0.082854
            d2 = 0.240942
            k2 = 0.00171
            Kd = 96.81974
            n = 0.662886
            k3 = 0.000519
            d3 = 0.000544
            k4 = 1.25
            d4 = 0.0000924
            k5 = 0.00144

            Pu = z[0]
            Pb = z[1]
            Pa = z[2]
            mRNA = z[3]
            mCherry1 = z[4]
            mCherry2 = z[5]

            dPudt = d1*Pb - k1*I(t)**n/(Kd**n+I(t)**n)*Pu
            dPbdt = k1*I(t)**n/(Kd**n+I(t)**n)*Pu + d2*Pa - d1*Pb - k2*Pb
            dPadt = k2*Pb - d2*Pa
            dmRNA dt = k3*Pa - d3*mRNA
            dmCherry1dt = k4*mRNA-(d4 + k5)*mCherry1
            dmCherry2dt = k5*mCherry1-d4*mCherry2
            return [dPudt,dPbdt,dPadt,dmRNA dt,dmCherry1dt,dmCherry2dt]

        z = odeint(model,z0,t)
        #mCherry2[i] = z[:,5]
        out[i] = z
    return out
```

```

In [3]: model1 = np.asarray(func(t))
        mCherry2 = model1[:, :, 5]
        Pu = model1[:, :, 0]
        Pb = model1[:, :, 1]
        Pa = model1[:, :, 2]
        mRNA = model1[:, :, 3]

        #total = Pu+Pb+Pa+Pi
        #print(total)

        #for i in range(42):
        for i in range(len(lightdata[:, 0])):
            last = mCherry2[i]
            ##total = Pi[i]+Pu[i]+Pb[i]+Pa[i]
            print(last[-1])

```

```

41.17686226716469
20.24710640564201
10.930911025234339
0.08437443531981269
0.0008482944249234306
46.21271096932903
24.118579456029
12.147650038928507
0.001054463708083116
52.23091089378779
29.41185146008409
16.61055014148735
5.632074833346439
0.00426518618515847
0.004364404427776982
53.576556930349895
30.710385372012794
17.731737263093624
0.0014497948991095207
0.0014498013294926629
35.031826032642236
11.987949735759258
5.996832205542819
2.003135480714289
0.504344556676506
38.439909740946604
13.175518559599382
6.590839293280174
2.2014322794048917
0.038599005477463325
0.4026586492777783
42.288430601287466
14.519083354159488
7.262857631580353
2.4257411236288955
0.8071849236348283
0.05052859532146162
43.11753702921904
14.808875394531546
7.407802508777846
2.4741171547094023
0.04337709652530454
39.128481181282034
29.371710399187116
9.732079094186968
4.419201771527629
1.159646403059794
0.04140974808894174

```

42.41278372626423
31.83752671091111
10.548558615338402
4.789713050106694
1.2568483564979434
0.04488070785611775
46.05496082730935
34.572101552695976
11.453929102070251
5.200518787002526
1.3646170616120439
0.04872899745577272
46.830758390347874
35.15458547690946
11.646765856561917
5.288011441219499
1.3875688067725334
0.04954856560173404
48.45214161648883
28.477378990465365
13.333942014851088
3.5414742361596905
1.4245763478196585
0.1264935136211393
52.36690384490537
30.777353593449863
14.4099619936919
3.82717297992565
1.5394994409193035
0.13669797605540687
56.683954322772514
33.31352164498809
15.59633002734296
4.142155171290812
1.6662017398324305
0.14794829774923926
57.60026729331739
33.851812862039196
15.848110580270694
4.20900128801384
1.6930907094155772
0.15033588160308728
52.0680430595039
36.53625328045639
27.878866765131324
7.643582419446352
3.076589193586688
0.2731863857593181
56.231144559229136
39.456035972824964
30.105965852485657
8.253920195950354
3.3222518818027225
0.29500004479416714
60.8147900849642
42.67051601400487
32.55770801050618
8.925776321269675
3.5926754618815138
0.3190123679026818
61.786714741634846
43.352086434195485
33.07753534115101
9.068219496025568
3.6500091490494353

0.32410332080765625
 55.38048542349211
 59.76346693310007
 64.58168665013828
 65.60234257639438

```
In [5]: import csv
pp = PdfPages('multipage.pdf')

#for i in range(42):
for i in range(len(lighdata[:,0])):
#    mCherry2 = model1[:, :, 6]
#    with open('dataout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in mCherry2:
#            csvwriter.writerow(row)
#    Pu = model1[:, :, 0]
#    with open('Puout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in Pu:
#            csvwriter.writerow(row)
#    Pb = model1[:, :, 1]
#    with open('Pbout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in Pb:
#            csvwriter.writerow(row)
#    Pi = model1[:, :, 2]
#    with open('Piout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in Pi:
#            csvwriter.writerow(row)
#    Pa = model1[:, :, 3]
#    with open('Paout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in Pa:
#            csvwriter.writerow(row)
#    mRNA = model1[:, :, 4]
#    with open('mRNAout.csv', 'w') as csvfile:
#        csvwriter=csv.writer(csvfile)
#        for row in mRNA:
#            csvwriter.writerow(row)

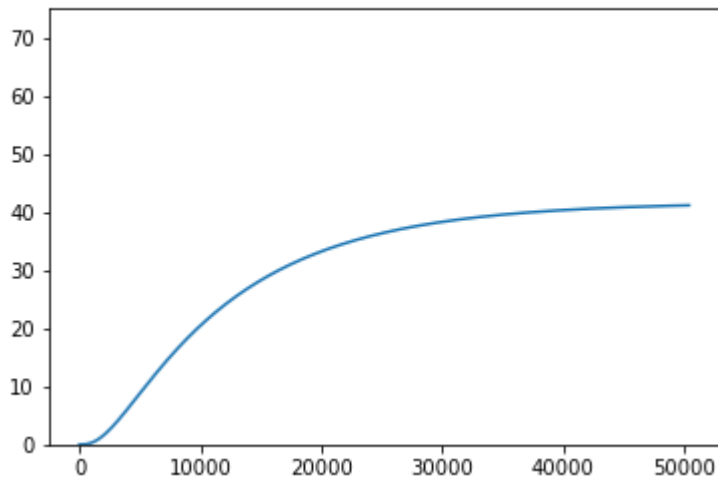
#U=U*100;
#Pu=Pu*100;
#Pb=Pb*100;
#Pa=Pa*100;
#t = np.linspace(0, 50395, num=10080)
#plt.plot(t, U[i])
#plt.plot(t, Pu[i])
#plt.plot(t, Pb[i])
#plt.plot(t, Pa[i])
#plt.legend(['U', 'Pu', 'Pb', 'Pa'])
#plt.plot(t, lighdata[i])
plt.plot(t, mCherry2[i])
plt.xlabel('Time [s]')
plt.ylabel('Fluoresence [a.u.]')
plt.suptitle('condition '+str(i+1), fontsize = 14)
plt.ylim((0, 75))

#def annot_max (t, mCherry2, ax=None):
#    tmax = t[np.argmax(mCherry2[i])]
```

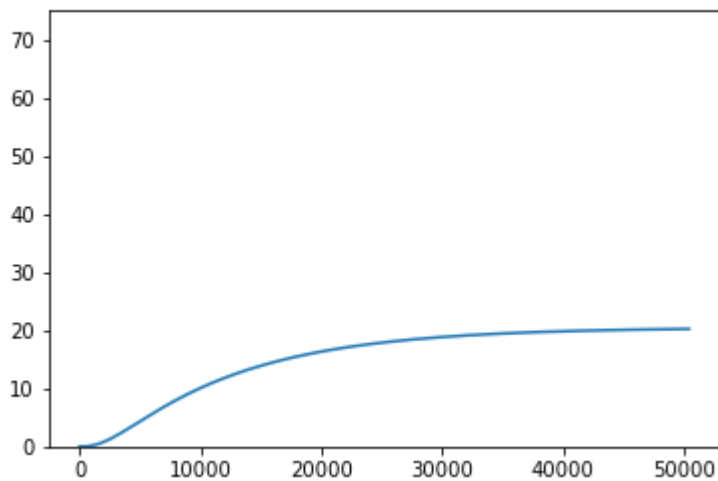
```
# ymax = mCherry2[i].max()
# text = "t={:.3f}, y={:.3f}".format(tmax,ymax)
# print(i+1, ymax)
# if not ax:
#     ax=plt.gca()
#     bbox_props = dict(boxstyle = "square,pad=0.3", fc="w", ec="k",lw=0.72)
#     arrowprops = dict(arrowstyle="->", connectionstyle = "angle,angleA=0,angleB=60")
#     kw = dict(xycoords='data', textcoords="axes fraction", arrowprops=arrowprops,
#               ax.annotate(text,xy=(tmax,ymax), xytext=(0.94,0.7), **kw)
# #annot_max(t,mCherry2)
pp.savefig()
plt.show()

