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 = 0.019905
            k1 = 0.08299
            d2 = 0.116948
            k2 = 0.001023
            Kd = 90.41
            n = 0.964487
            k3 = 0.000432
            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
            dmRNAdt = k3*Pa - d3*mRNA
            dmCherry1dt = k4*mRNA-(d4 + k5)*mCherry1
            dmCherry2dt = k5*mCherry1-d4*mCherry2
            return [dPudt,dPbdt,dPadt,dmRNAdt,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])
```

```
39.60732600549476
18.91702951710456
10.545627852390755
0.001574375945156159
0.0007741748803592678
47.01596823409629
24.423506987455212
14.084402713139566
0.0010579731227356025
54.35014575094039
30.903869640067363
18.571960706409854
5.886439523589754
0.0014372659141017733
0.0014372624817546326
55.629357515101425
32.159898946453424
19.485390780676255
0.0015172860345678944
0.00151727807296545
33.714894554543896
11.468889898139992
5.737235478100897
1.9165290914783337
0.637828319648861
38.70568830948649
13.192049705235844
6.599151666166402
2.2042895257000064
0.18949502775026827
0.010924210765730434
43.291080334722345
14.778626463859293
7.39272707898729
2.4691905723211156
0.012236278911742999
0.05143490822965255
44.05743833925751
15.044095176779686
7.525507362519572
2.513509635419496
0.8364008100883737
38.17010436087787
28.6497412478985
9.49364450060142
4.311262337048308
1.1313551138601943
0.04039950313488095
```

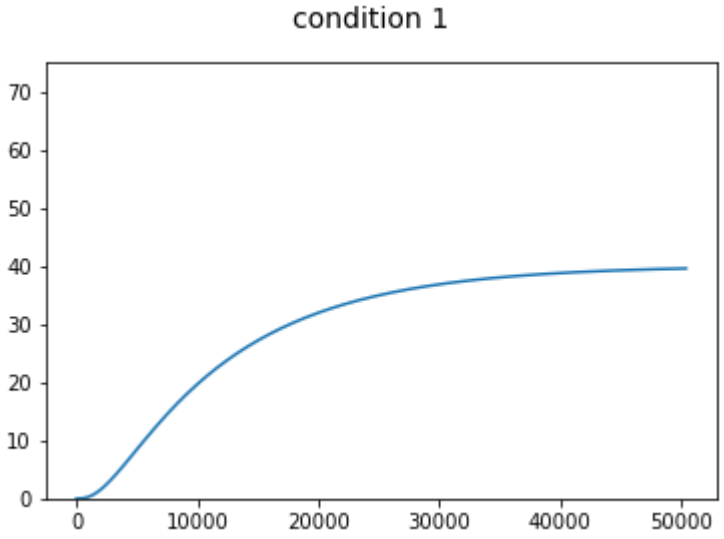
```
43.06331800434194
32.32287863944693
10.710048129579494
4.863322687593309
1.2761923556229955
0.04557147757849664
47.45274585963902
35.61790623682818
11.801087808835215
5.358422711466287
1.4060791770225003
0.05020956589559065
48.176977751880926
36.16157328223854
11.981091917674274
5.440100369065236
1.4275062782370294
0.050974705514409704
47.45544132100602
27.892358914326262
13.061373752921575
3.4692167200188315
1.3955114989274984
0.12391275818324514
53.32565287221827
31.341290413603073
14.67518595609041
3.897733720027067
1.567883802760046
0.13921828221656624
58.55488508770176
34.41337808303734
16.112438217569977
4.279345590494021
1.7213881794677104
0.1528485071073405
59.41442991182297
34.91832375541485
16.34865265234915
4.342061903326818
1.7466159910009487
0.15508858812777349
51.05199520954722
35.82510503856569
27.337563580940994
7.495603435556816
3.017030186168442
0.26789787418183053
57.305948376975756
40.2116006406189