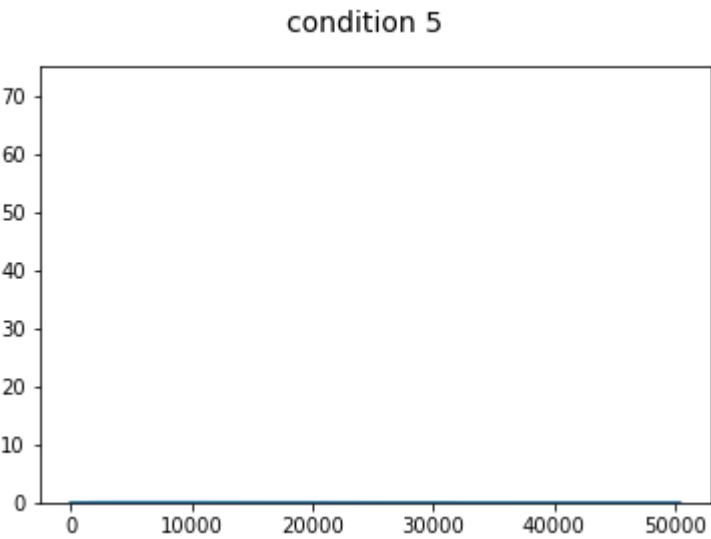
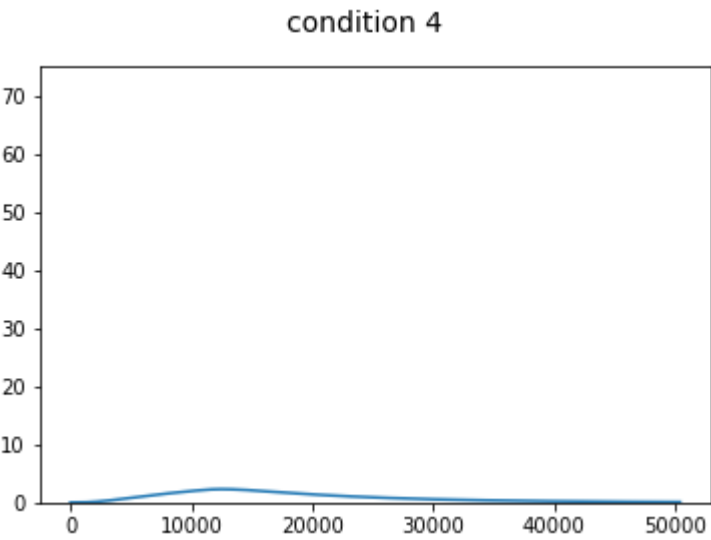
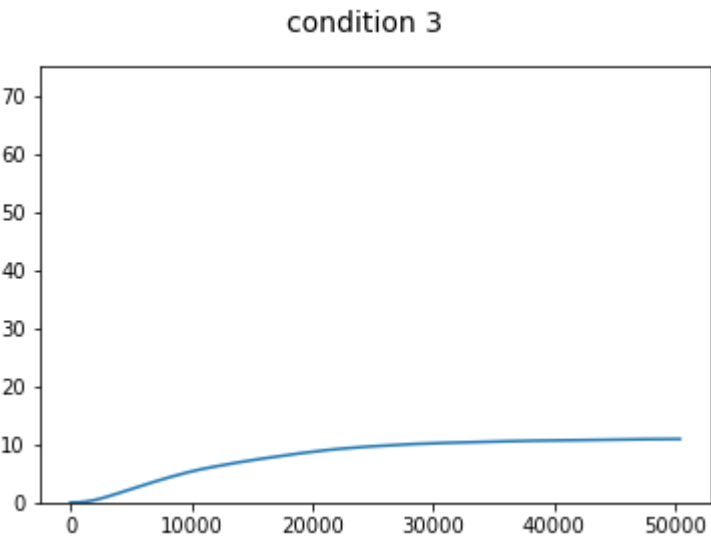
pp.close()
```

condition 1

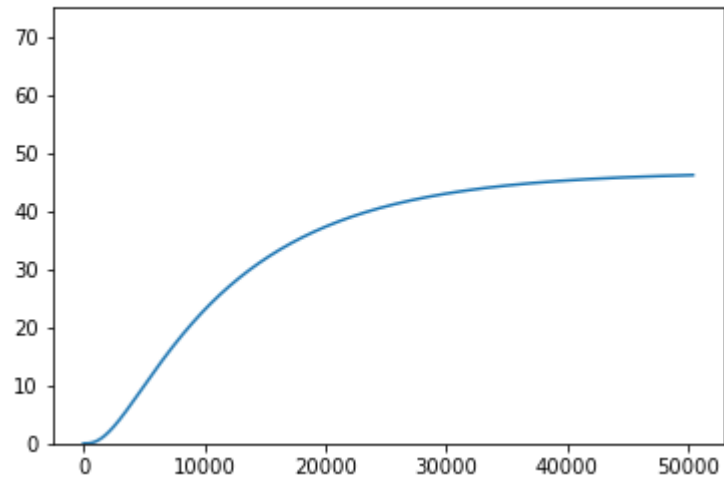


condition 2

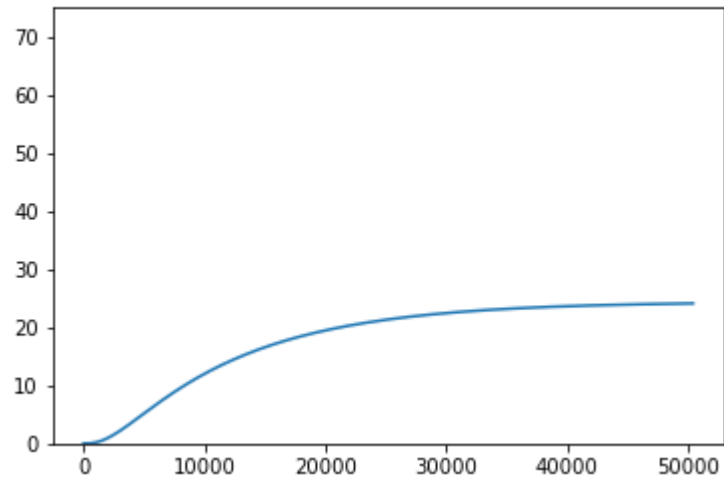




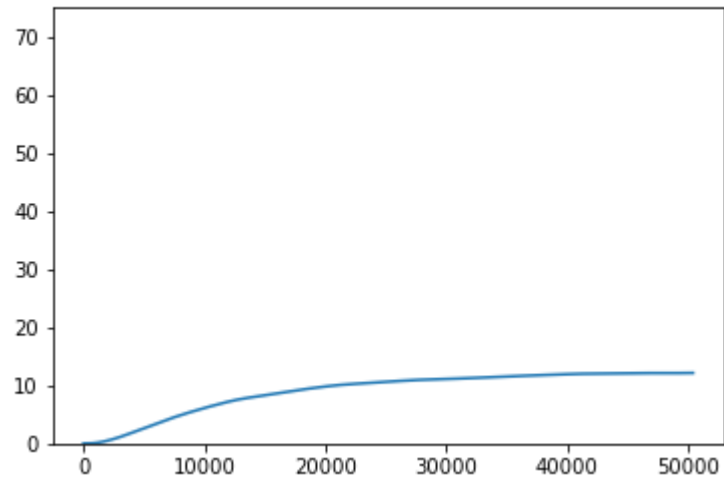
condition 6



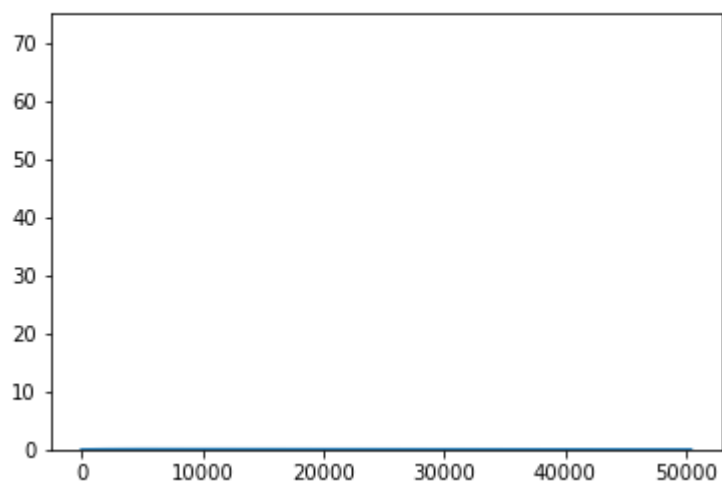
condition 7



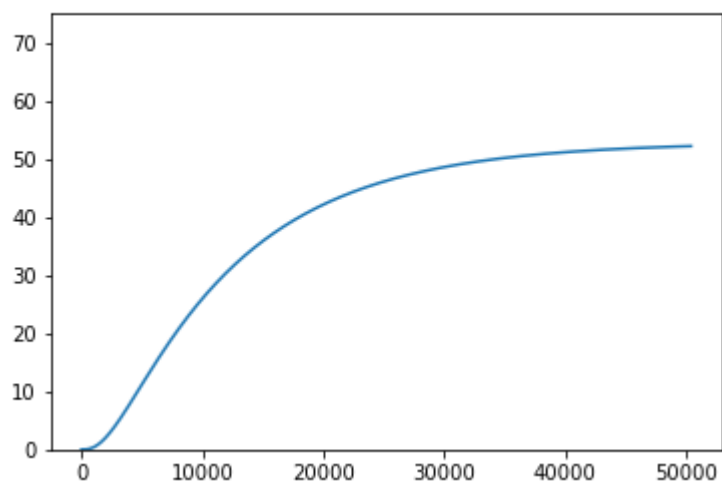
condition 8



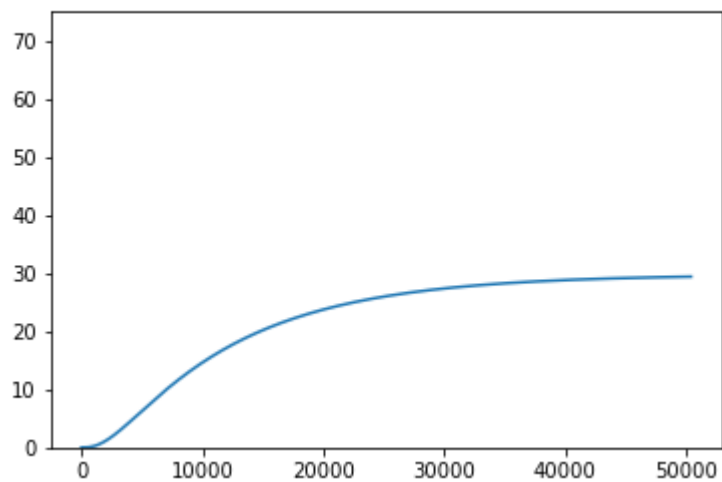
condition 9



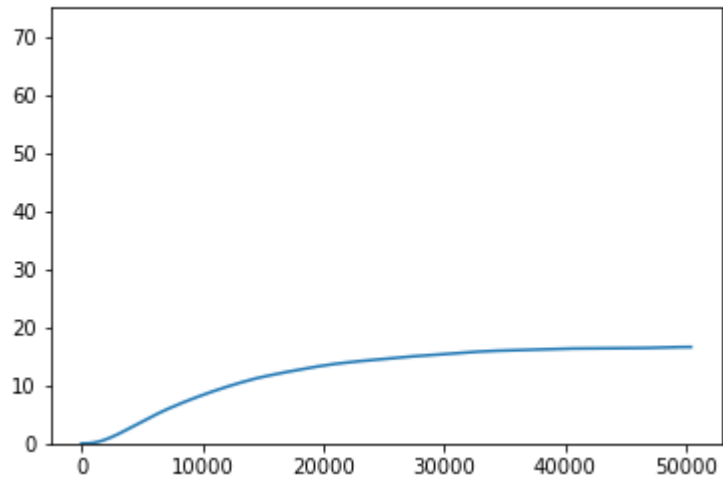
condition 10



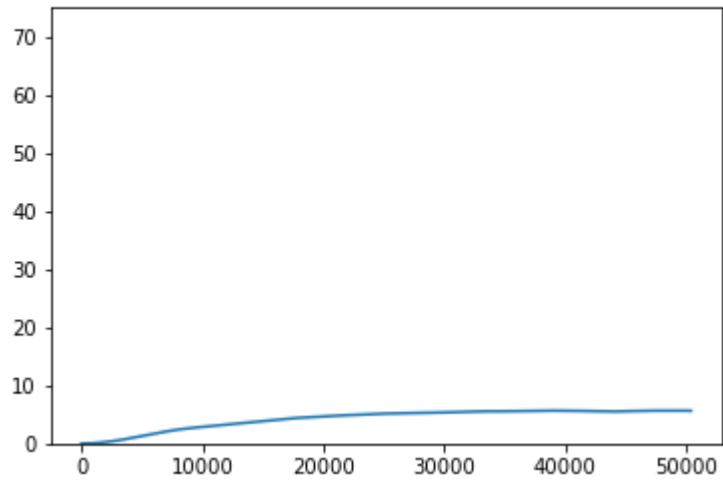
condition 11



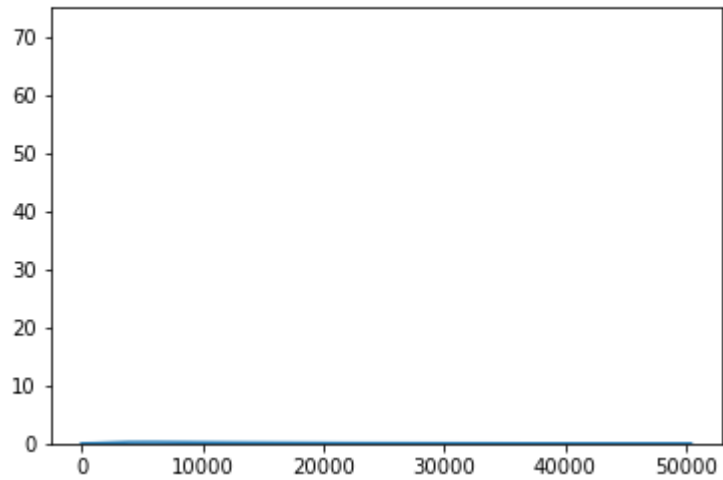
condition 12



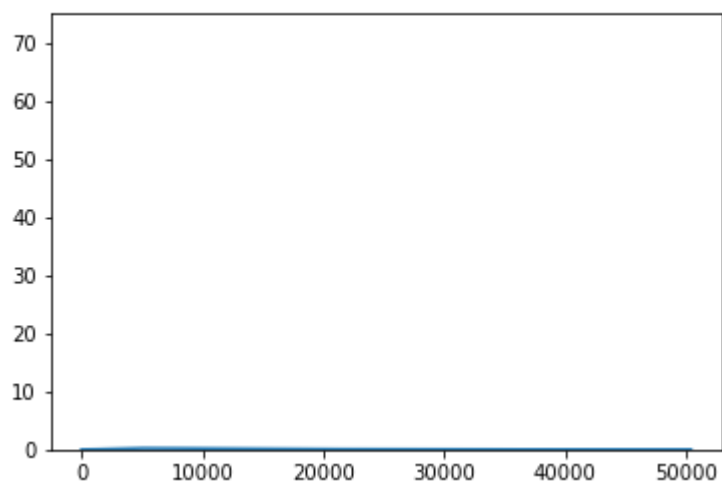
condition 13



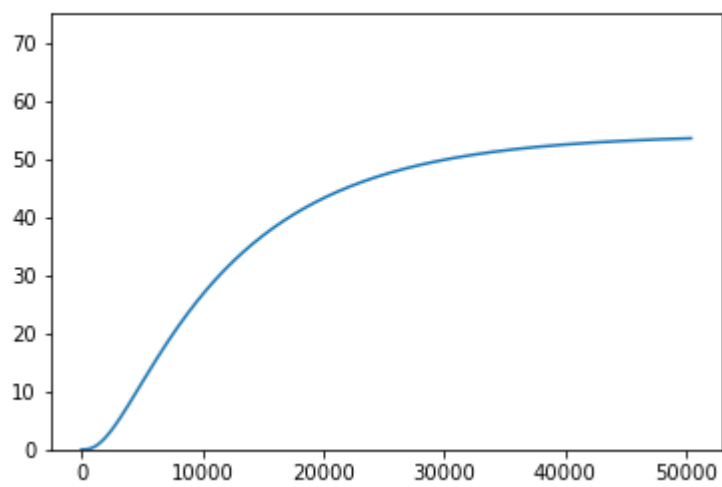
condition 14



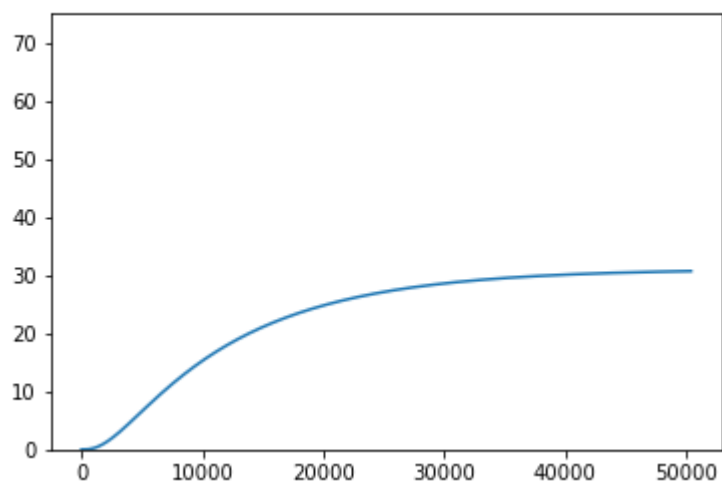
condition 15



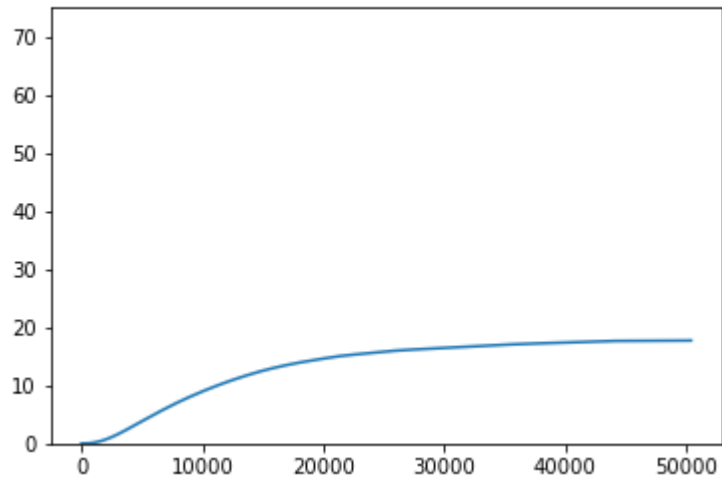
condition 16



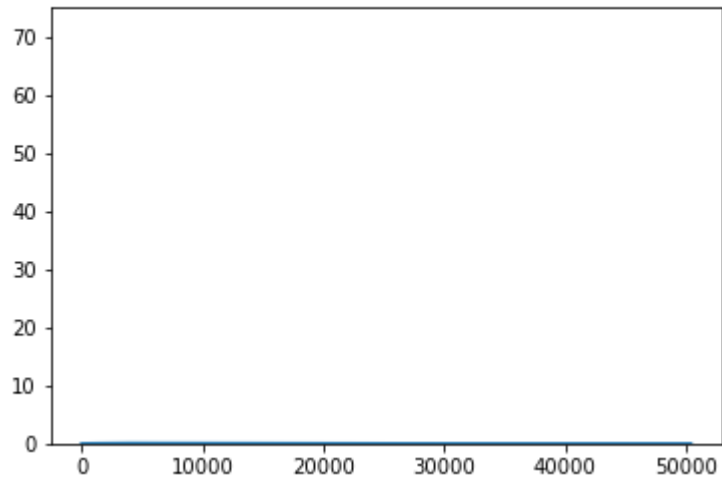
condition 17



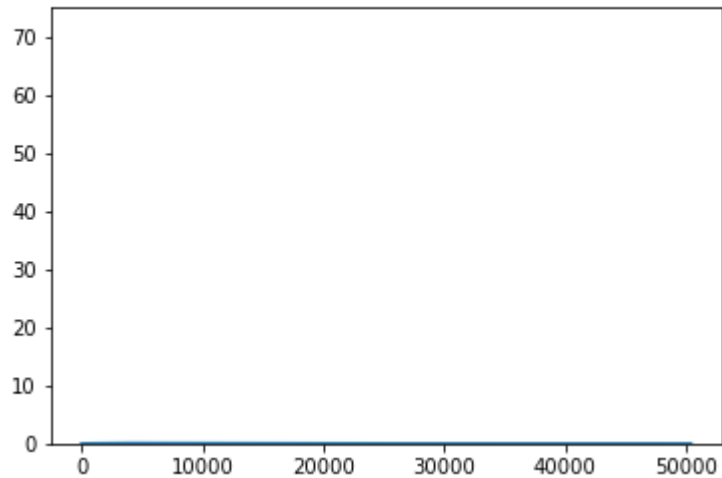
condition 18



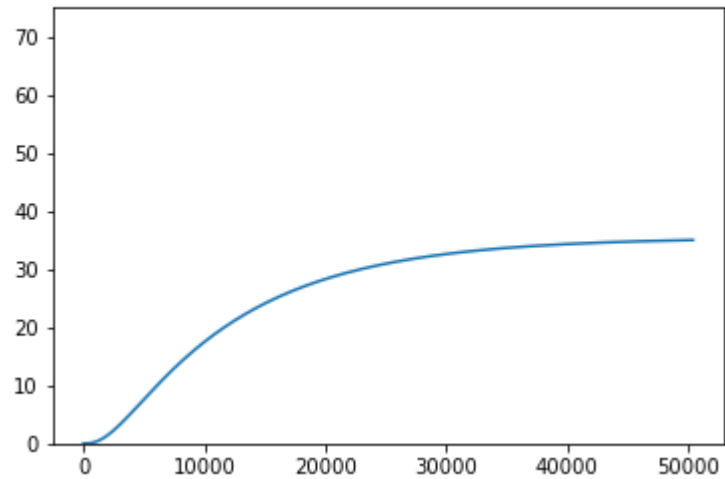
condition 19



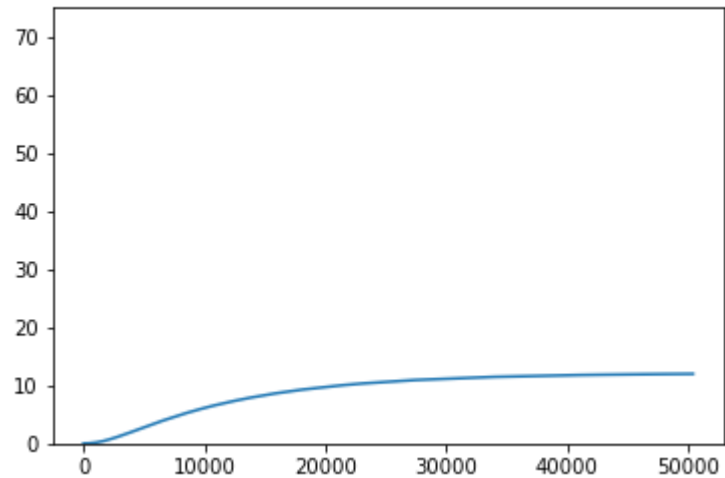
condition 20



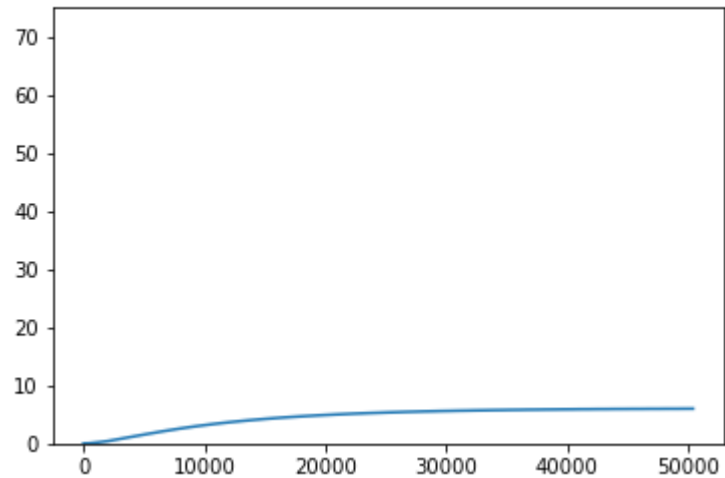
condition 21



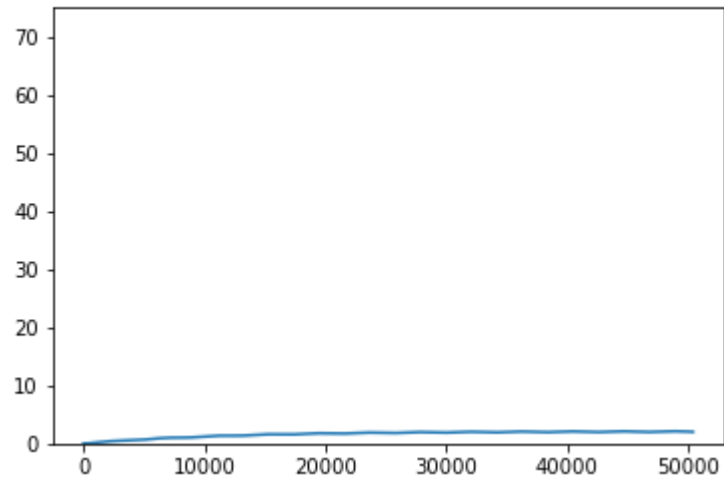
condition 22



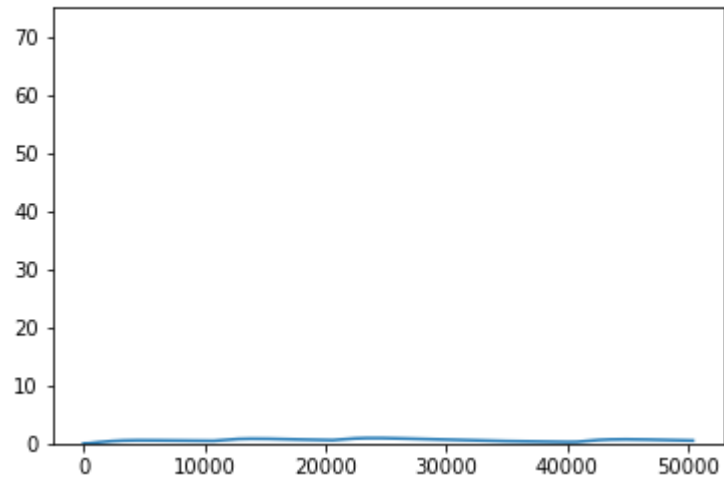
condition 23



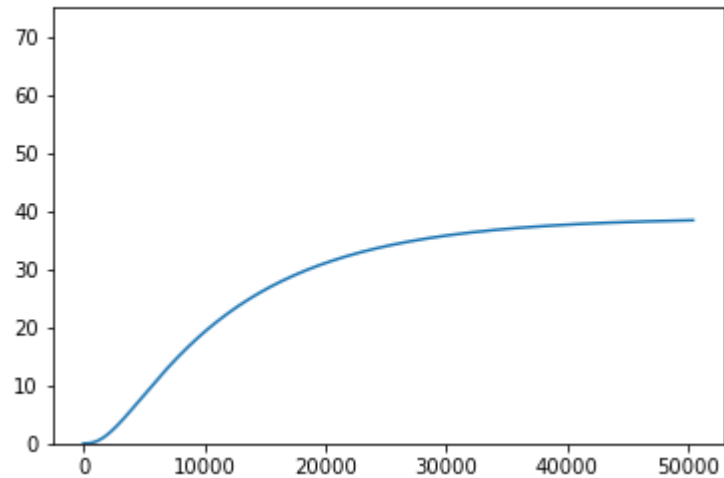
condition 24



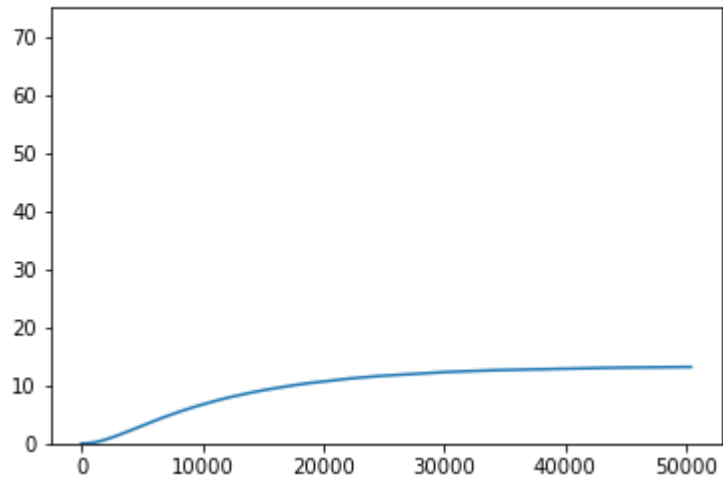
condition 25



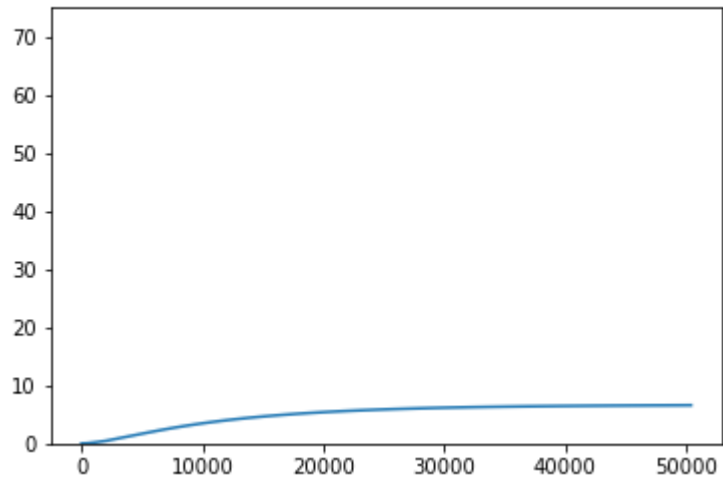
condition 26



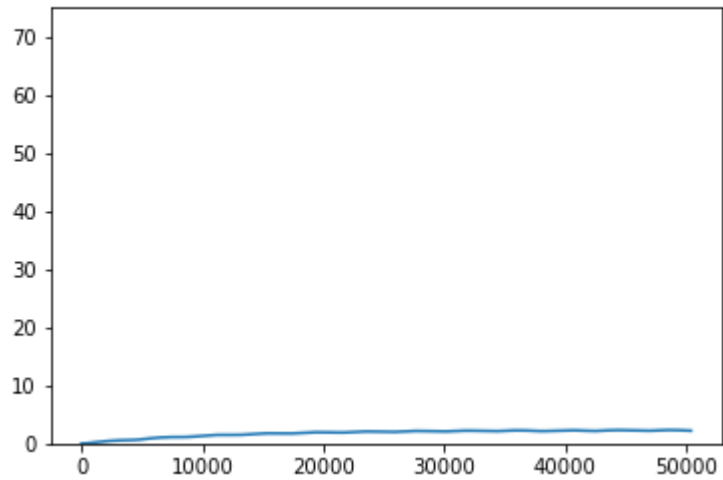
condition 27

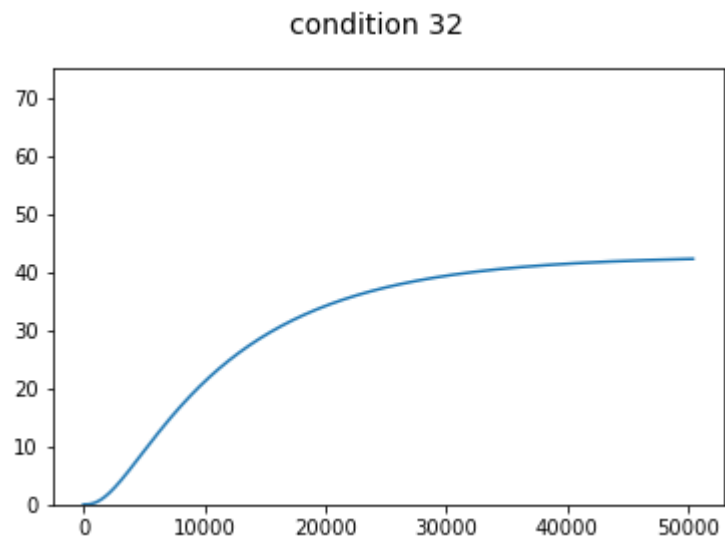
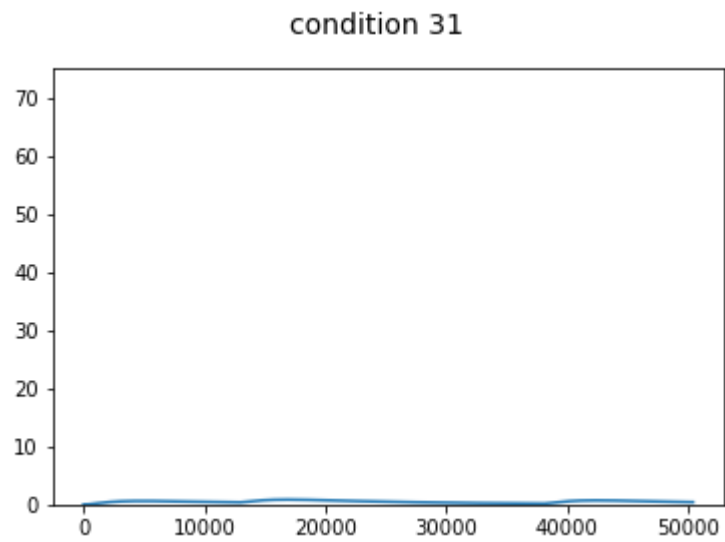
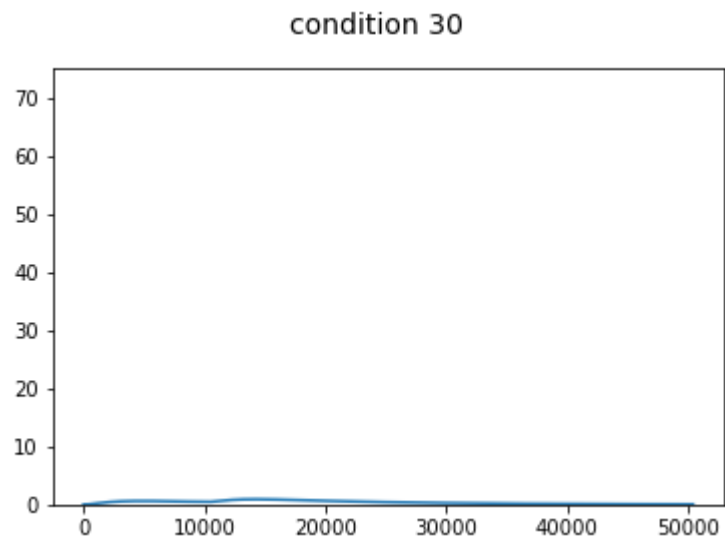


condition 28

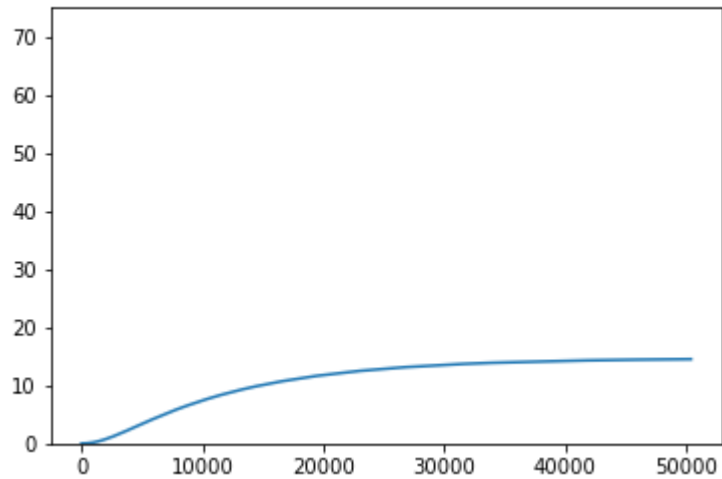


condition 29

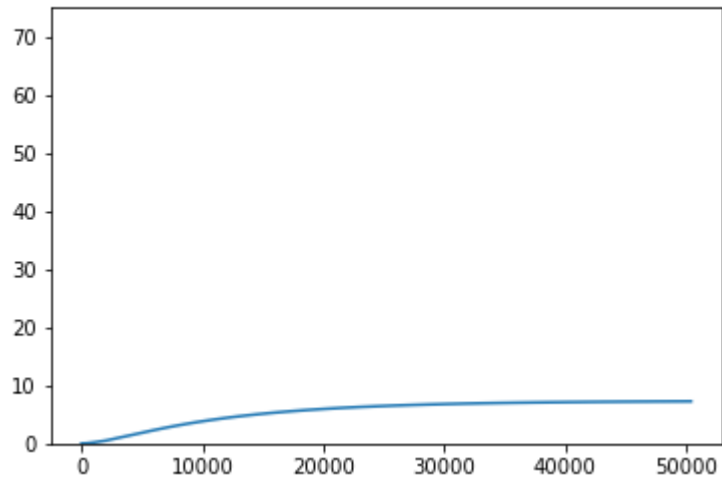




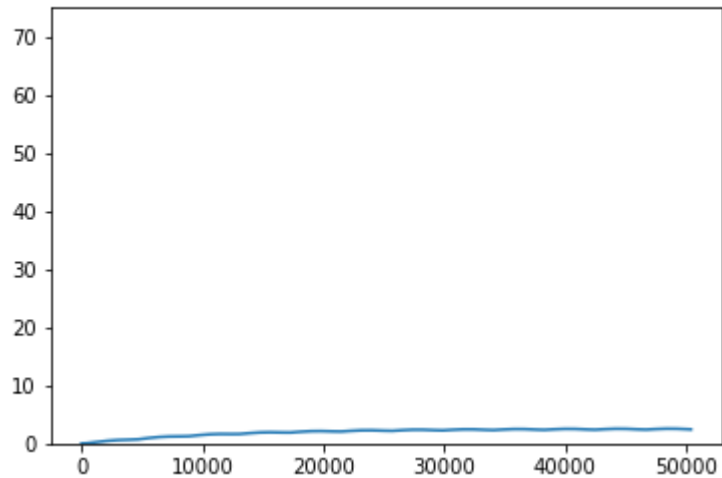
condition 33



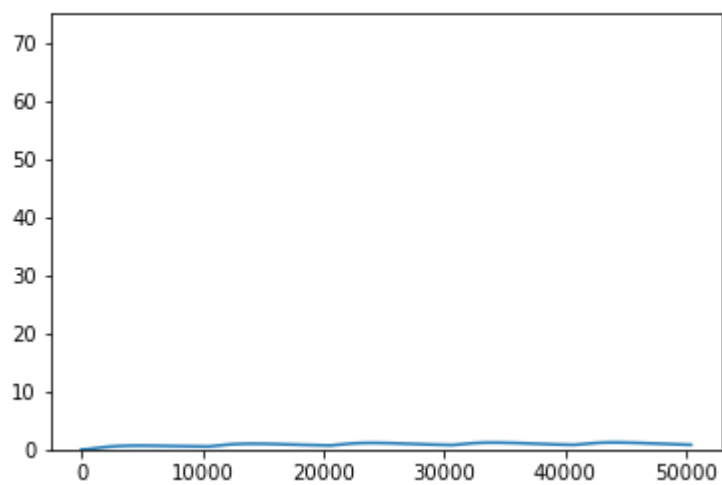
condition 34



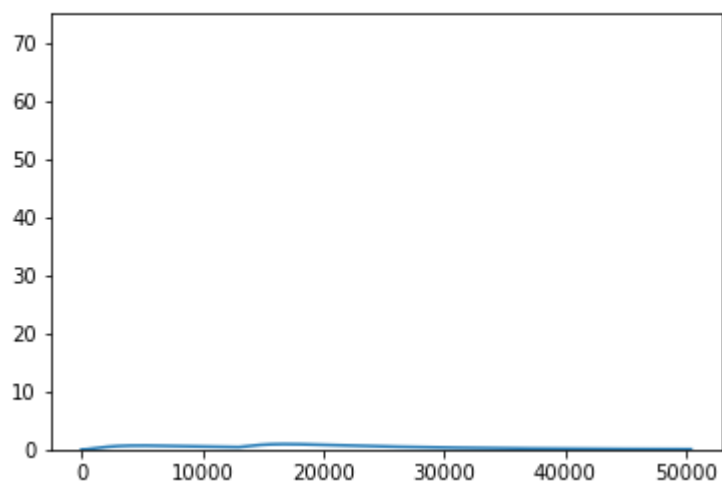