30.683669888497146
8.412688278277729
3.386159941909385
0.30067481509962146
62.86614613701633
44.11111132943395
33.65808514747393
9.227833979261339
3.7142584204562086
0.3298083456888808
63.77912026726536
44.75136858598275
34.14643263757913
9.361660784400044
3.768124048250364
```

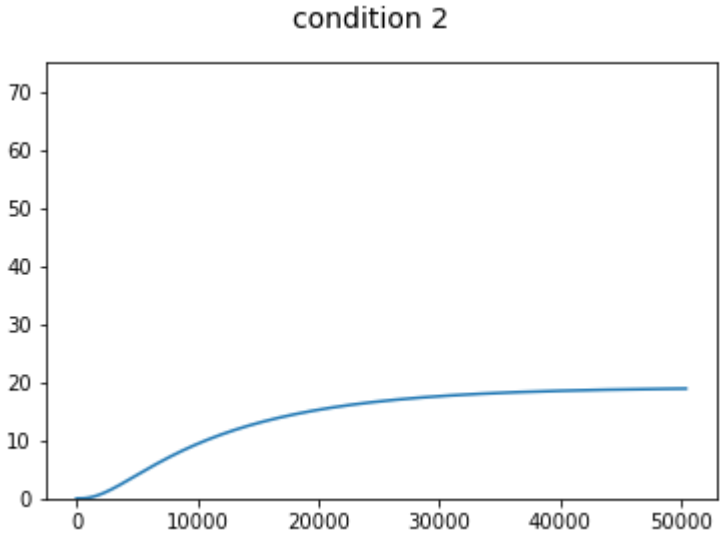
```
0.3345913978919848
54.35570778869119
60.95158251595553
66.80452106914996
67.76455482632973
```

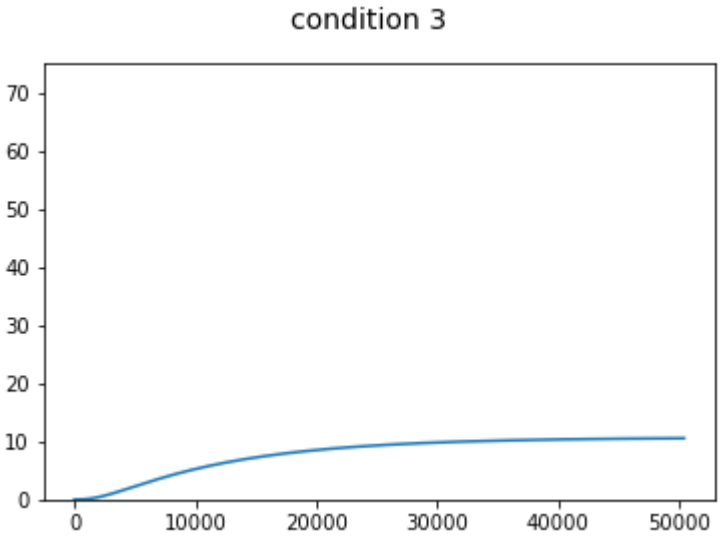
In [4]:

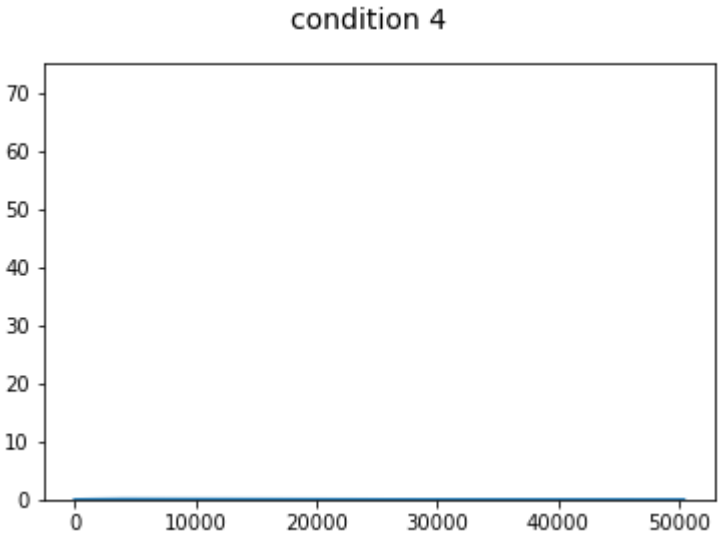
```
import csv
pp = PdfPages('multipage.pdf')

#for i in range(42):
for i in range(len(lightdata[:,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,lightdata[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])]
```

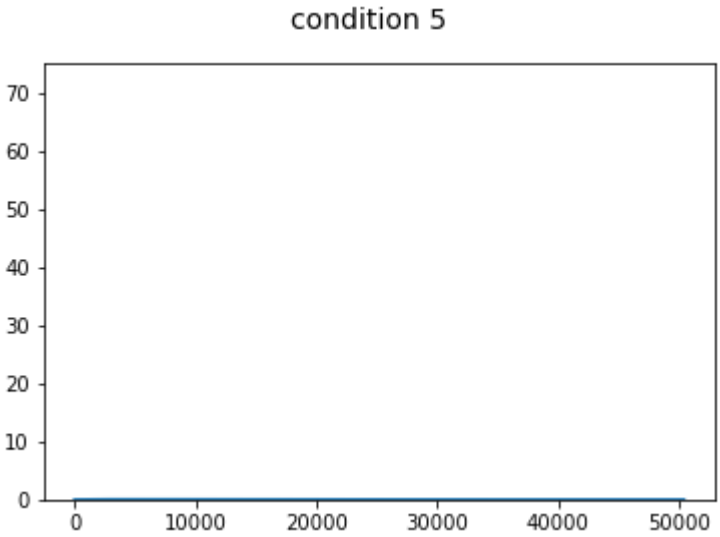
```
    #    ymax = mCherry2[i].max()
    #    text = "t={:.3f}, y={:.3f}".format(tmax,ymax)
    #    print(i+1, ymax)
    #    if not ax:
    #        ax=plt.gca()
    #    bbox_props = dict(boxstyle = "square,pad=0.3", fc="w", ec="k",lw=0.72)
    #    arrowprops = dict(arrowstyle="->", connectionstyle = "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()
```

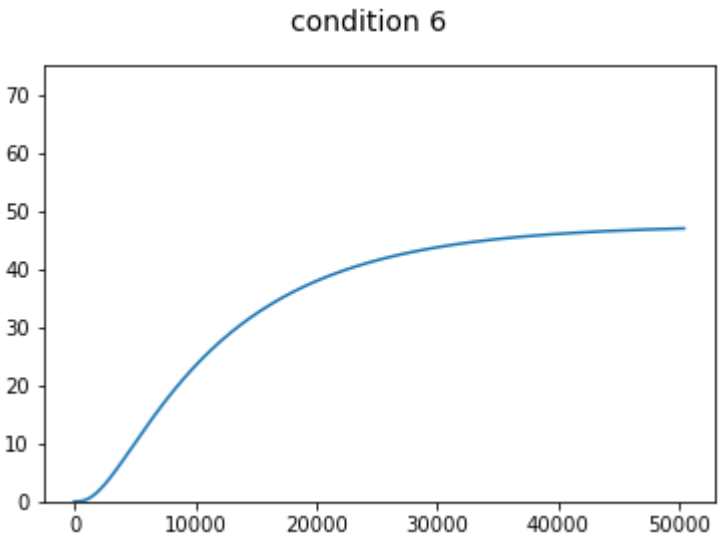