condition 35



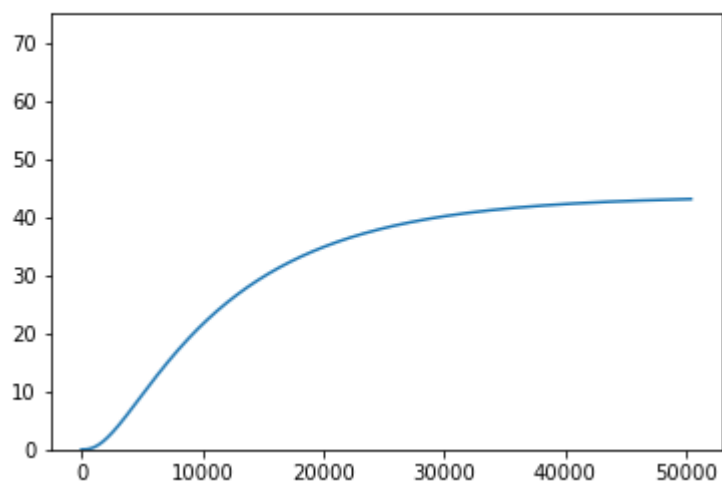
condition 36



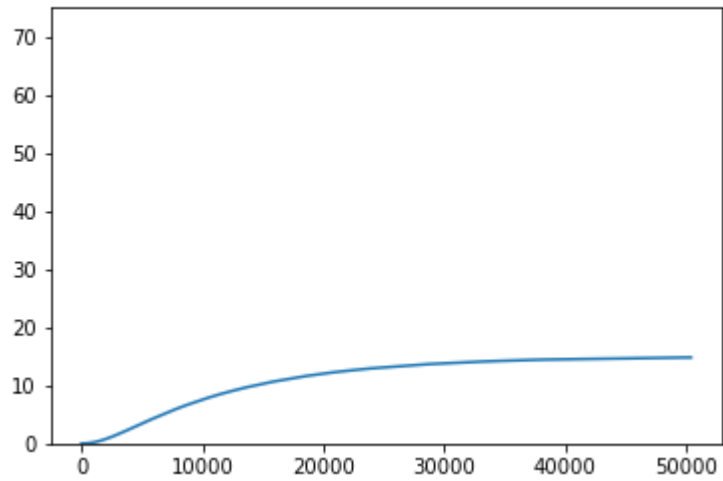
condition 37



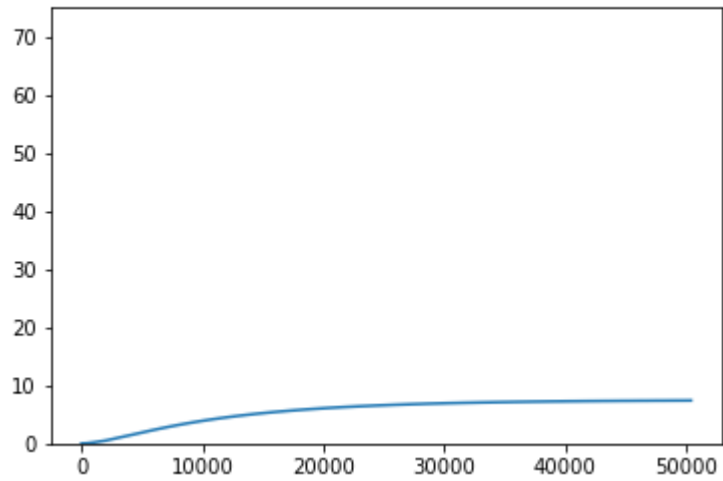
condition 38



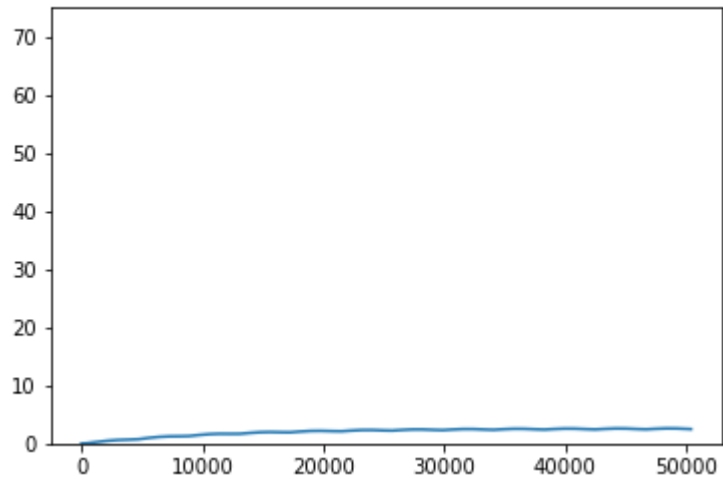
condition 39



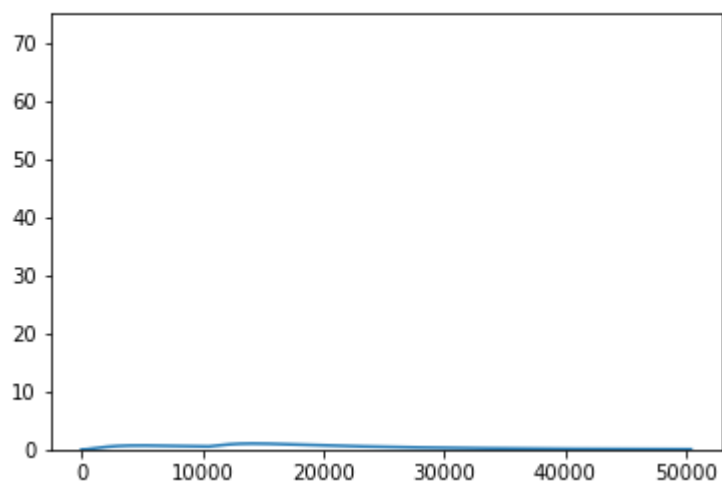
condition 40



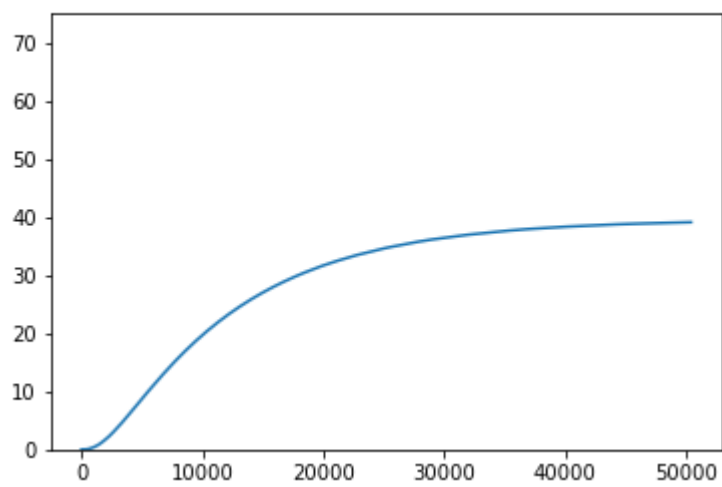
condition 41



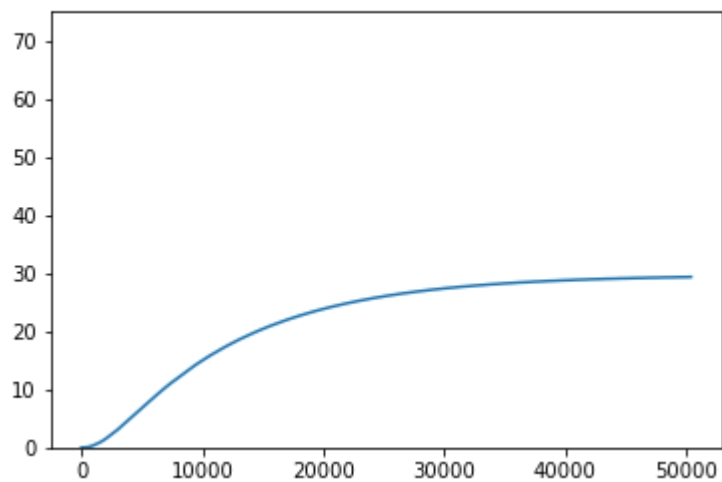
condition 42



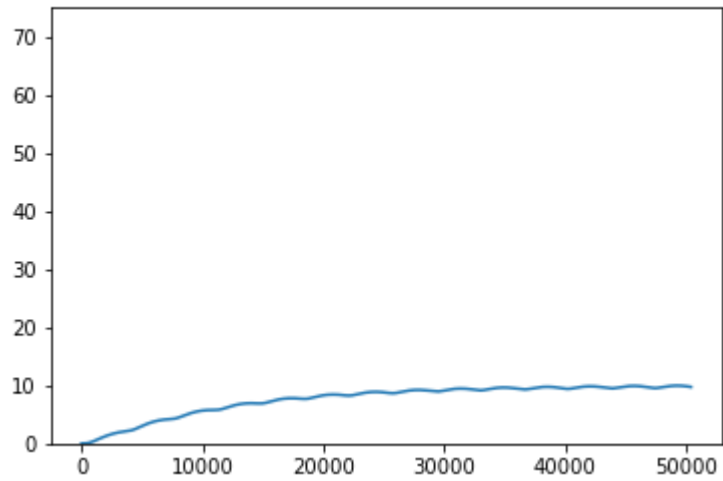
condition 43



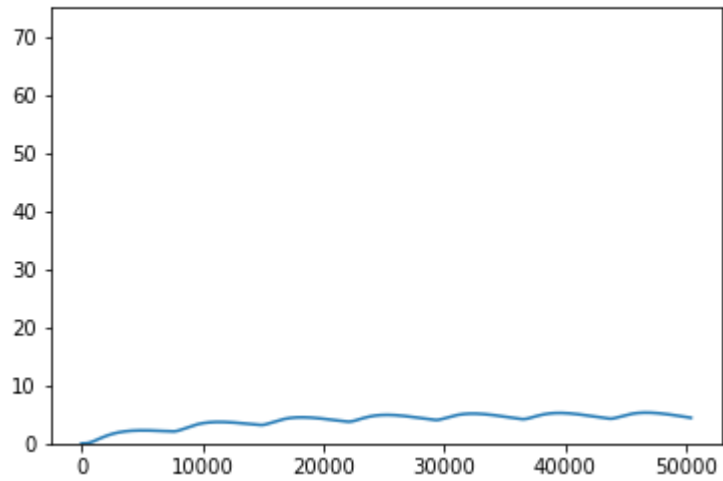
condition 44



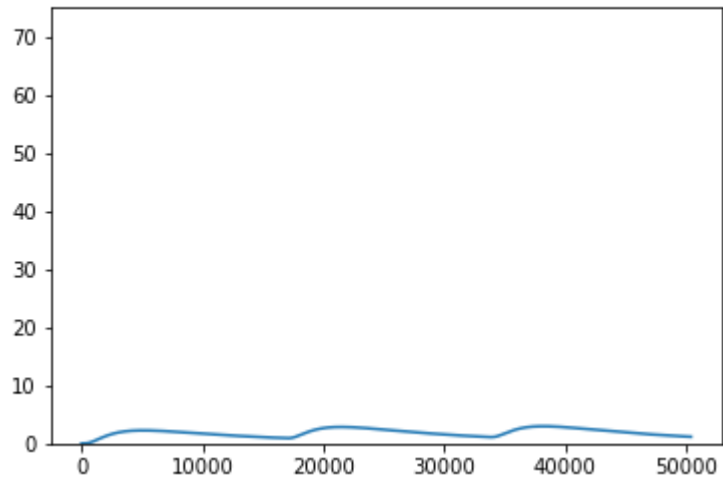
condition 45



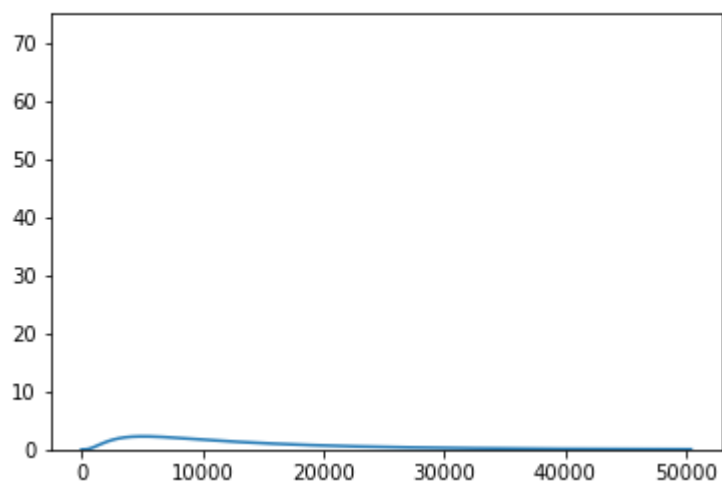
condition 46



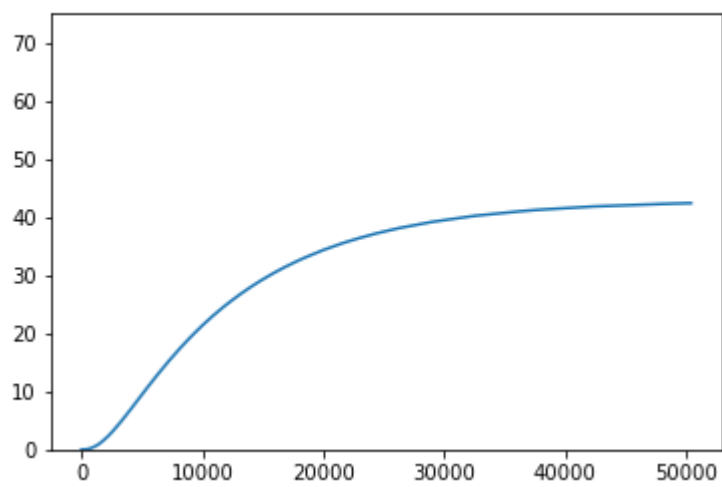
condition 47



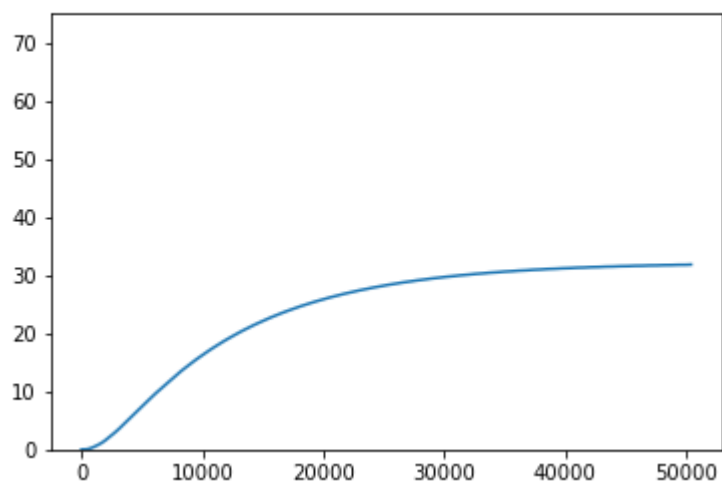
condition 48



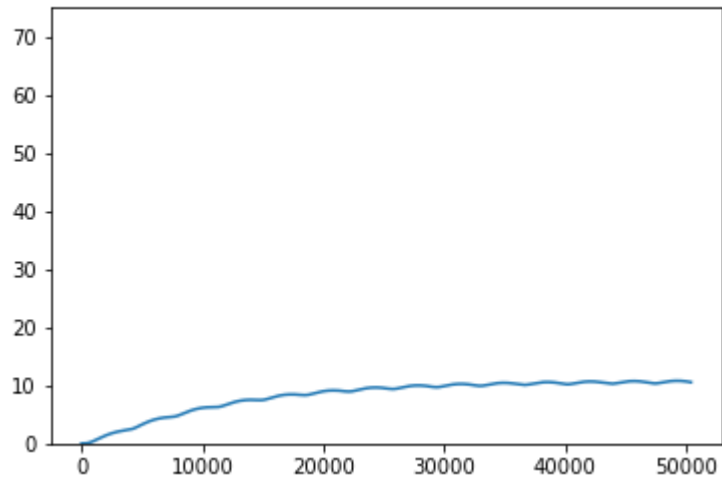
condition 49



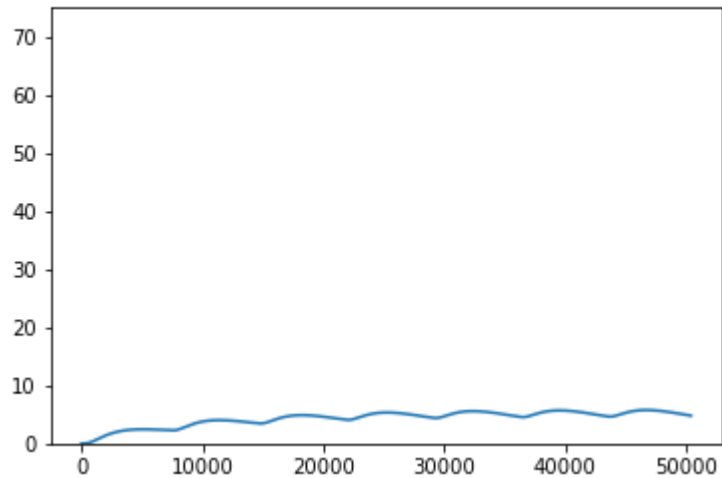
condition 50



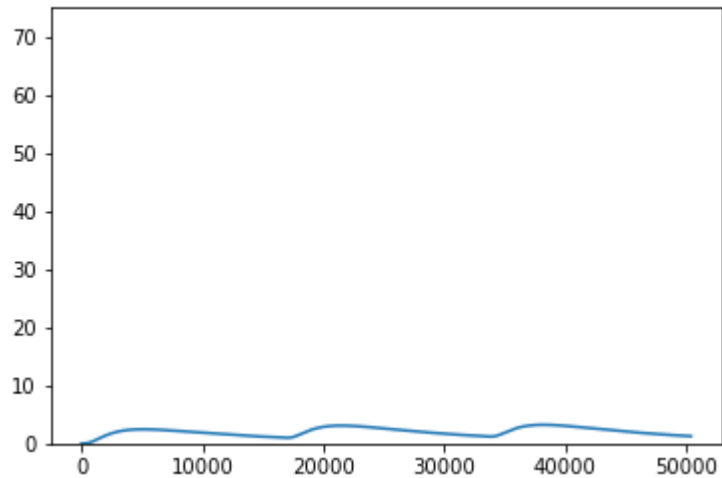
condition 51