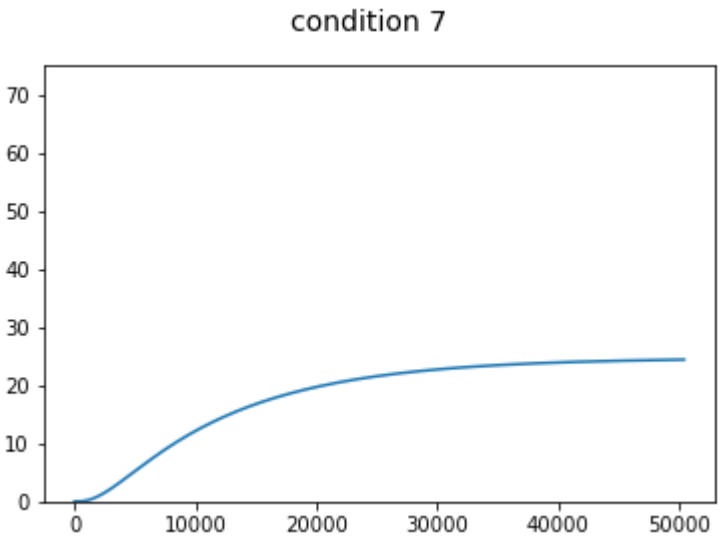


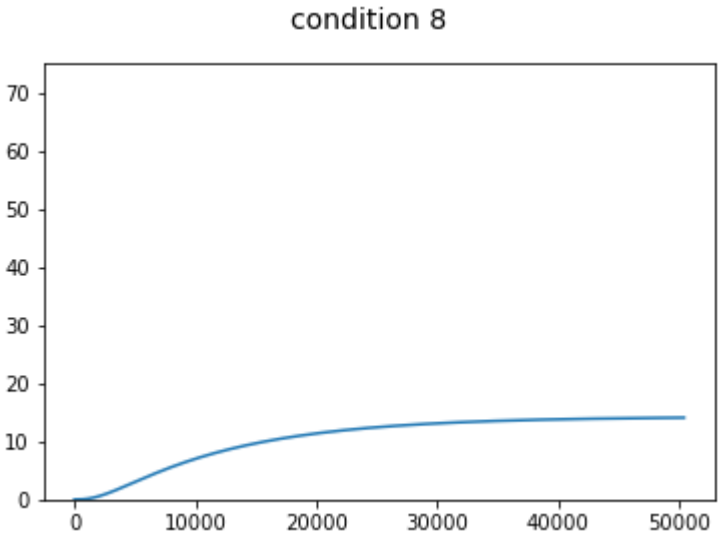

condition 3
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

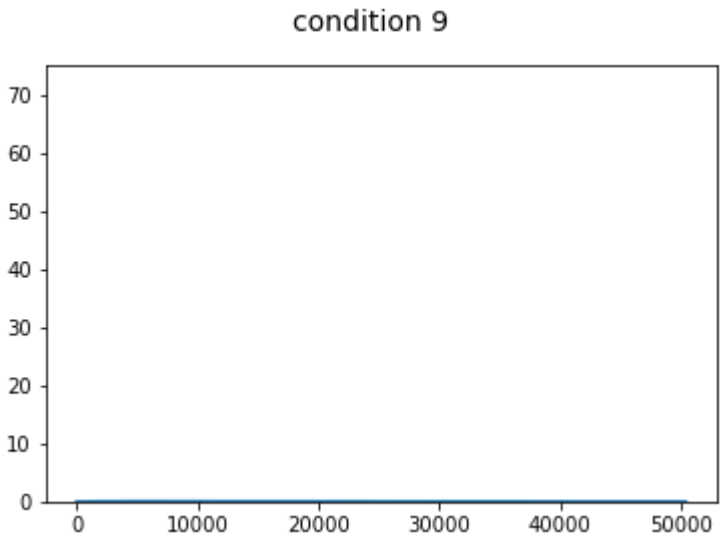

condition 4
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

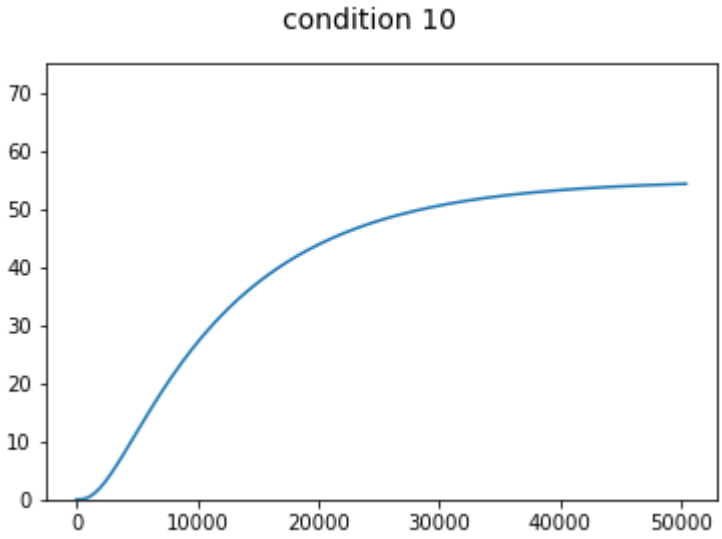

condition 5
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

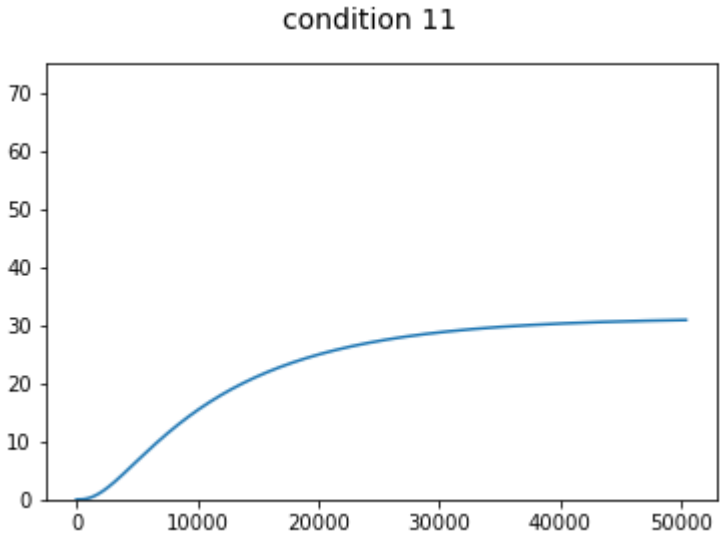

condition 6
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

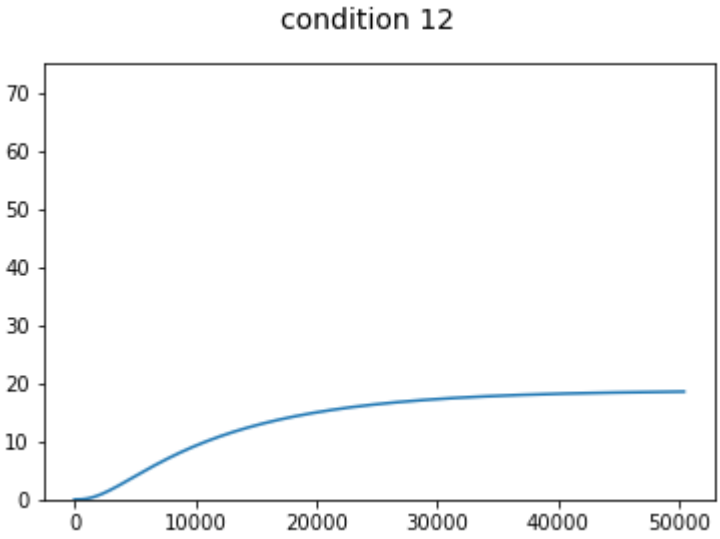

condition 7
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

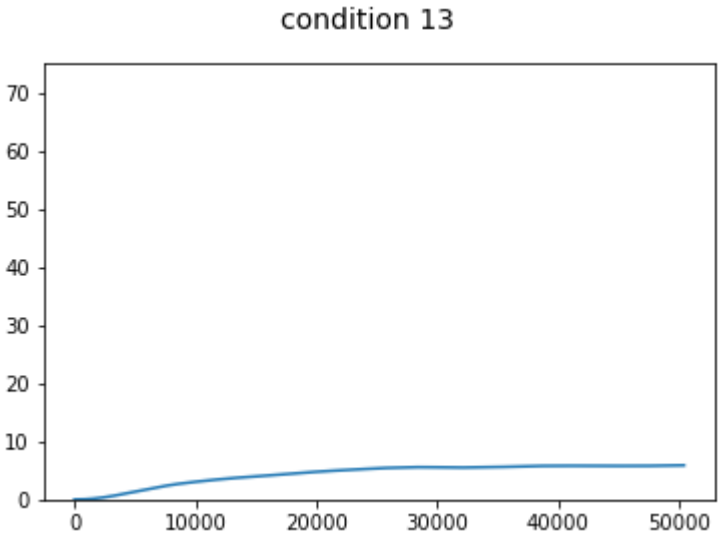

condition 8
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

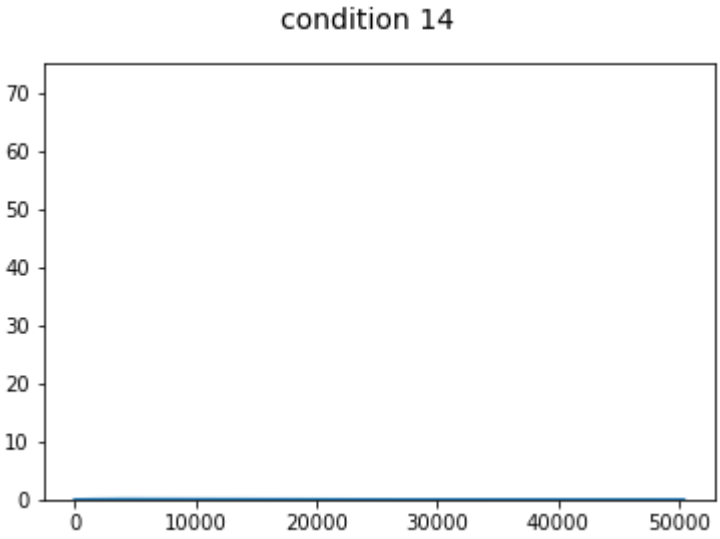

condition 9
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

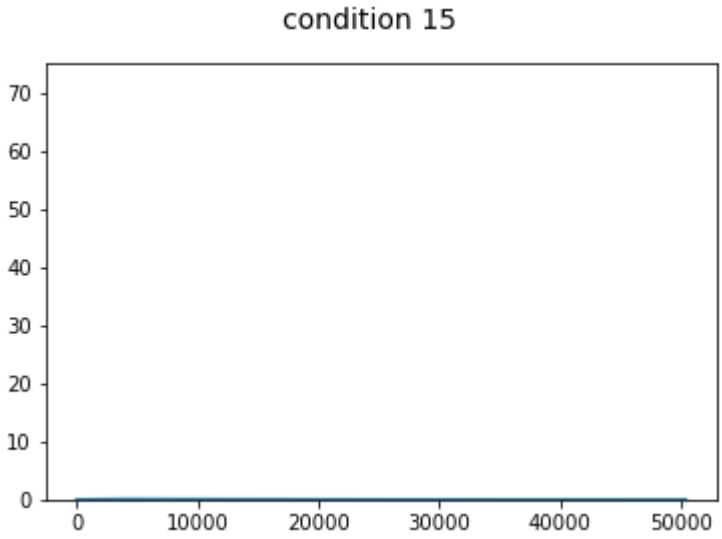

condition 10
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

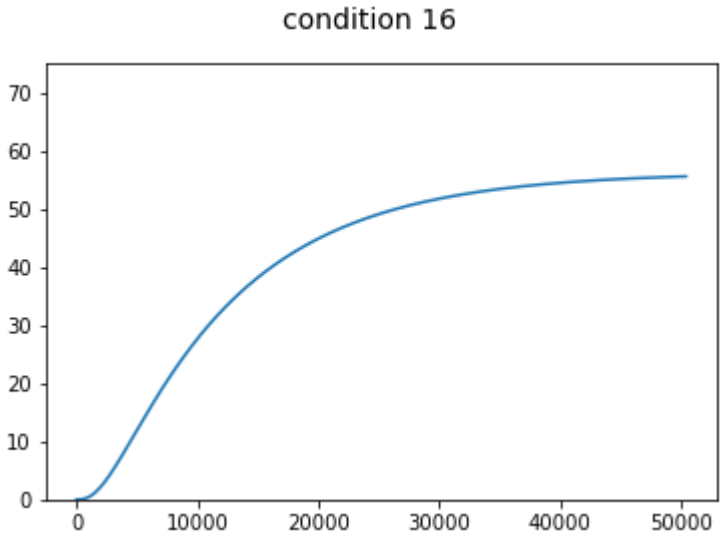

condition 11
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

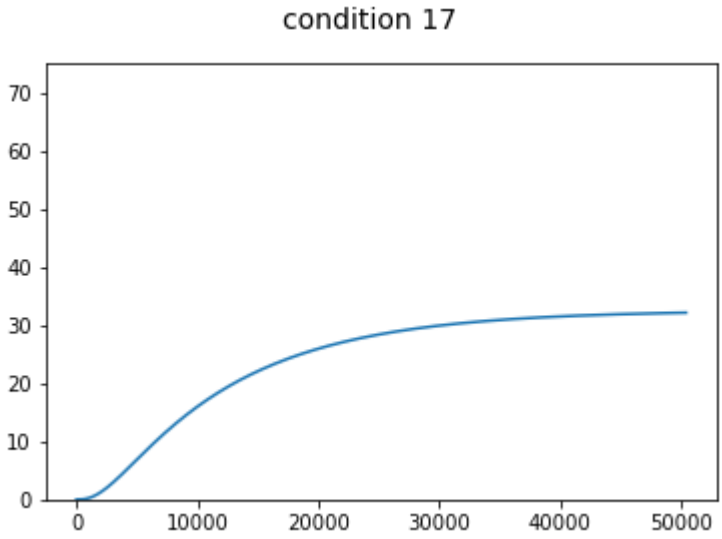

condition 12
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

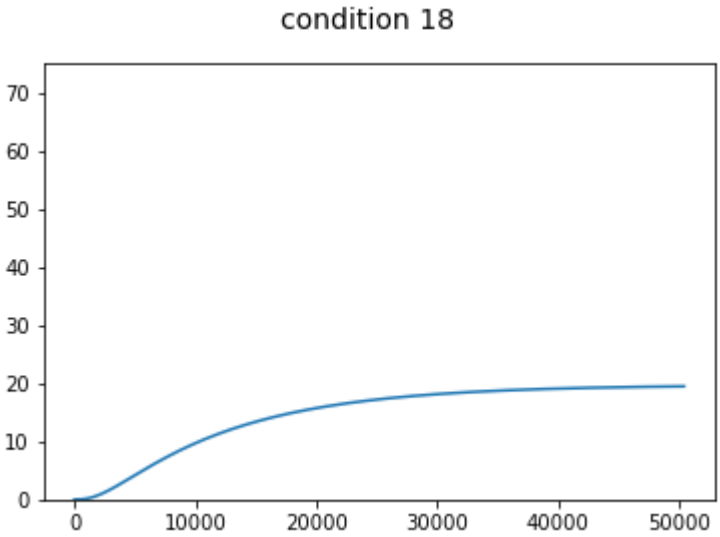

condition 13
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

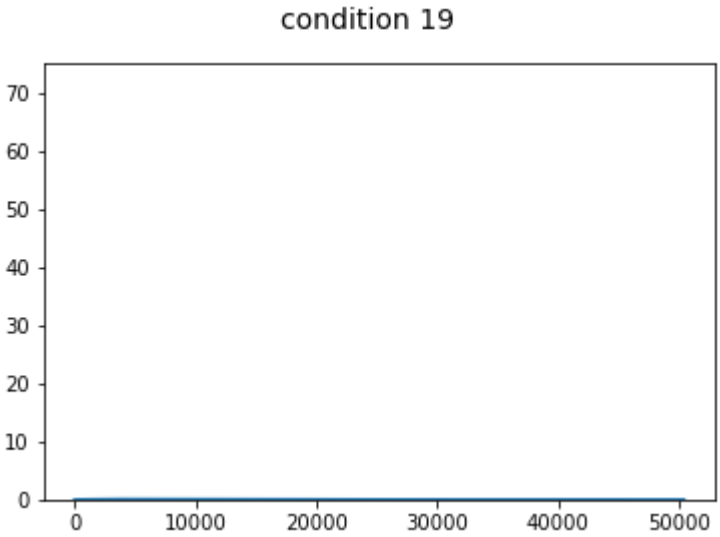

condition 14
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

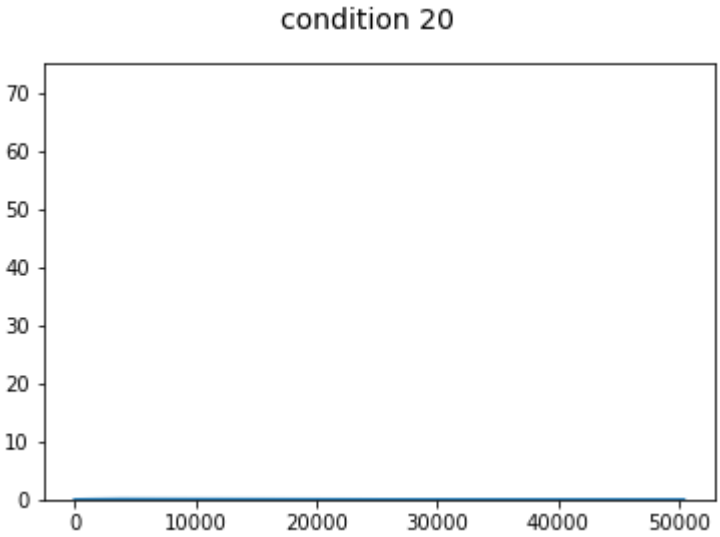

condition 15
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

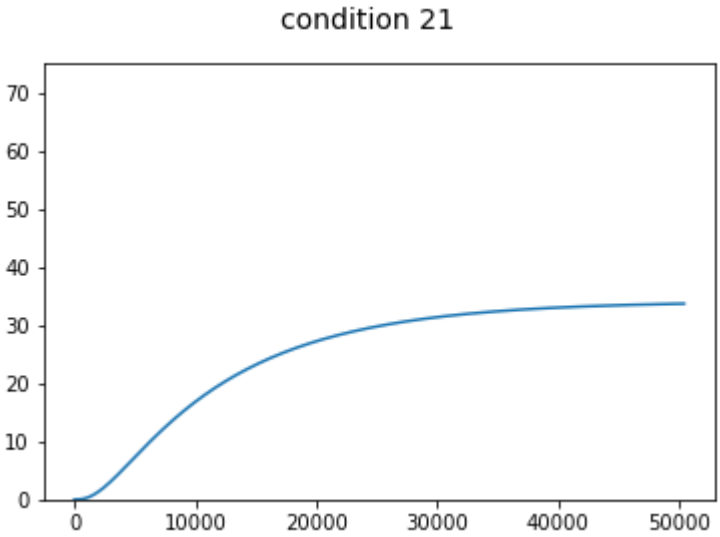

condition 16
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

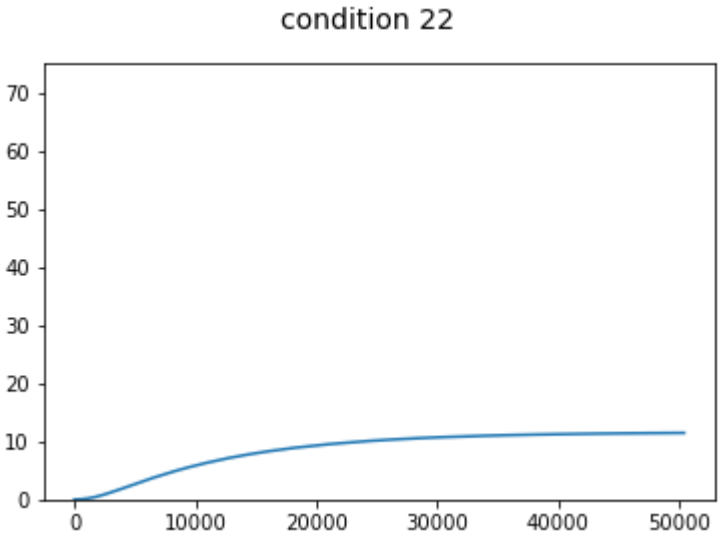

condition 17
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

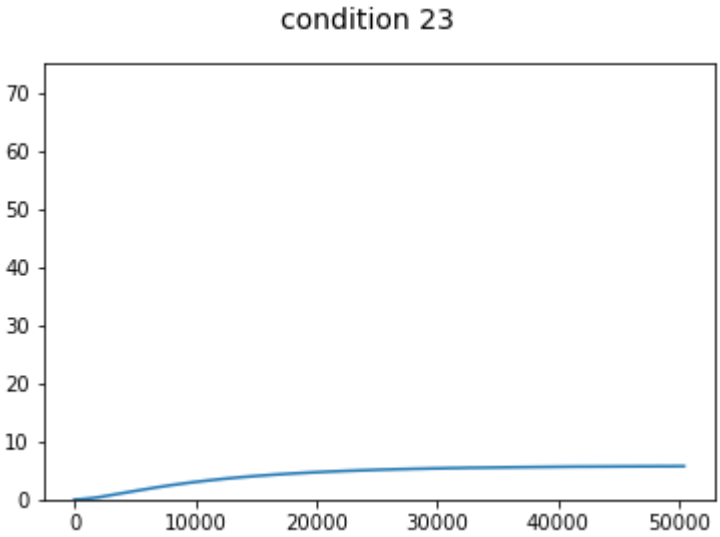

condition 18
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

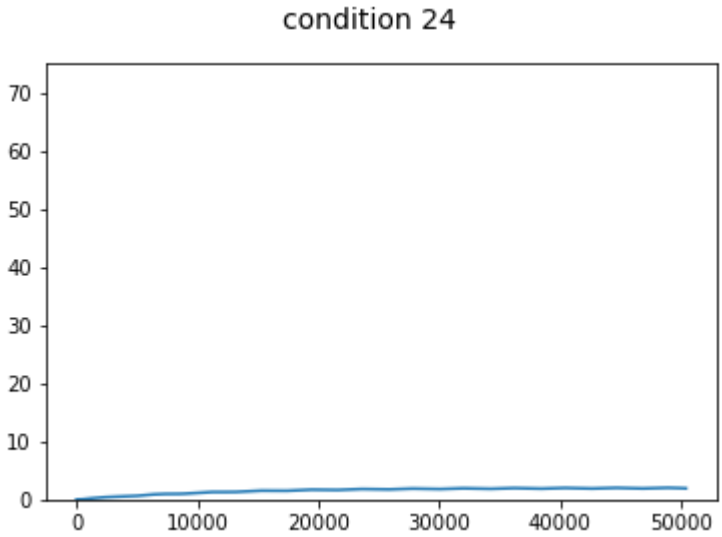

condition 19
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

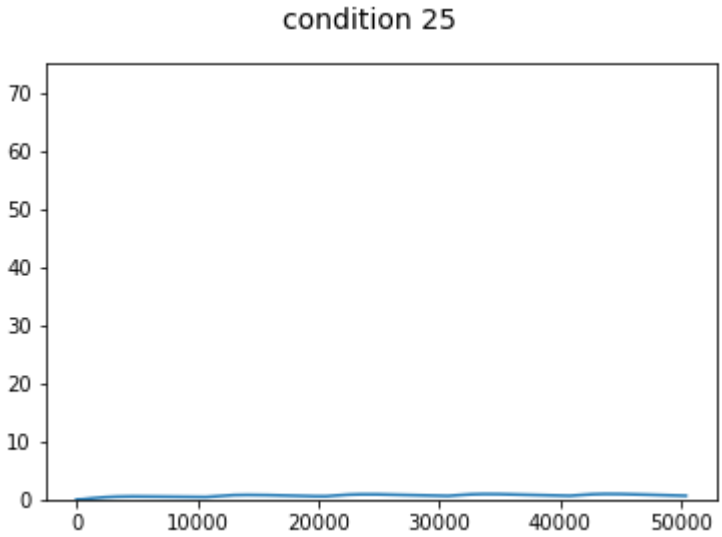

condition 20
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

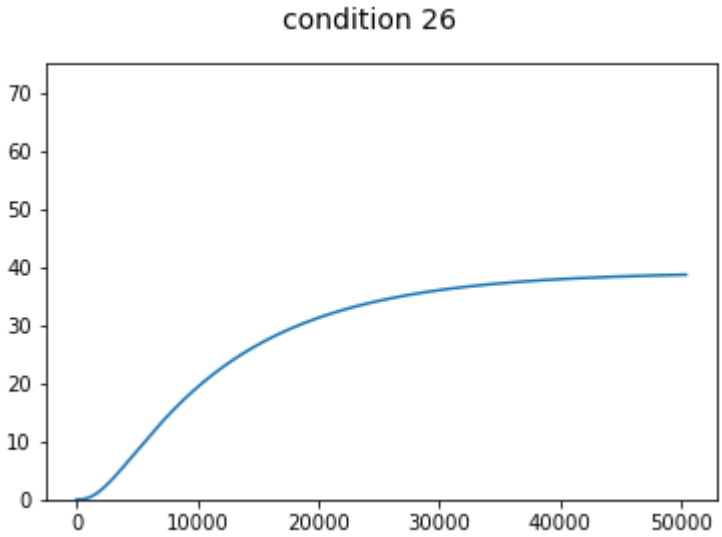

condition 21
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

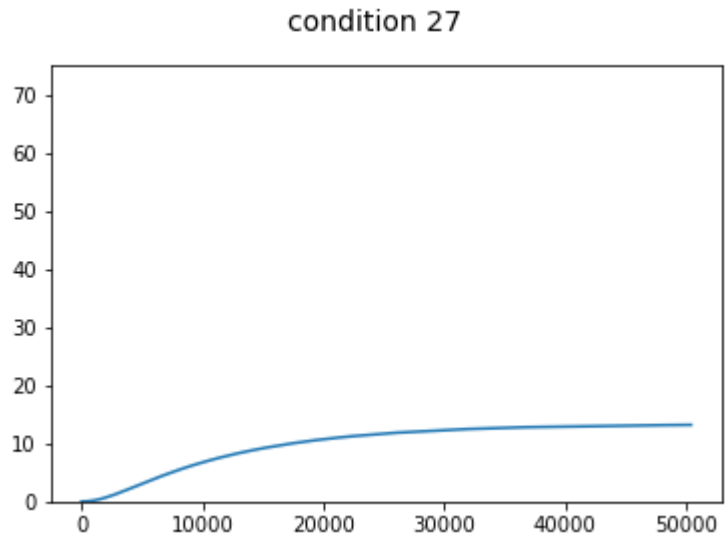

condition 22
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

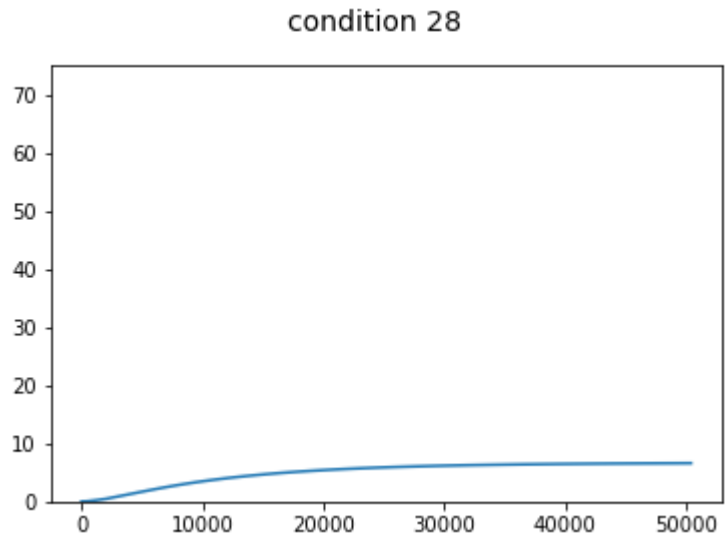

condition 23
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

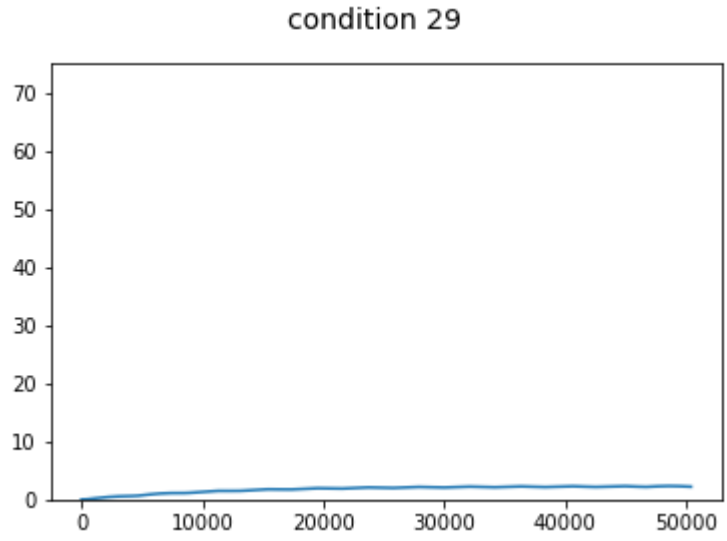

condition 24
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