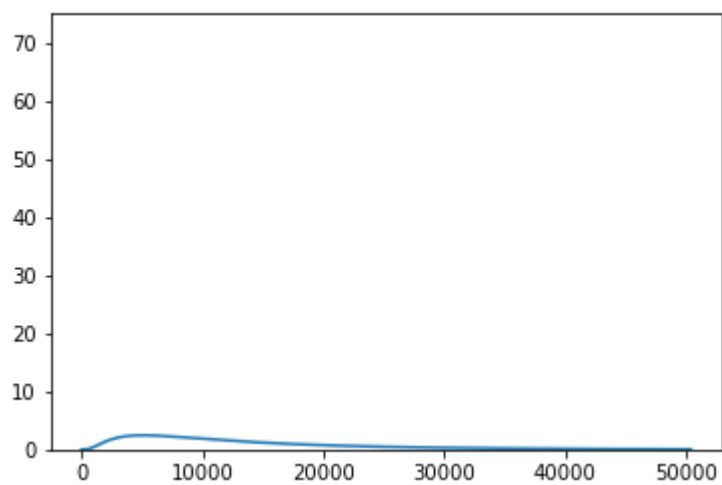
condition 52



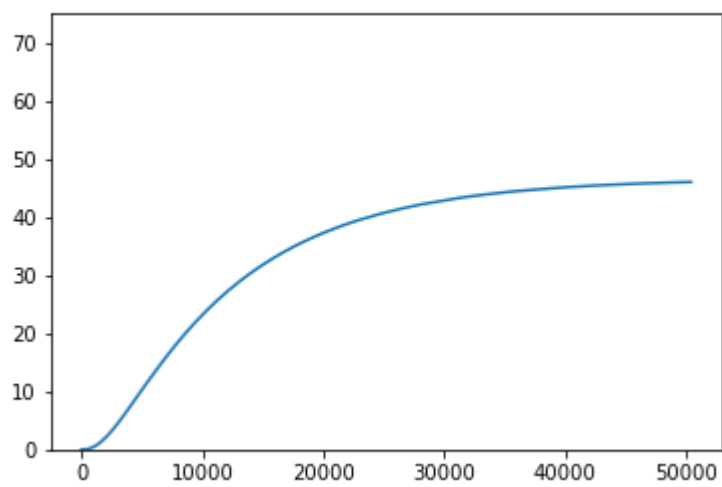
condition 53



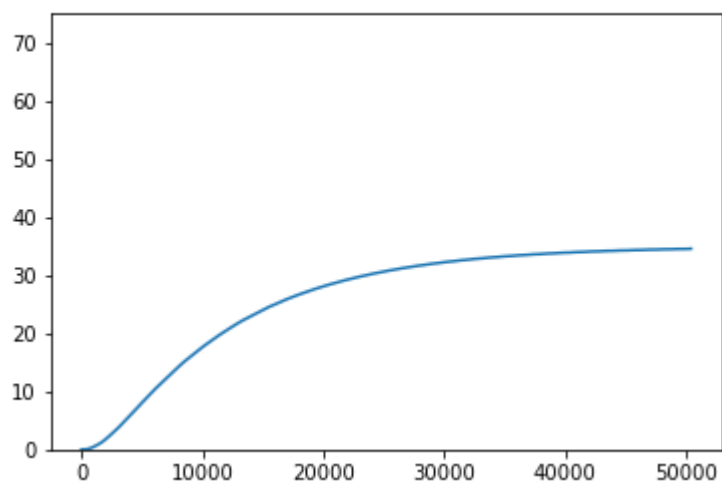
condition 54



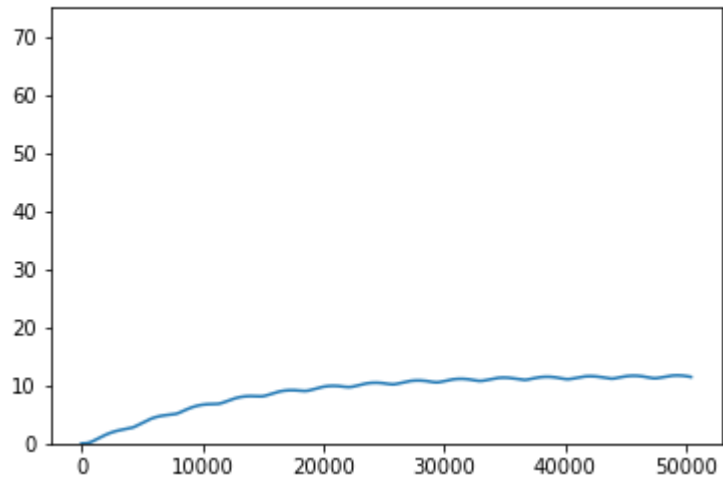
condition 55



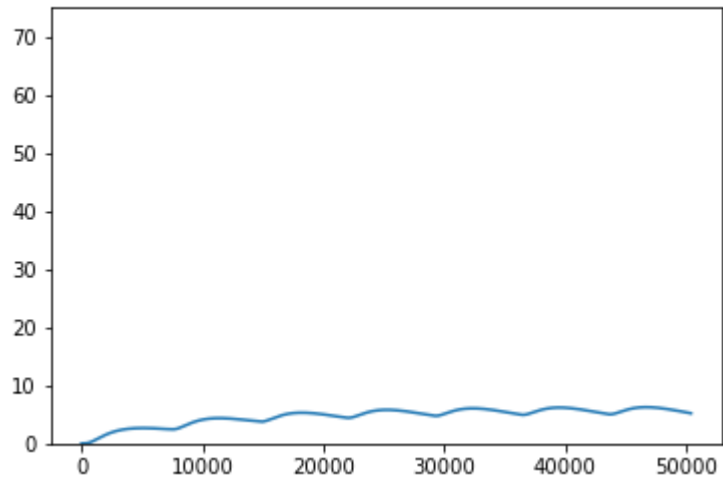
condition 56



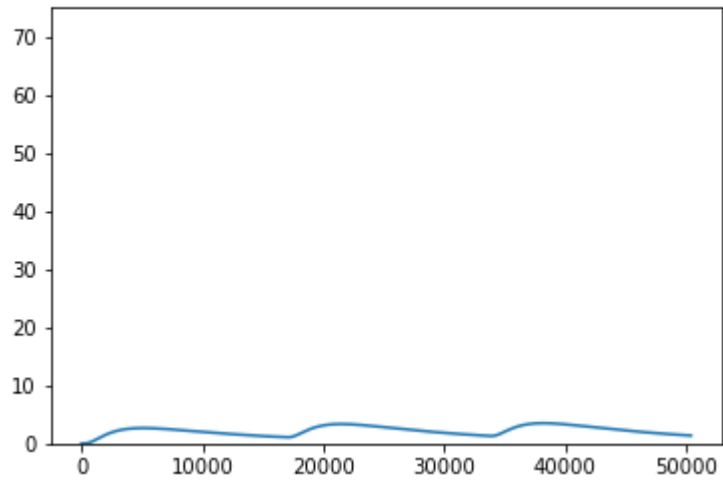
condition 57



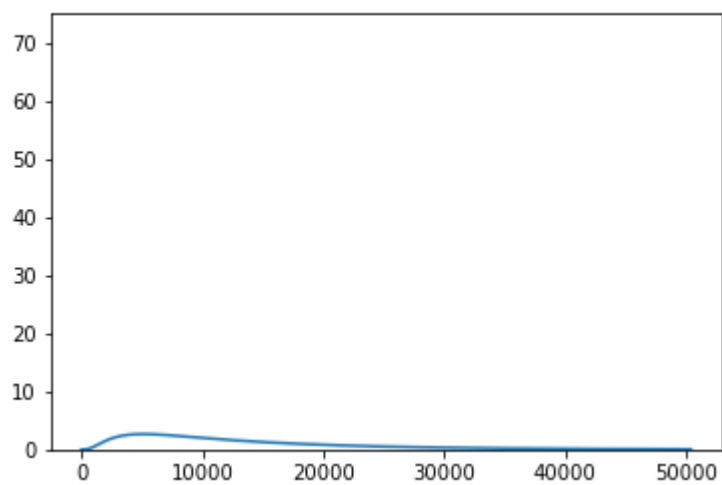
condition 58



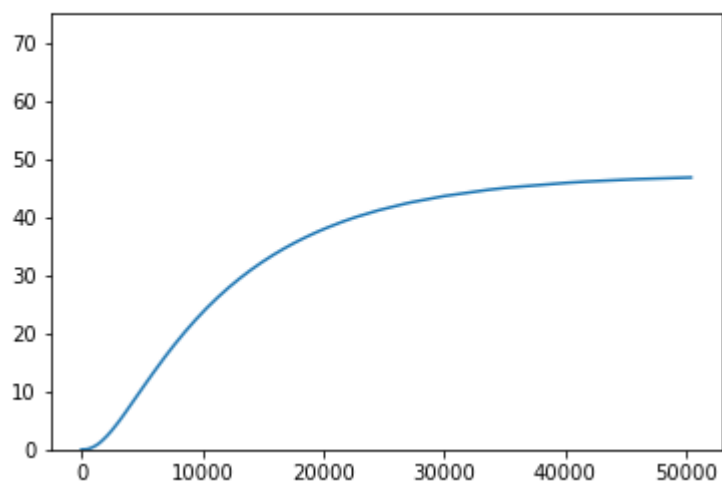
condition 59



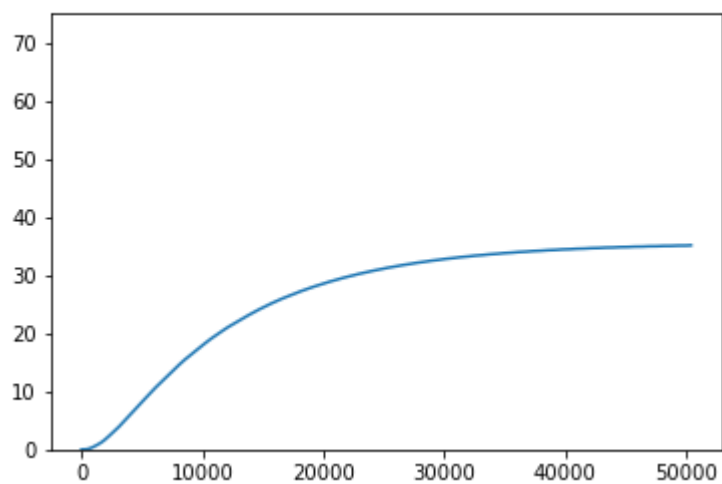
condition 60



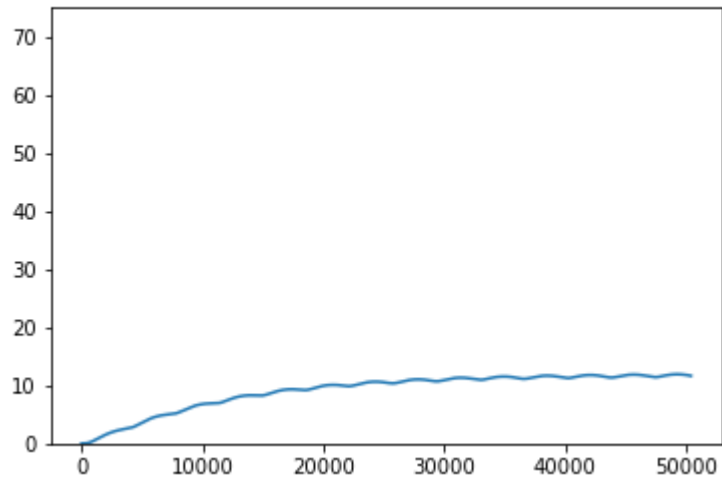
condition 61



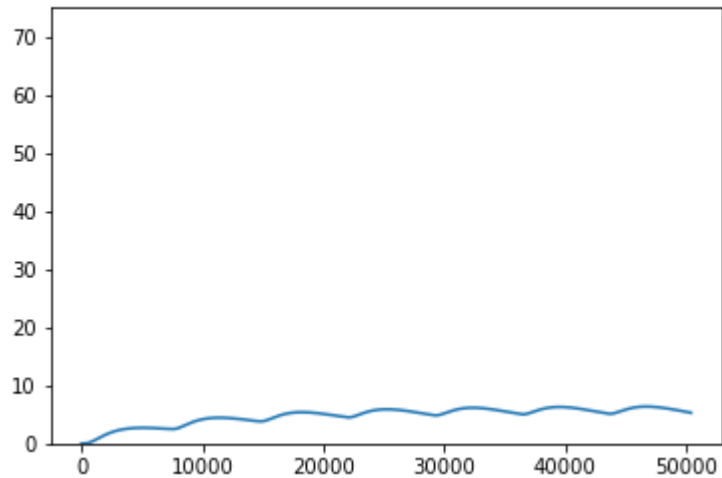
condition 62



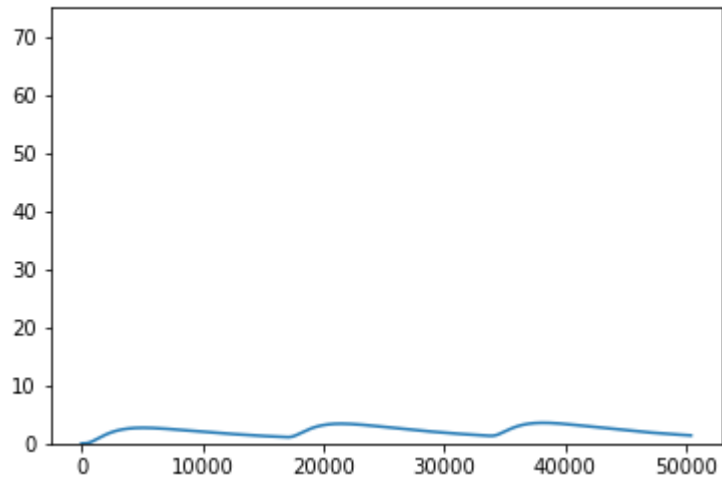
condition 63



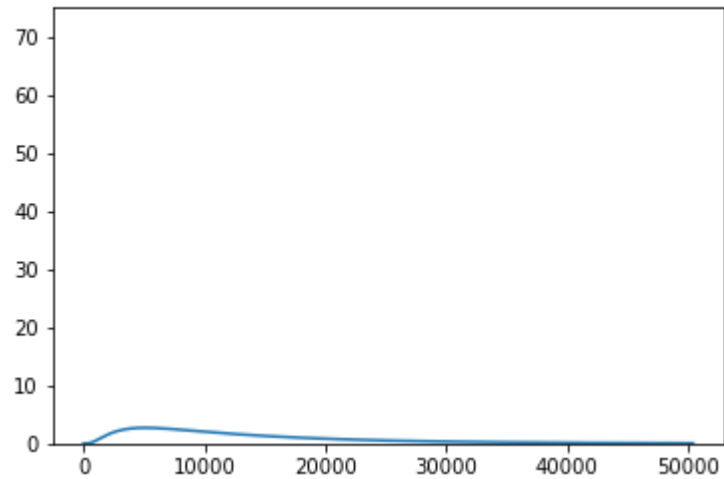
condition 64



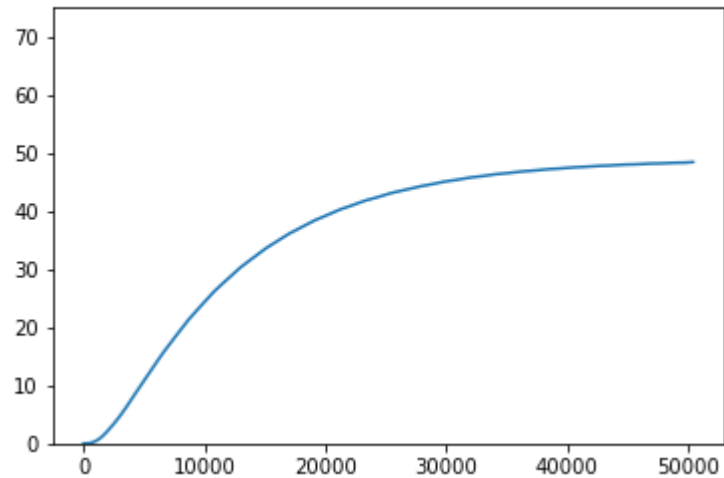
condition 65



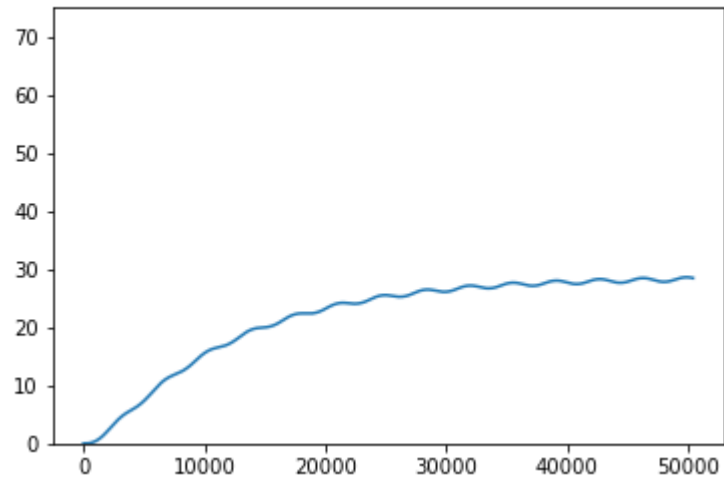
condition 66



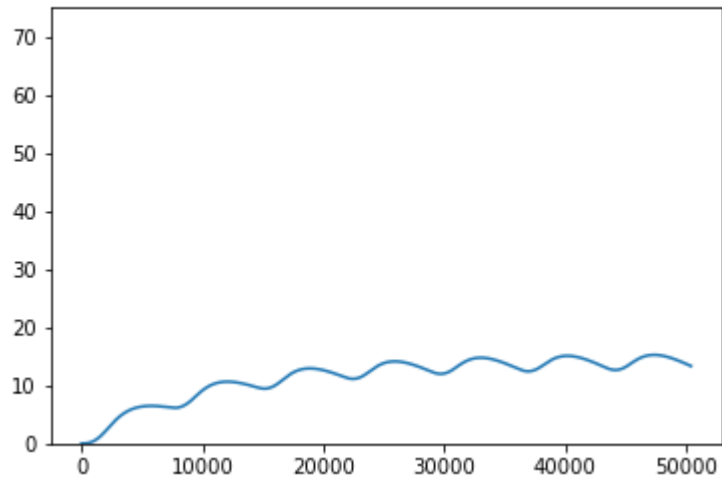
condition 67



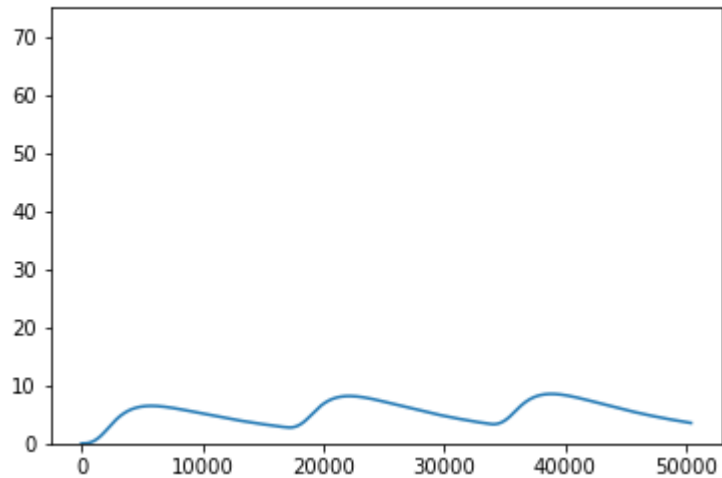