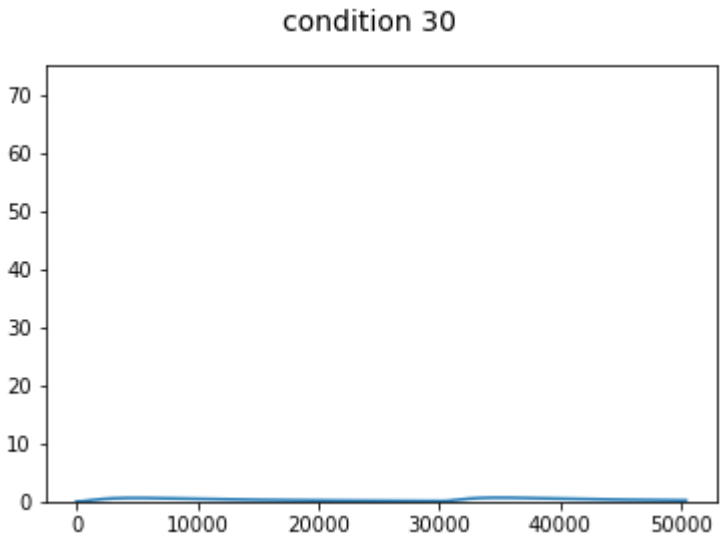

condition 25
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

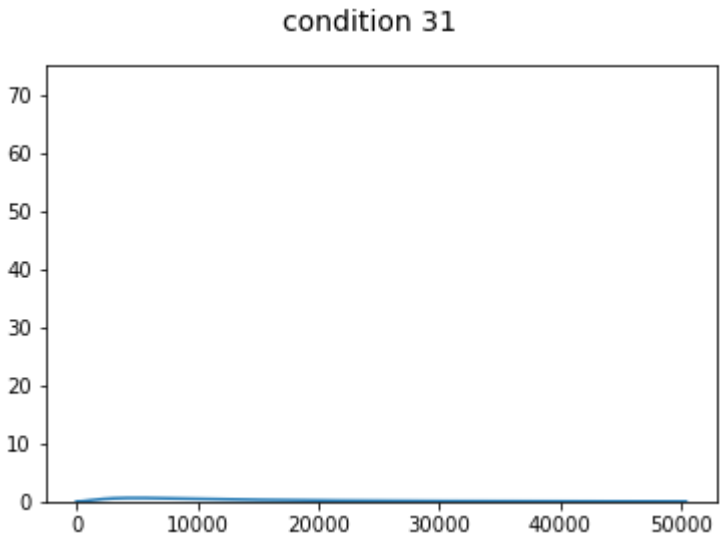

condition 26
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

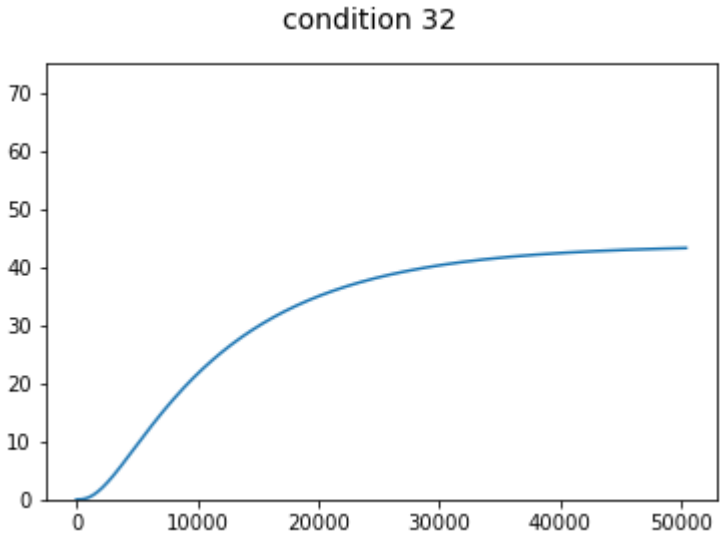

condition 27
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

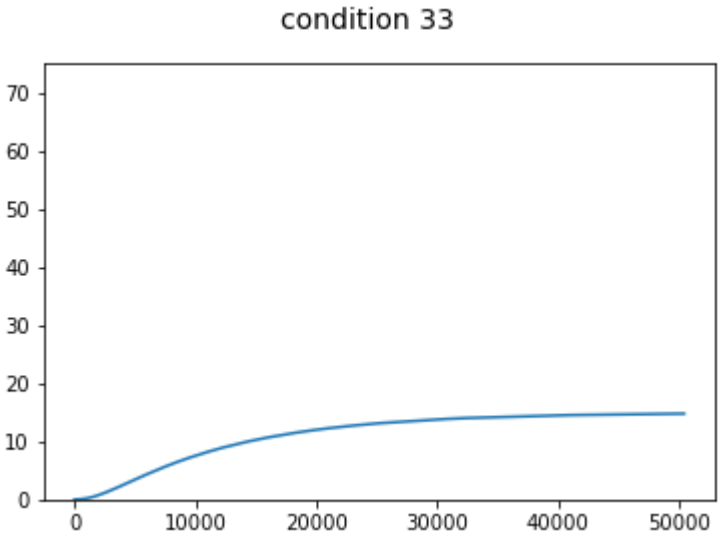

condition 28
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

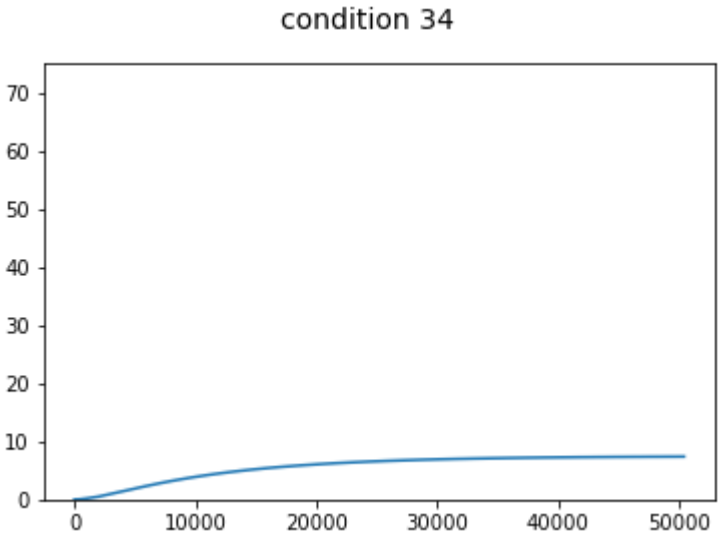

condition 29
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

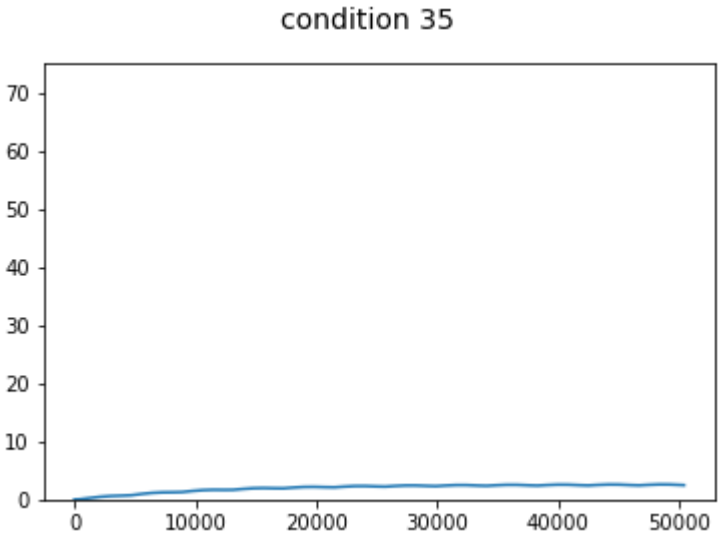

condition 30
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

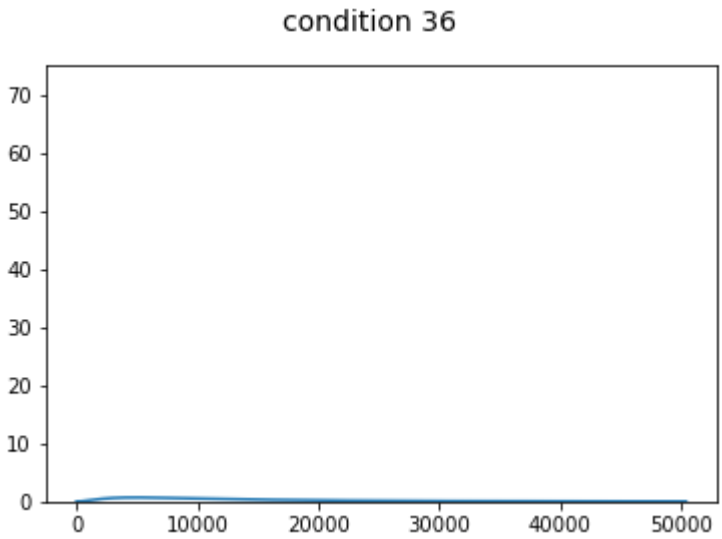

condition 31
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

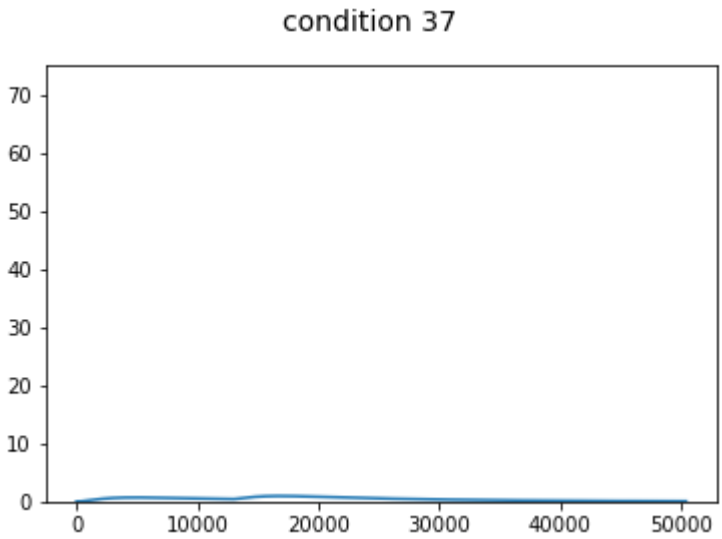

condition 32
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

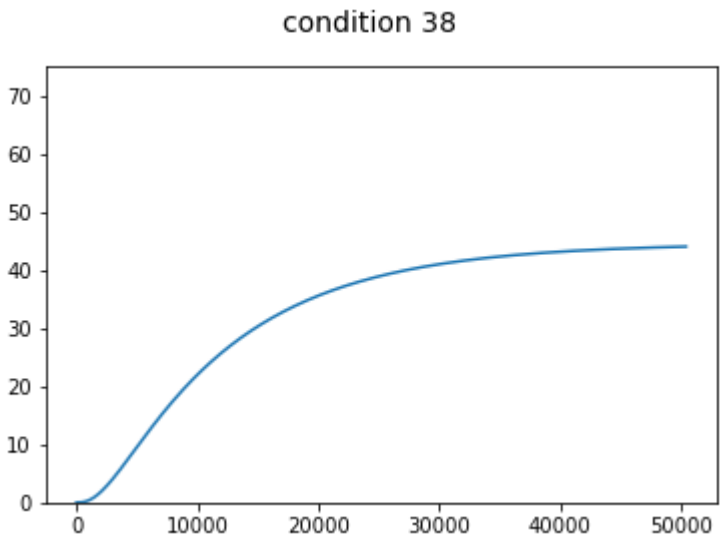

condition 33
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

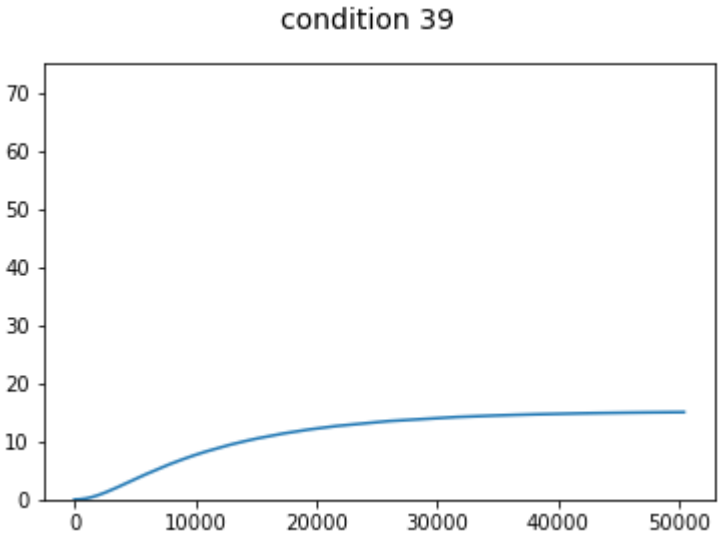

condition 34
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

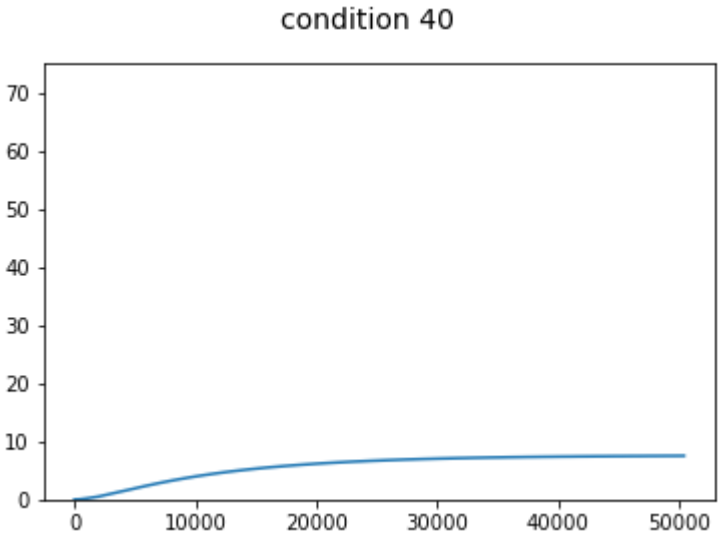

condition 35
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

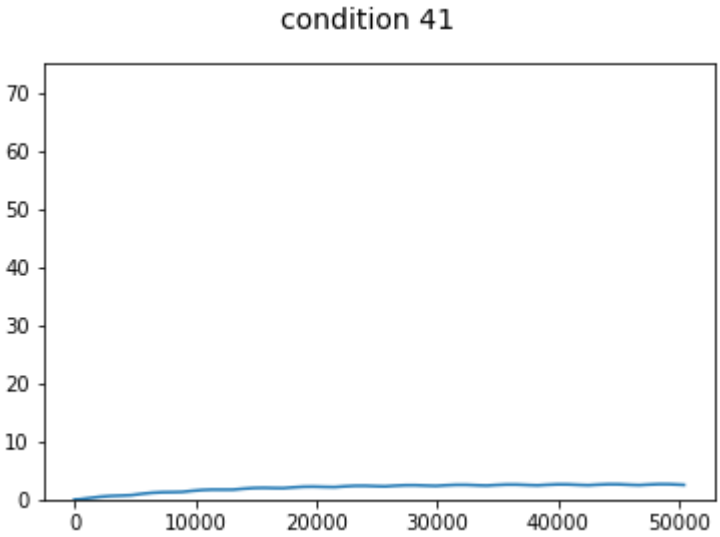

condition 36
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

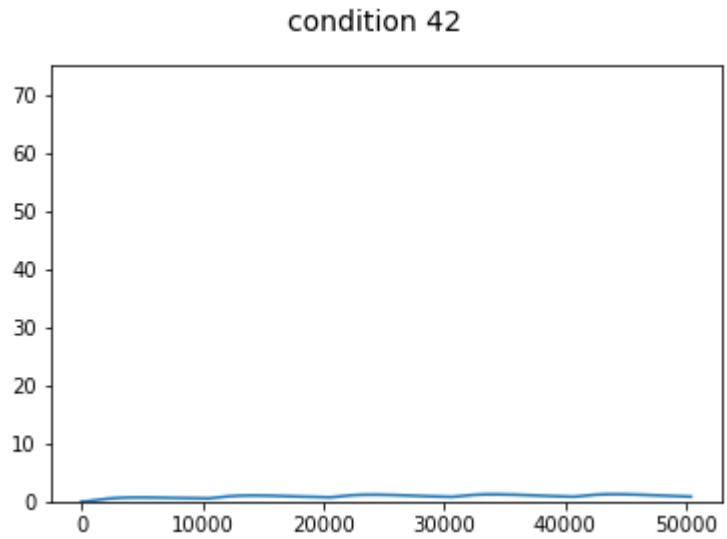

condition 37
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

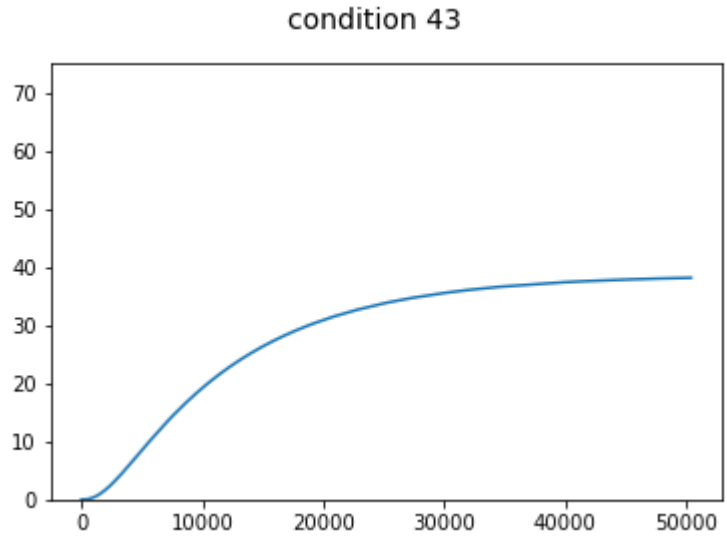

condition 38
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

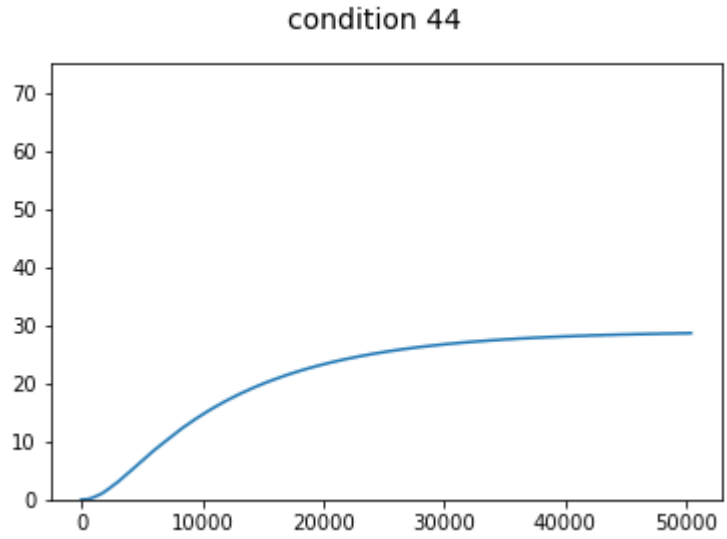

condition 39
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

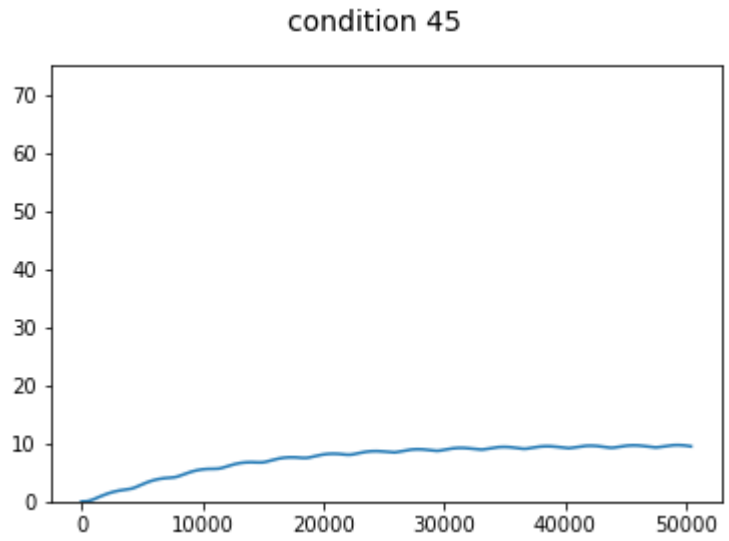

condition 40
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

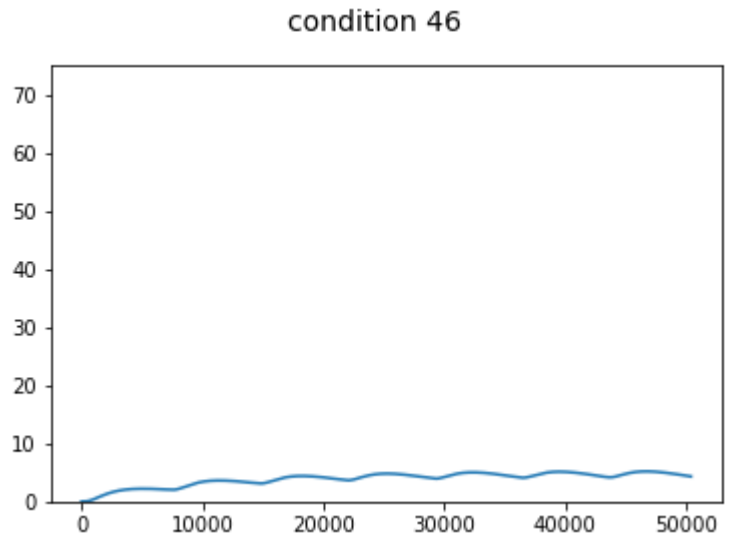

condition 41
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

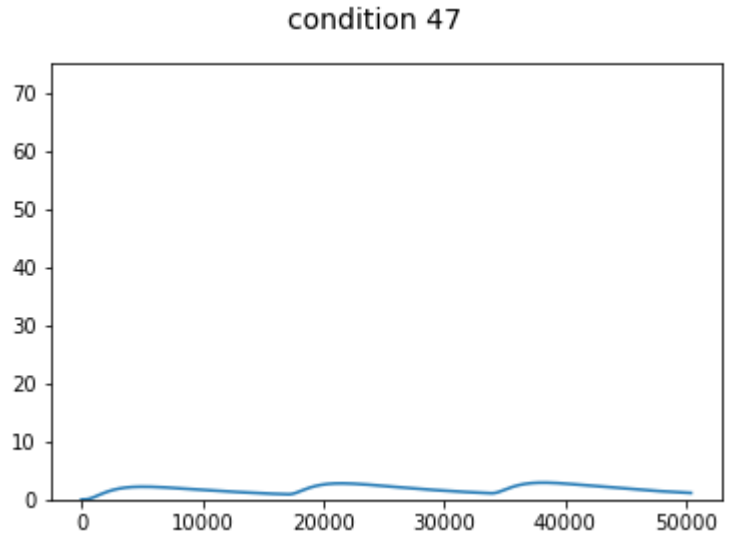

condition 42
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

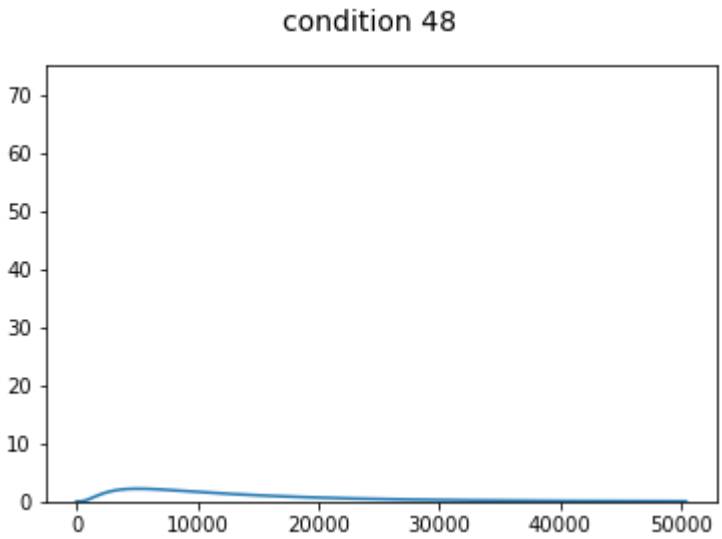

condition 43
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

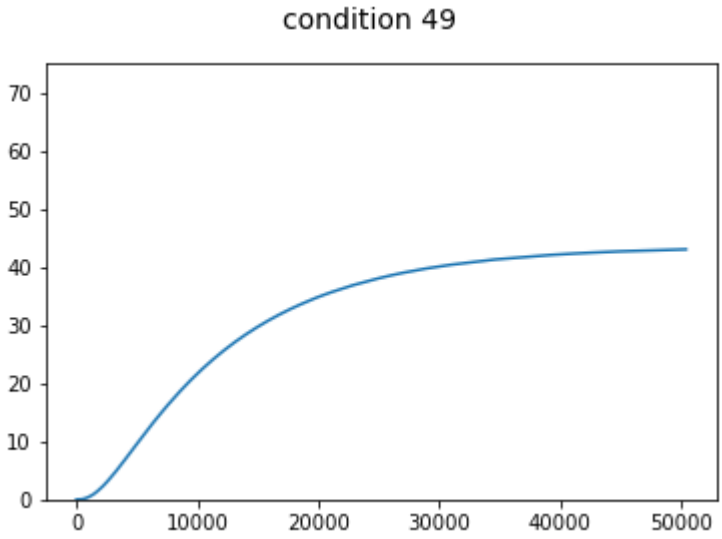

condition 44
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

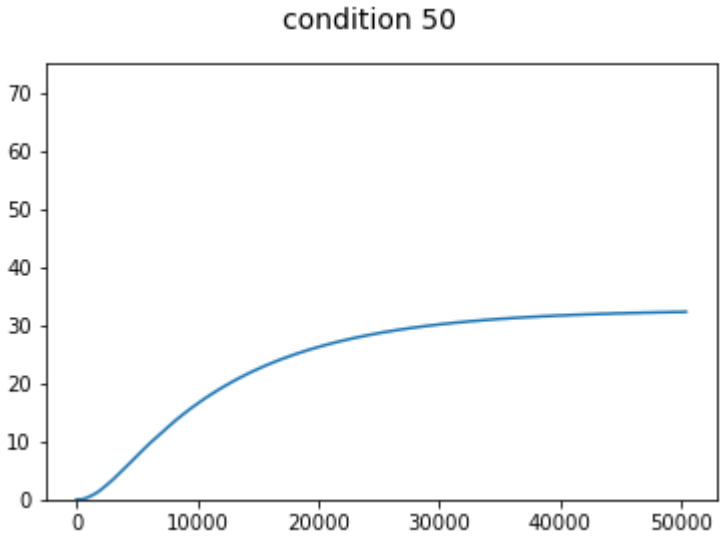

condition 45
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

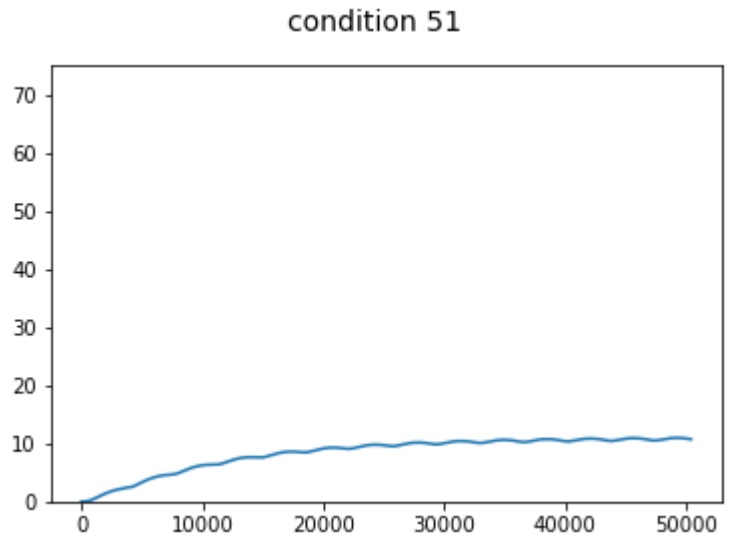

condition 46
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