condition 68



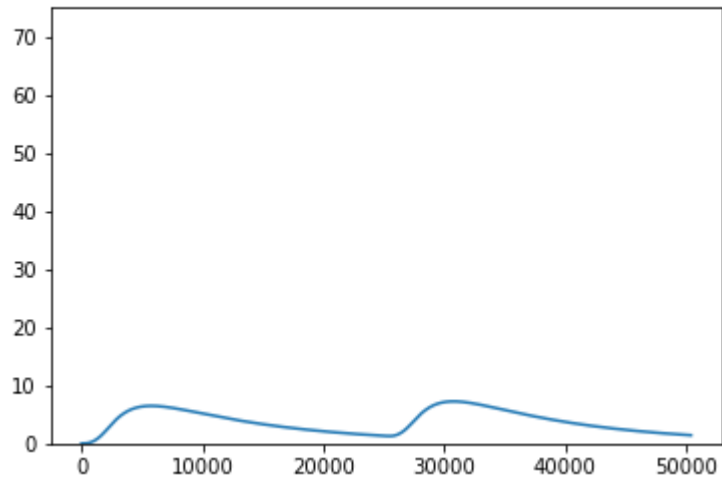
condition 69



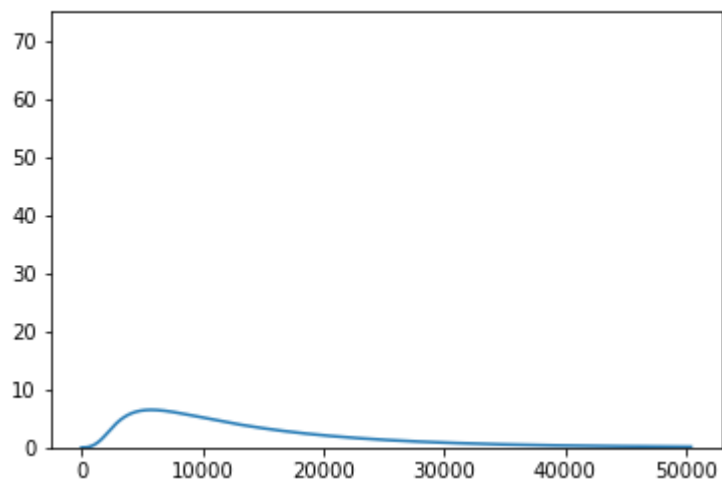
condition 70



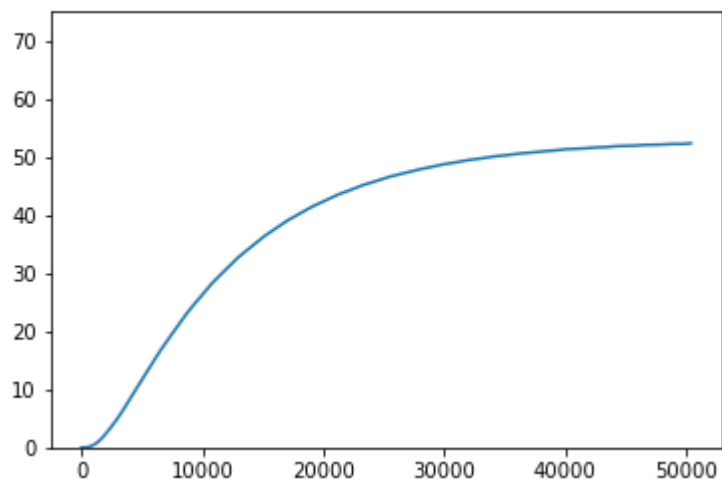
condition 71



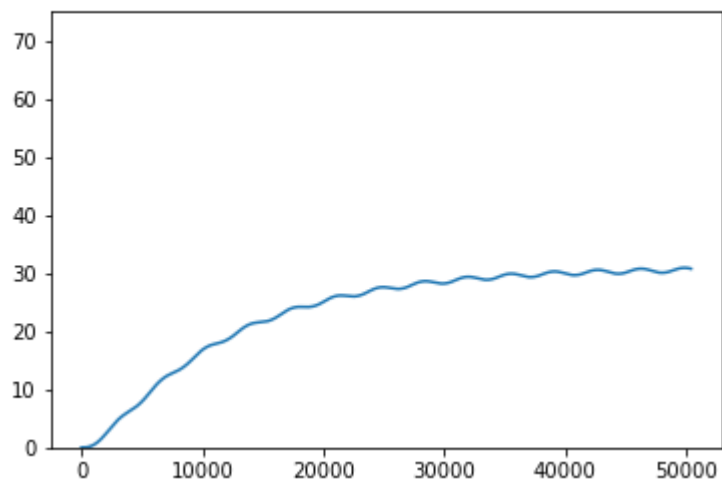
condition 72



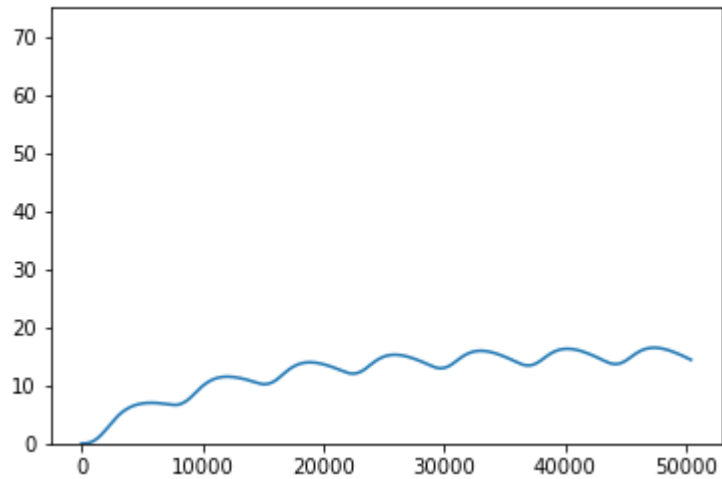
condition 73



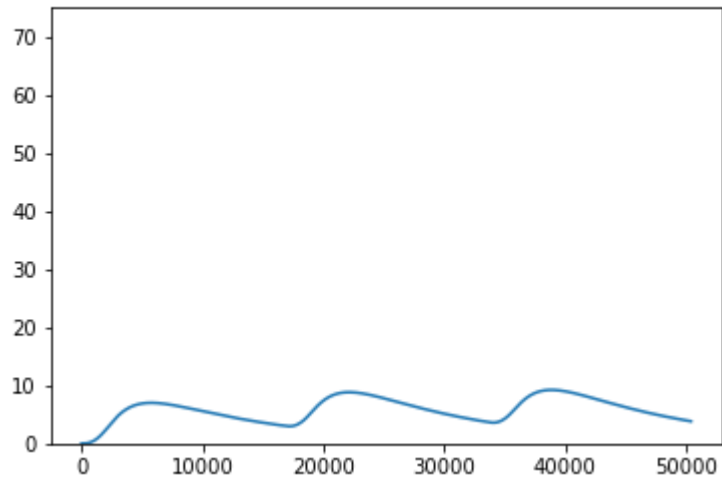
condition 74



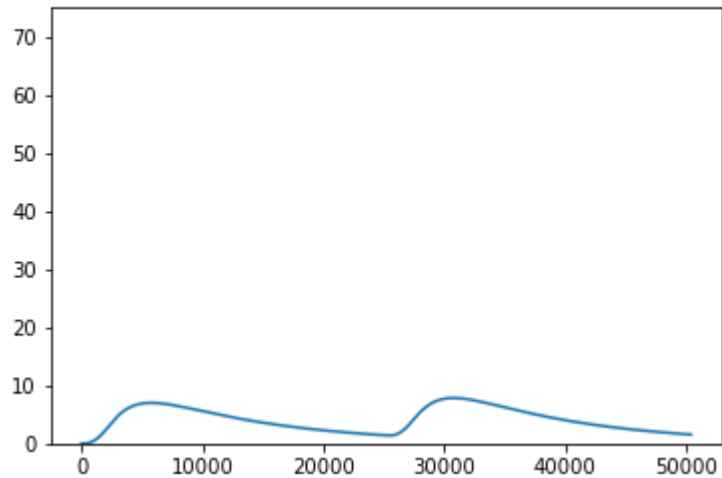
condition 75



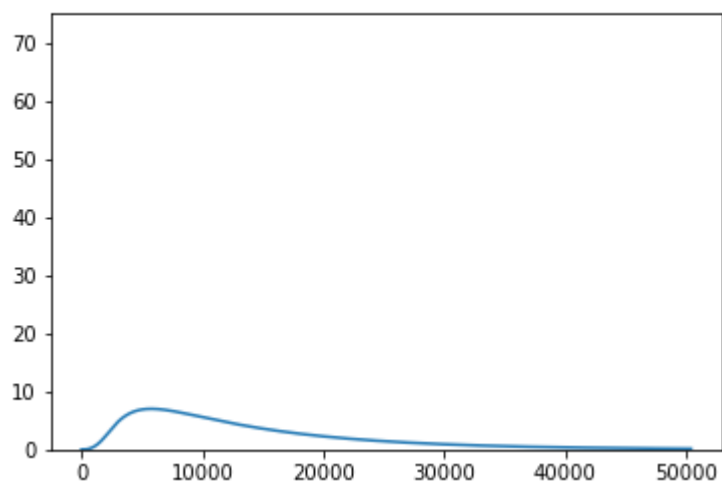
condition 76



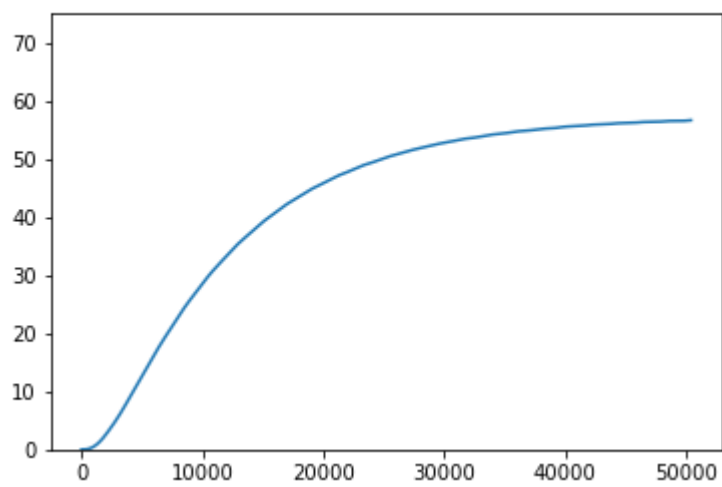
condition 77



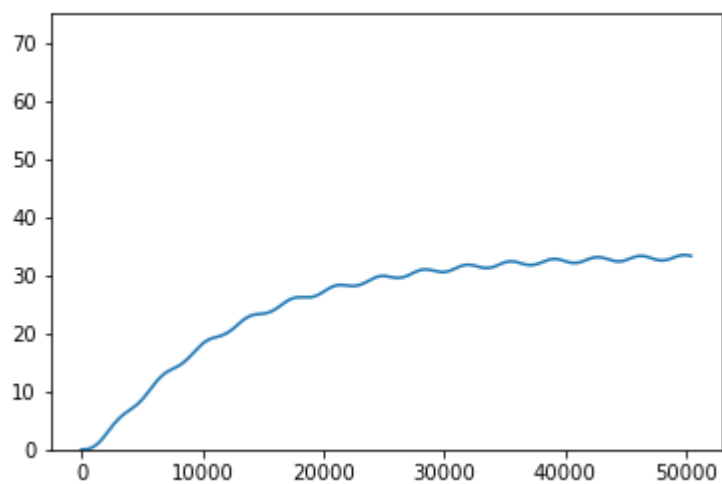
condition 78



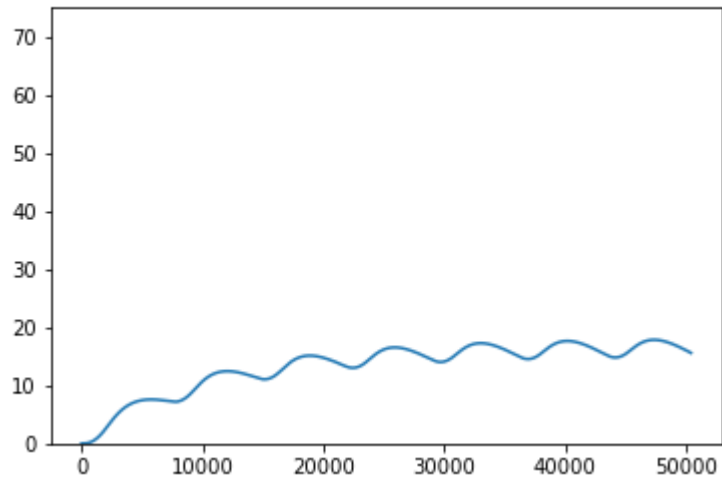
condition 79



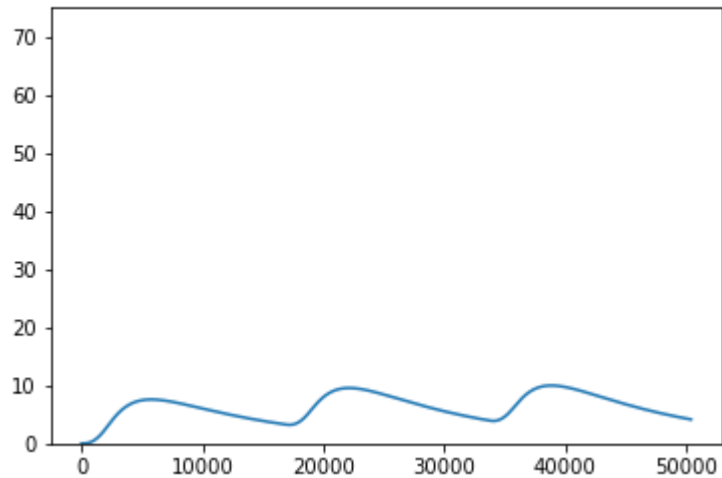
condition 80



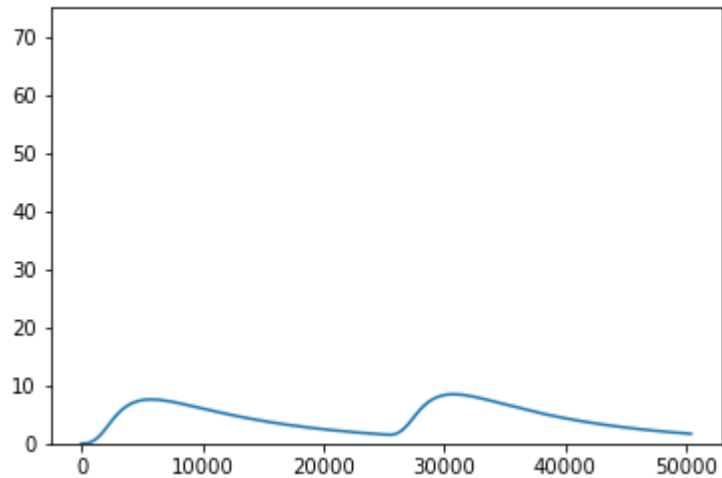
condition 81



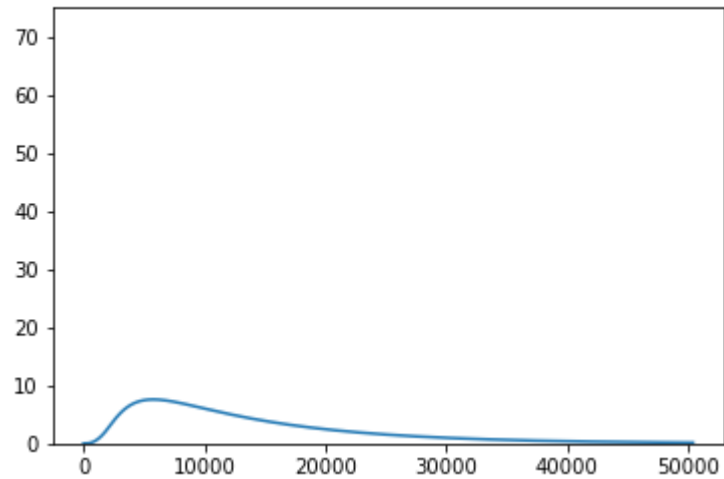
condition 82



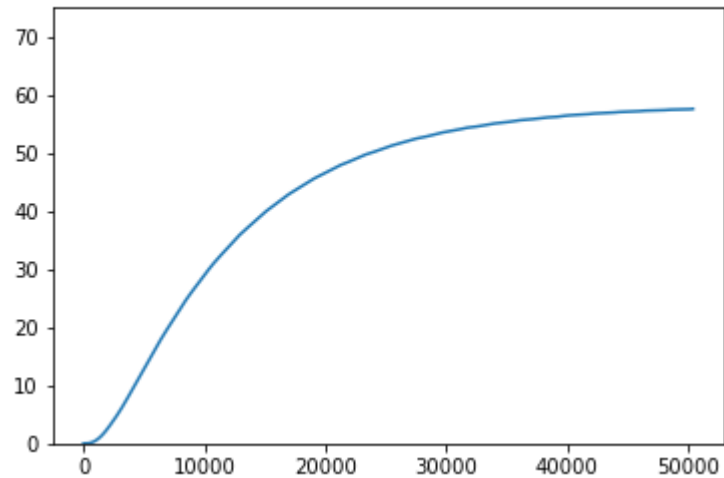
condition 83



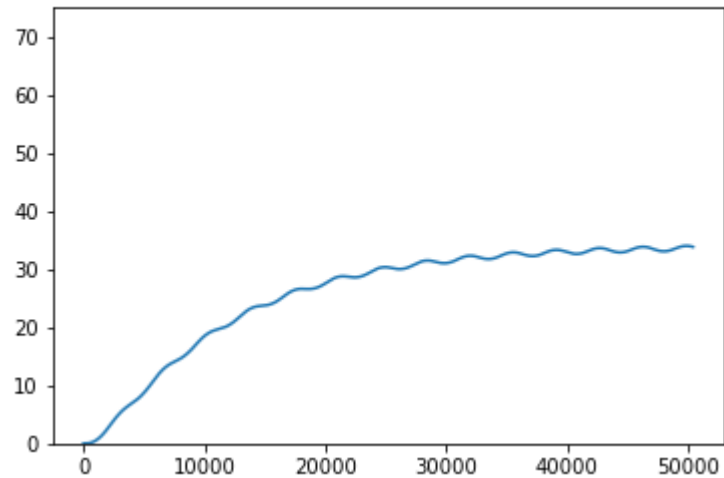
condition 84



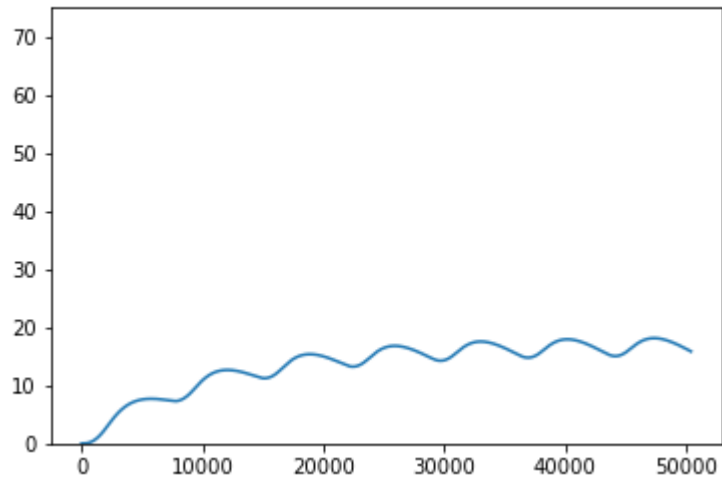
condition 85



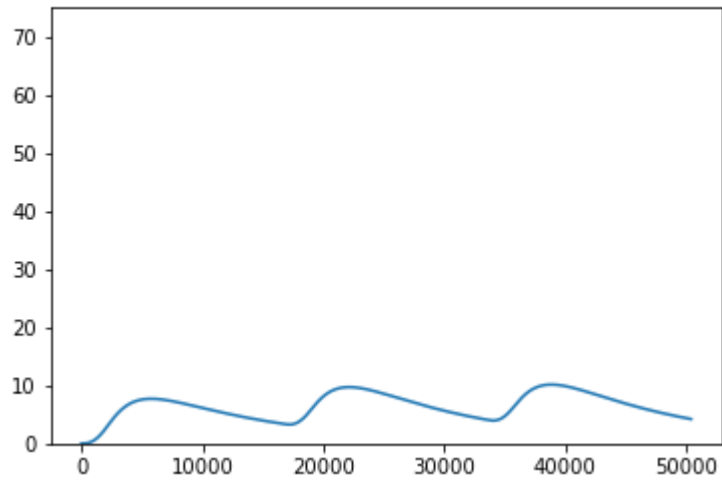
condition 86



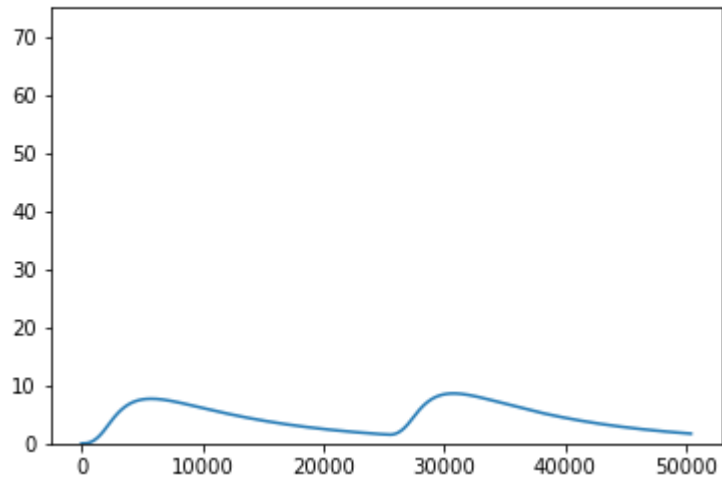
condition 87



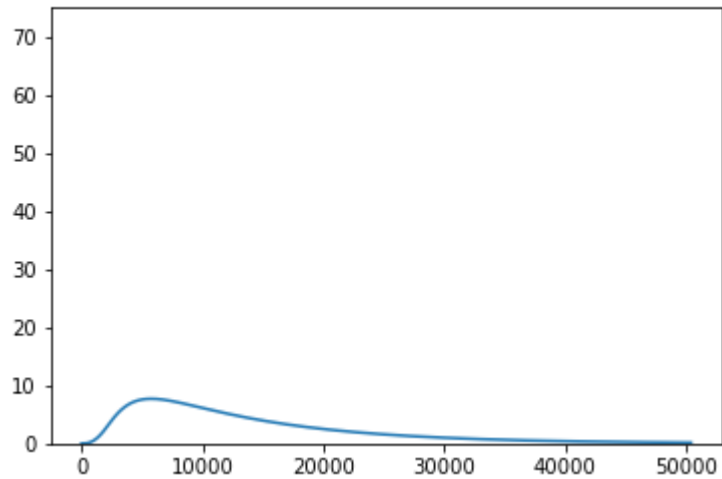
condition 88



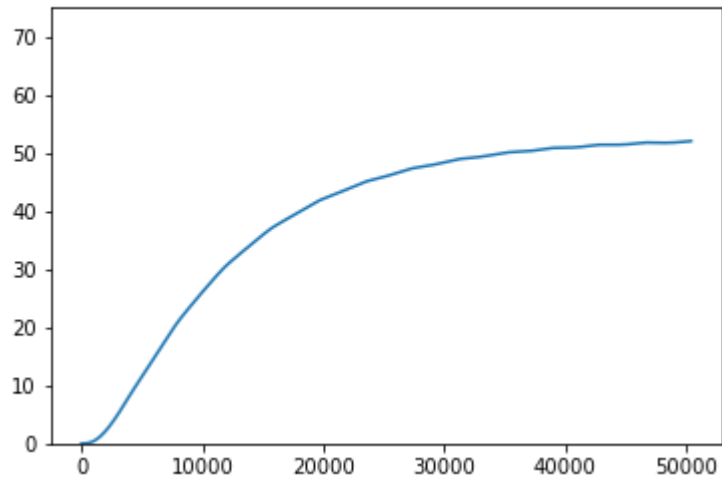
condition 89



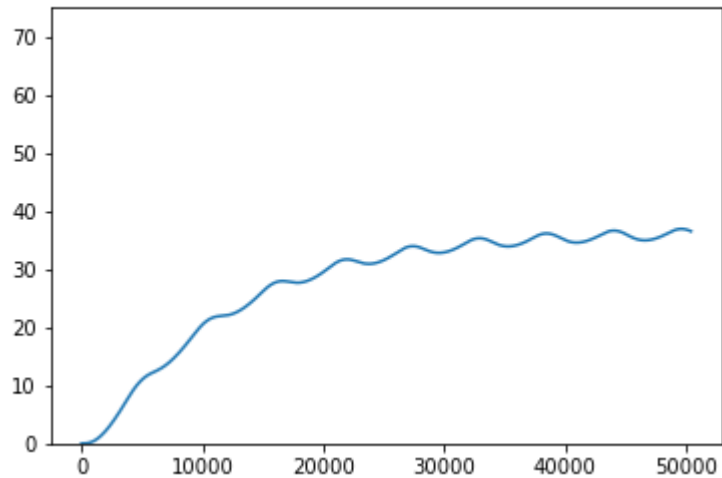
condition 90



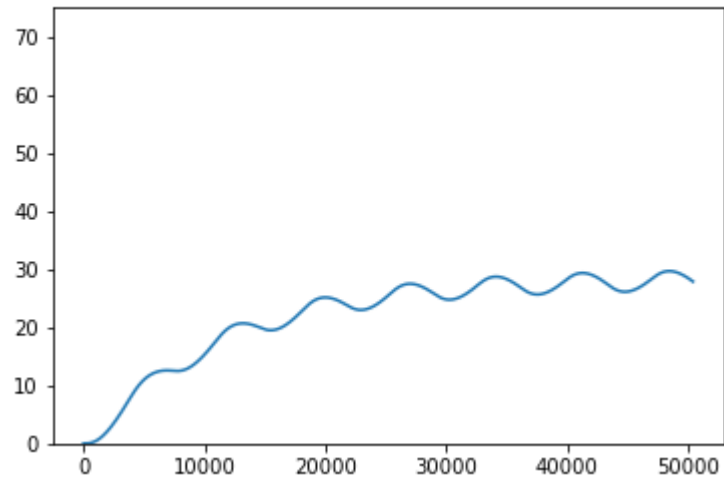
condition 91



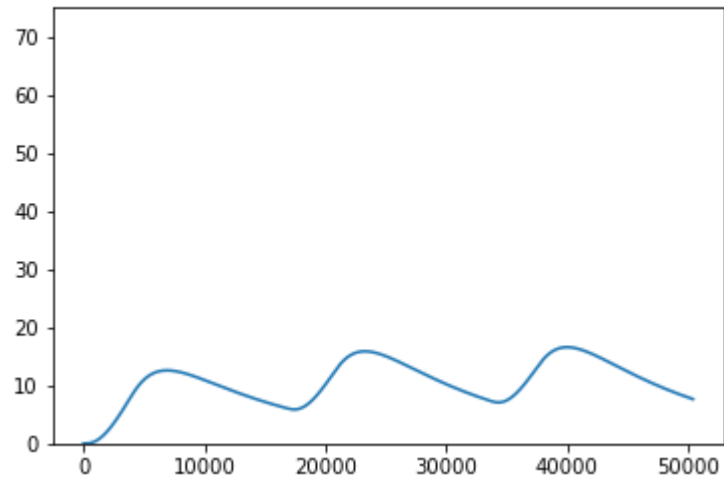
condition 92



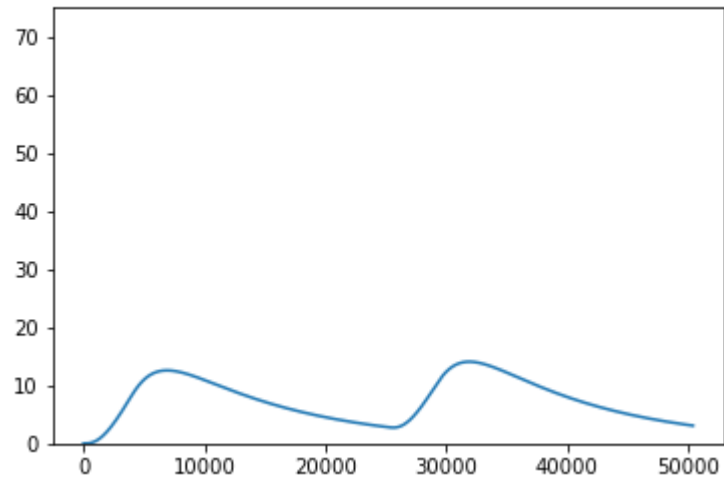
condition 93



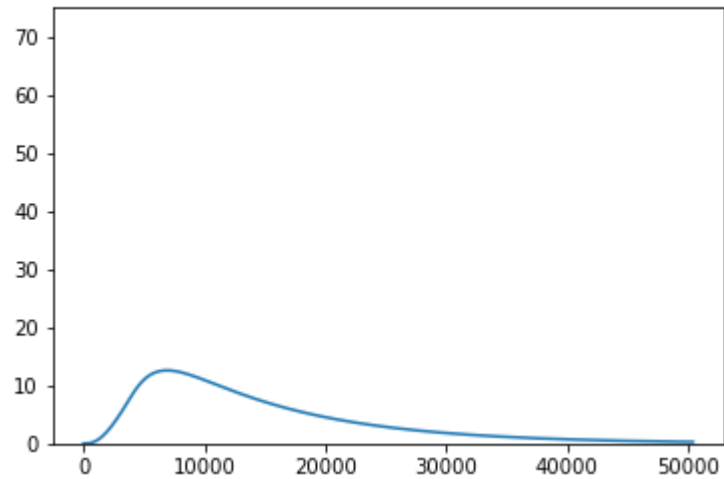
condition 94



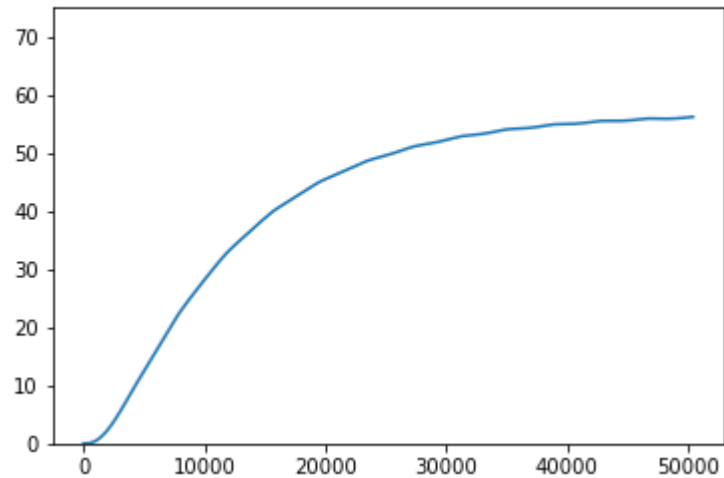
condition 95



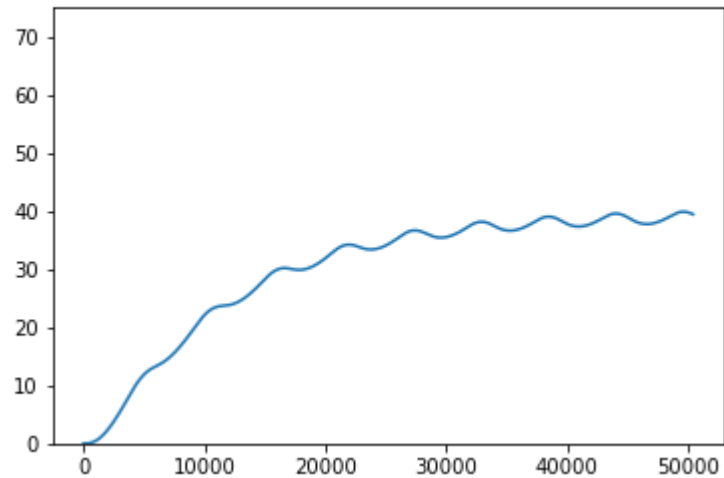
condition 96



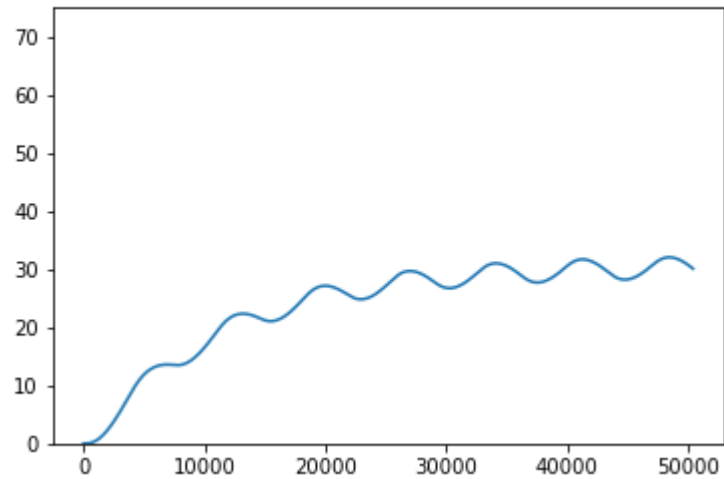
condition 97



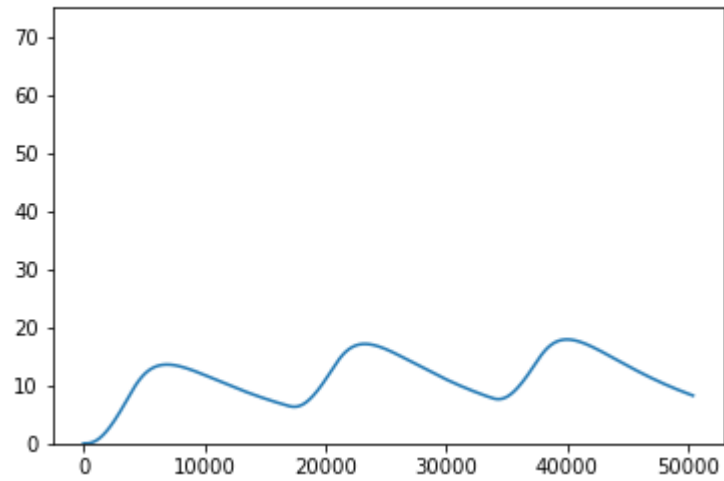
condition 98



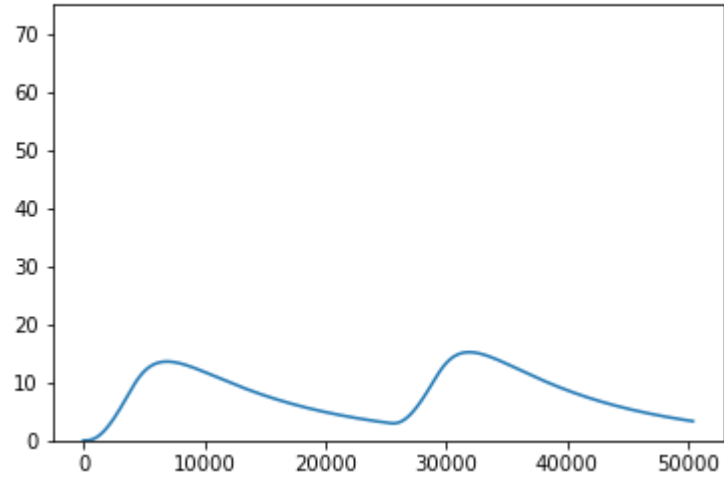
condition 99



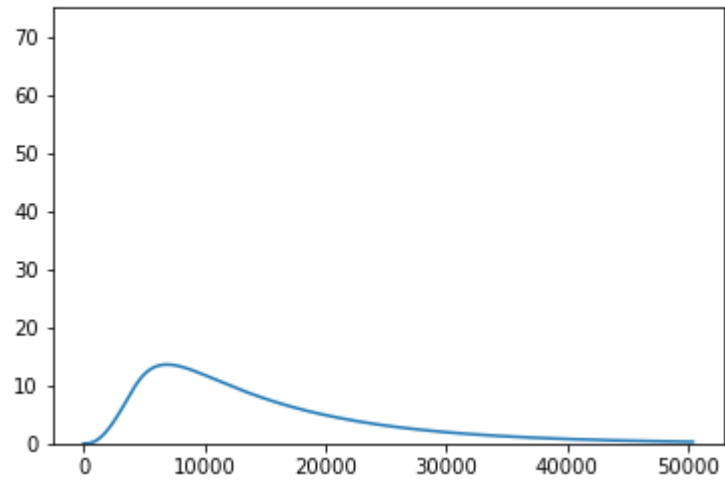
condition 100



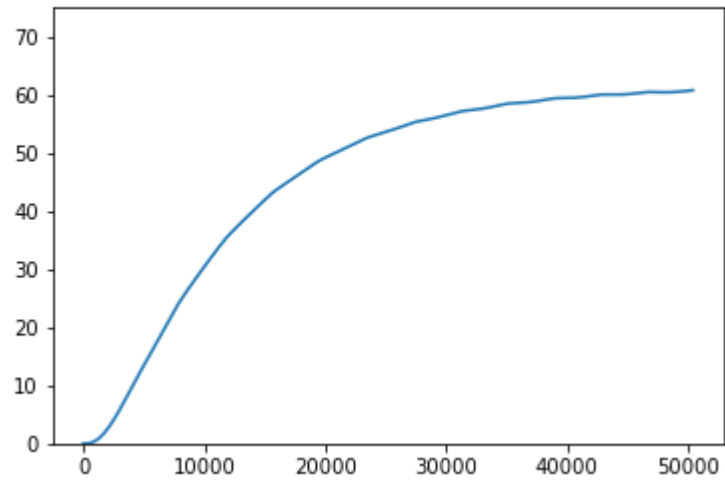
condition 101



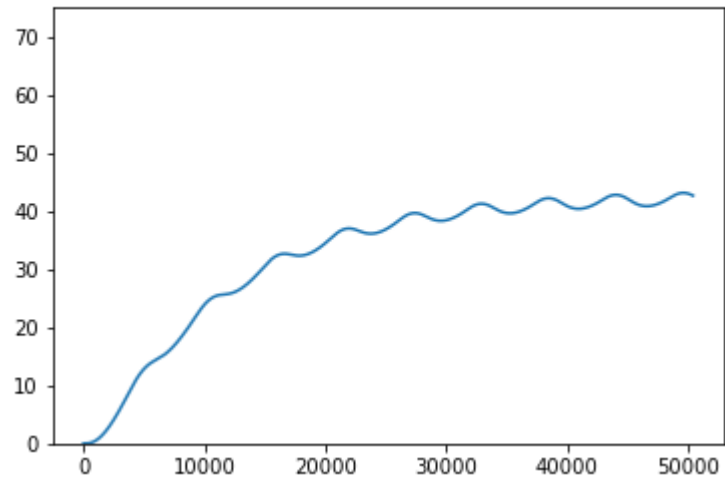
condition 102



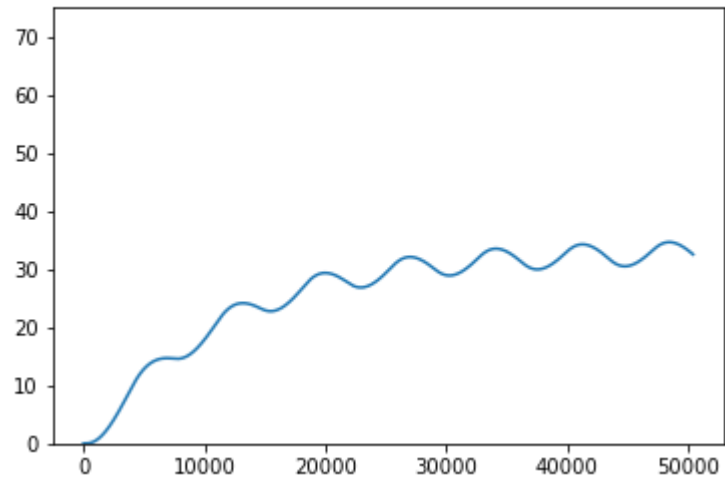
condition 103



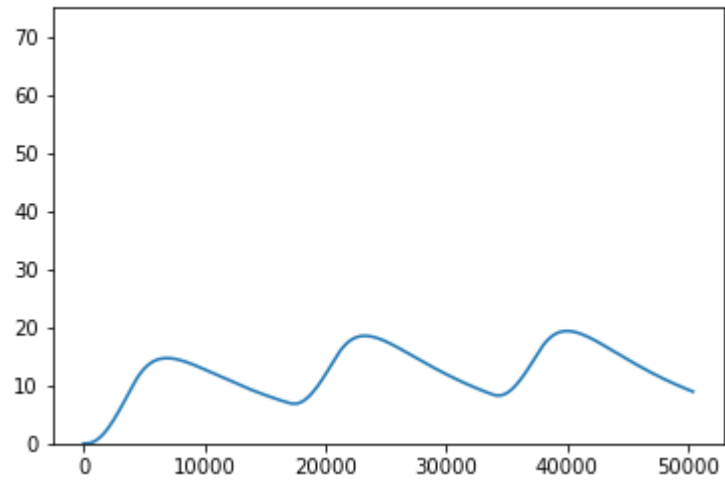
condition 104



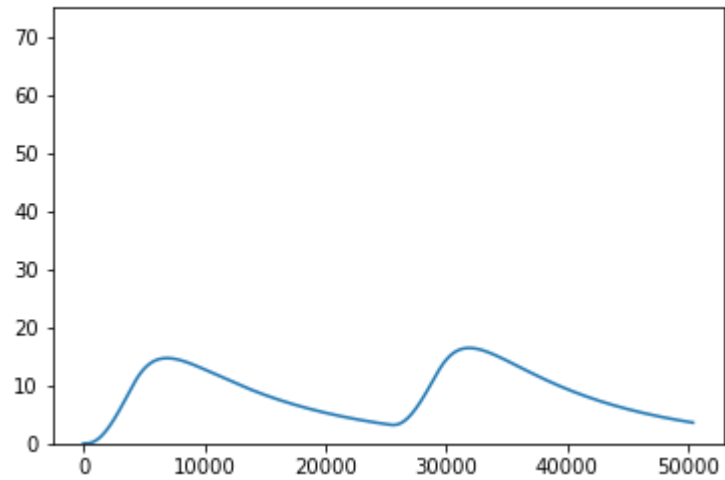
condition 105



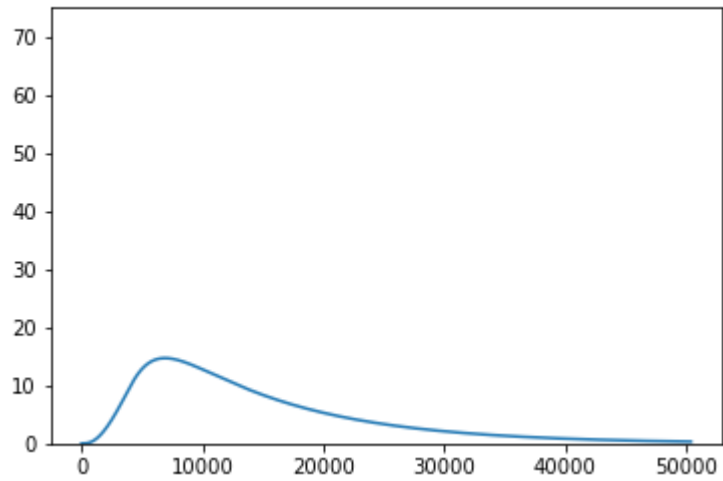
condition 106



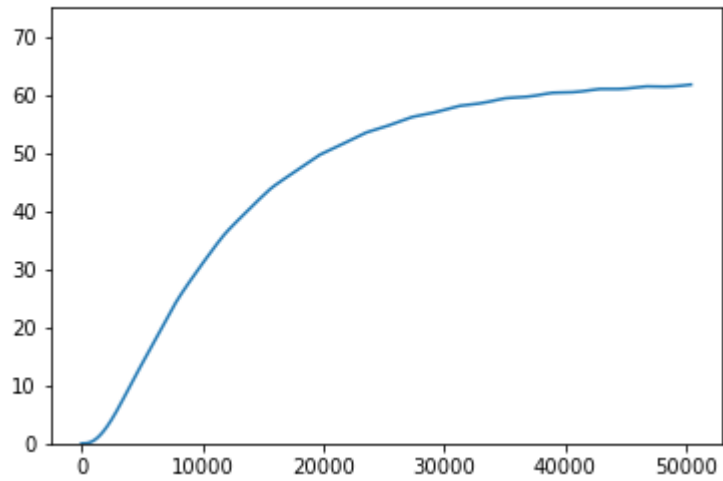
condition 107



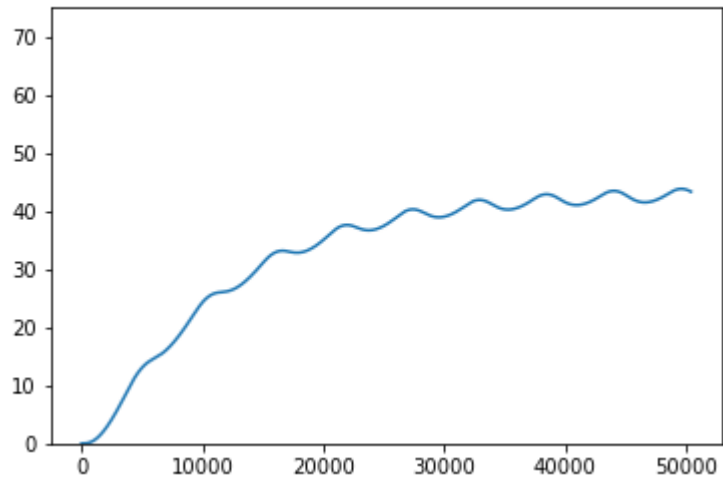
condition 108



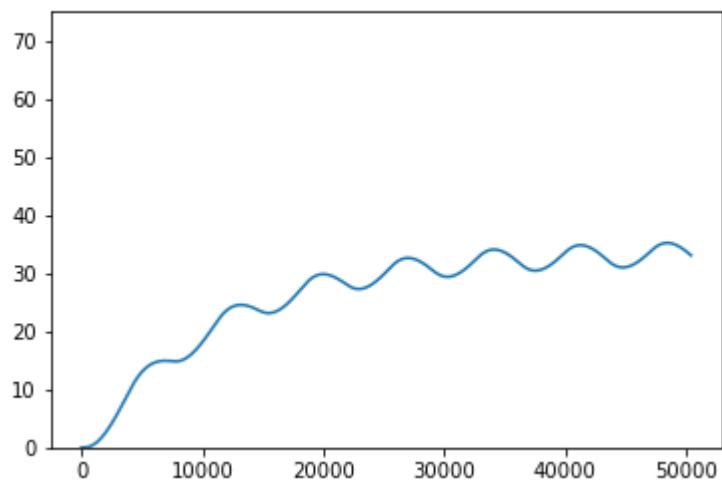
condition 109



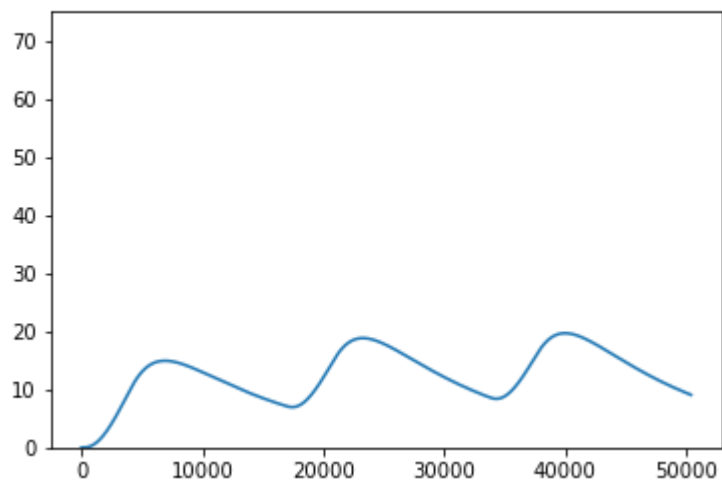
condition 110



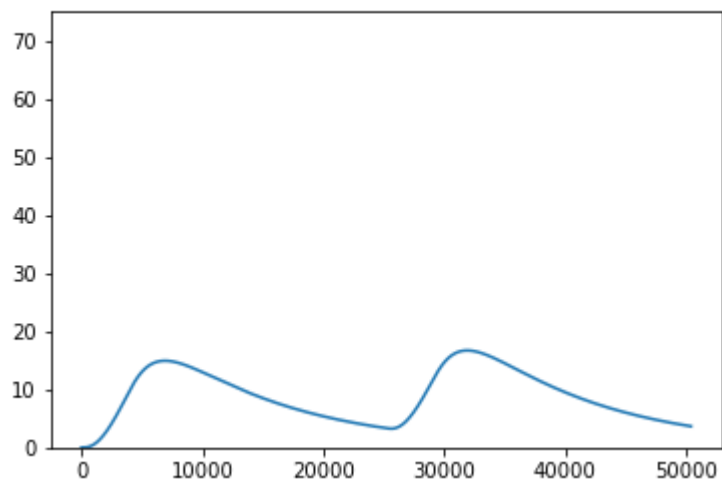
condition 111



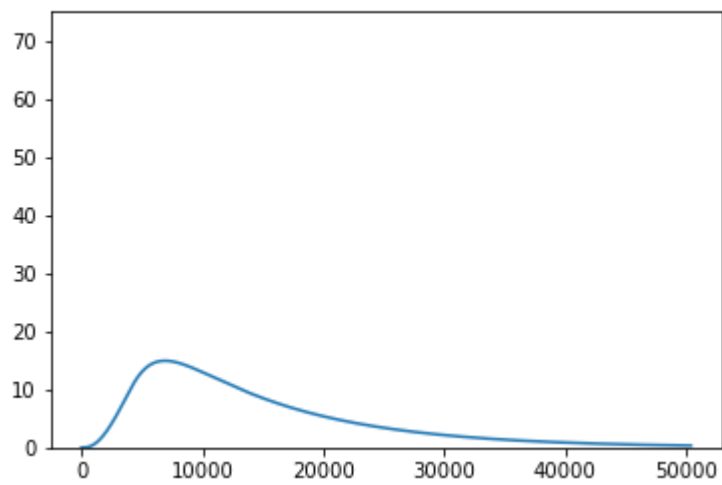
condition 112



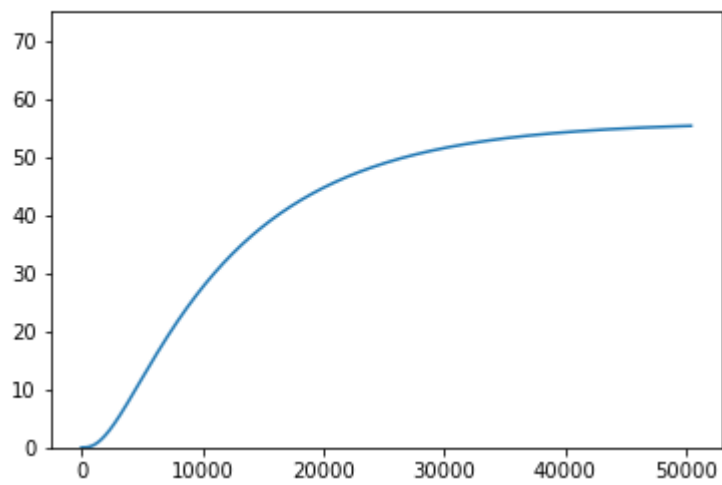
condition 113



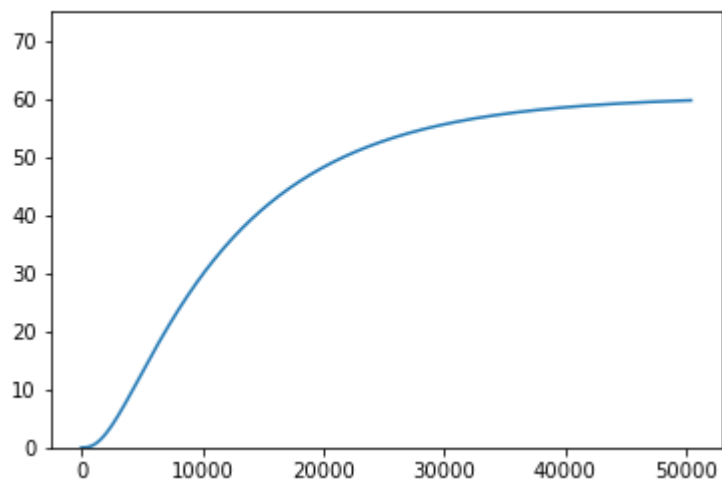
condition 114

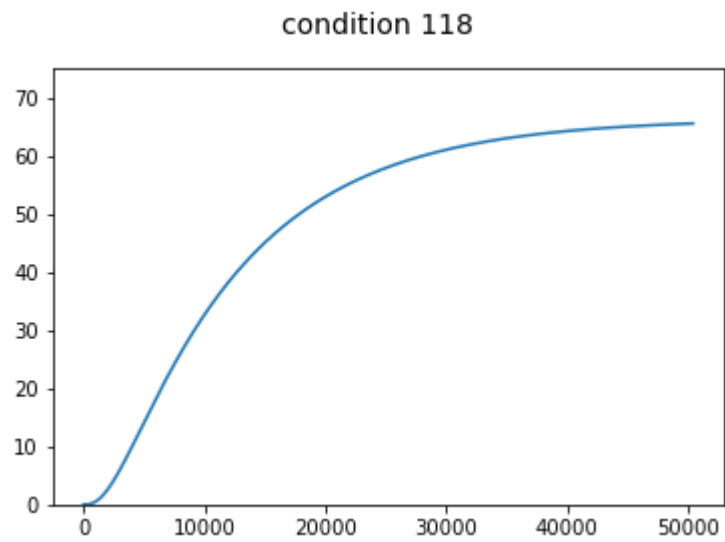
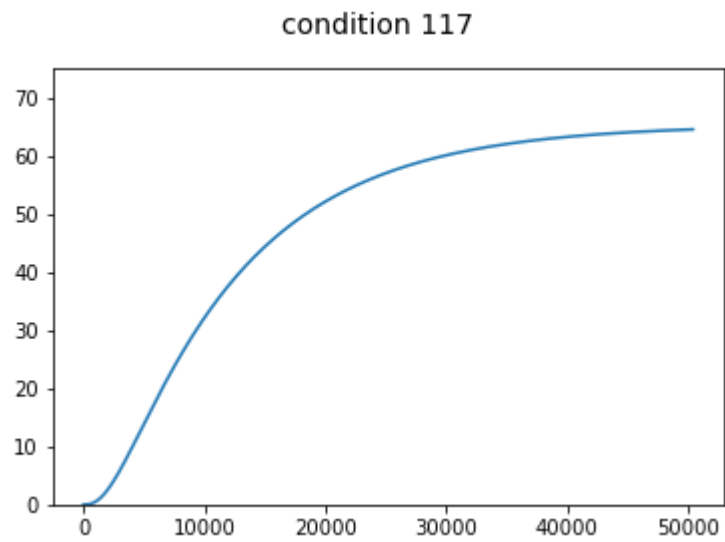


condition 115



condition 116





```
In [ ]:
```

```
In [ ]:
```