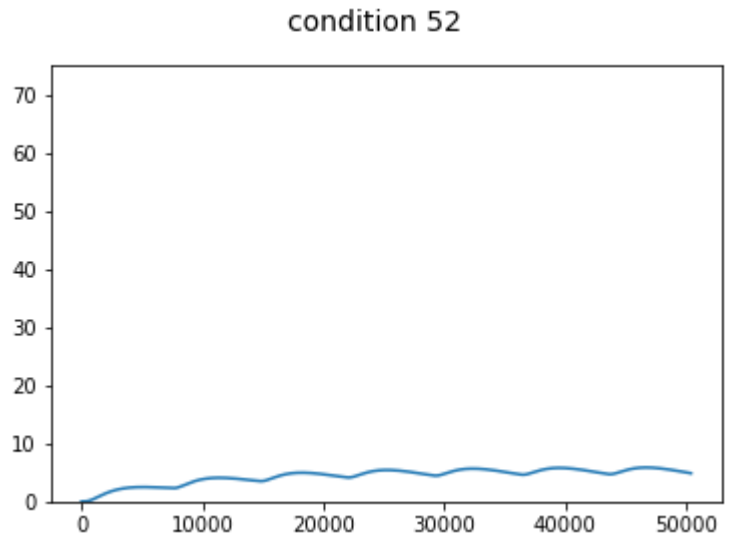

condition 47
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

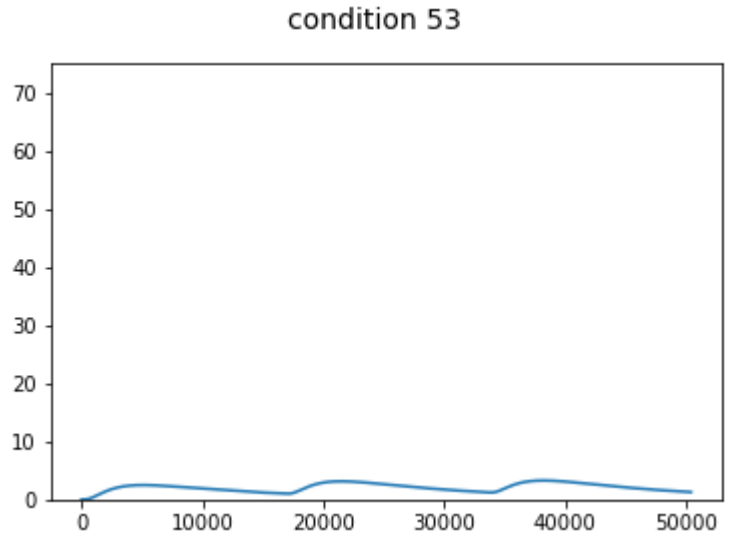

condition 48
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

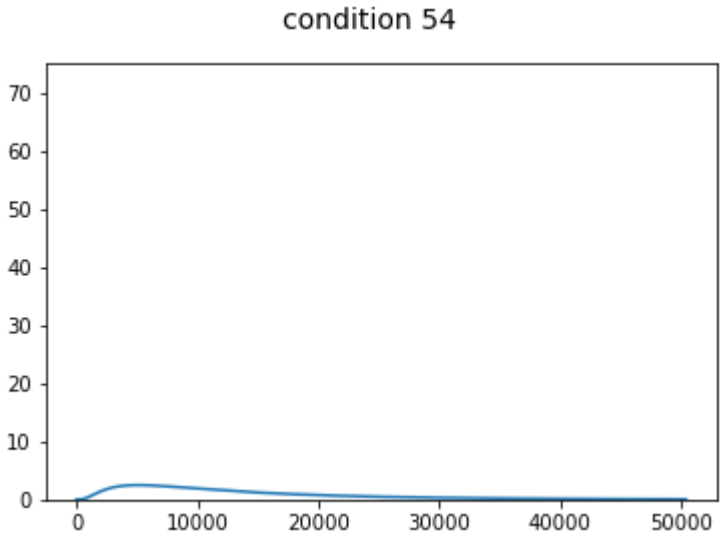

condition 49
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

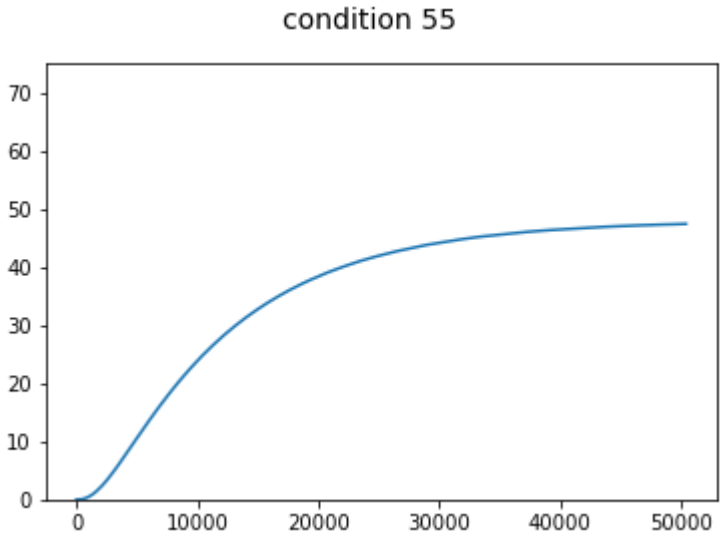

condition 50
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

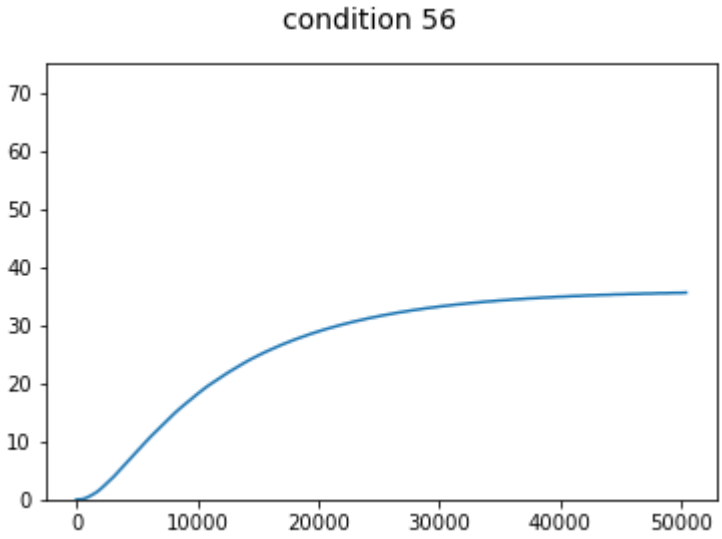

condition 51
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

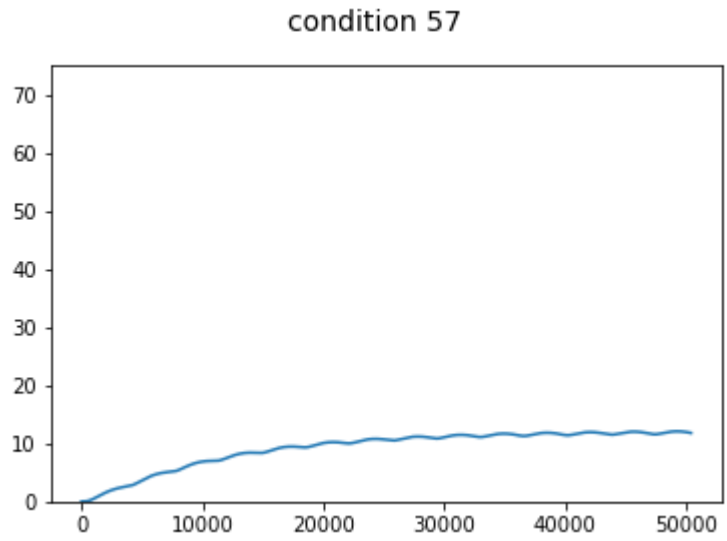

condition 52
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

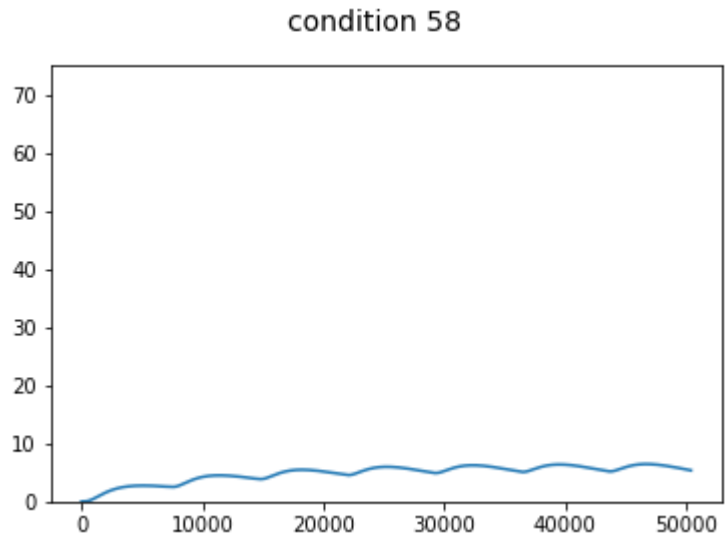

condition 53
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

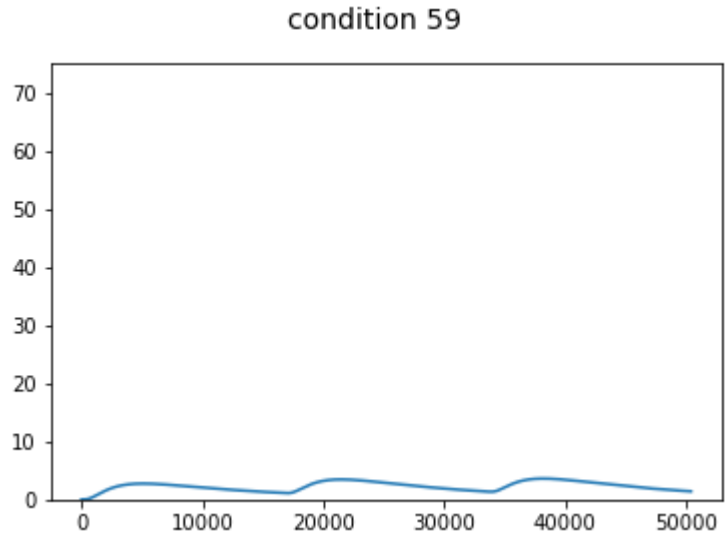

condition 54
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

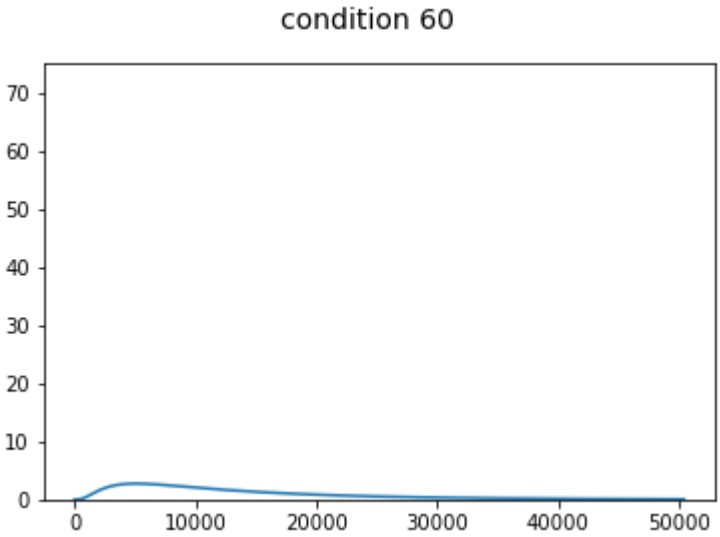

condition 55
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

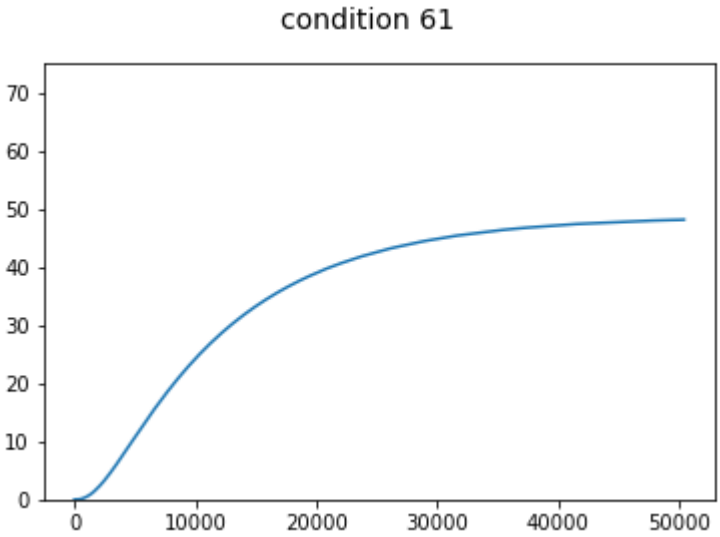

condition 56
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

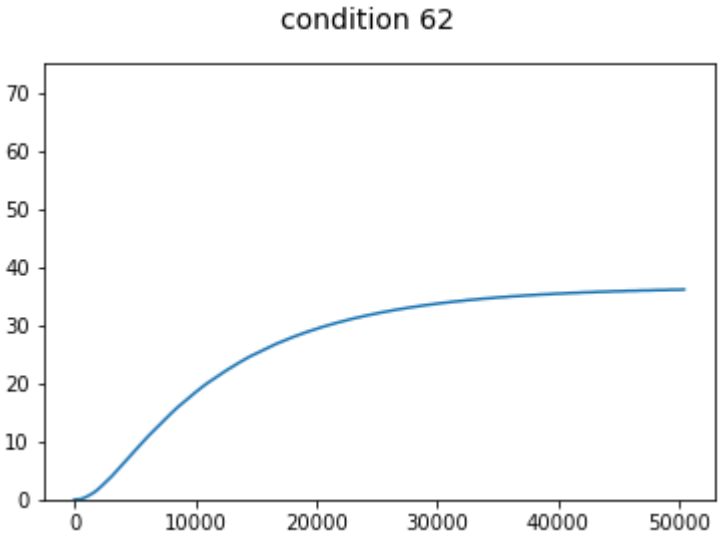

condition 57
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

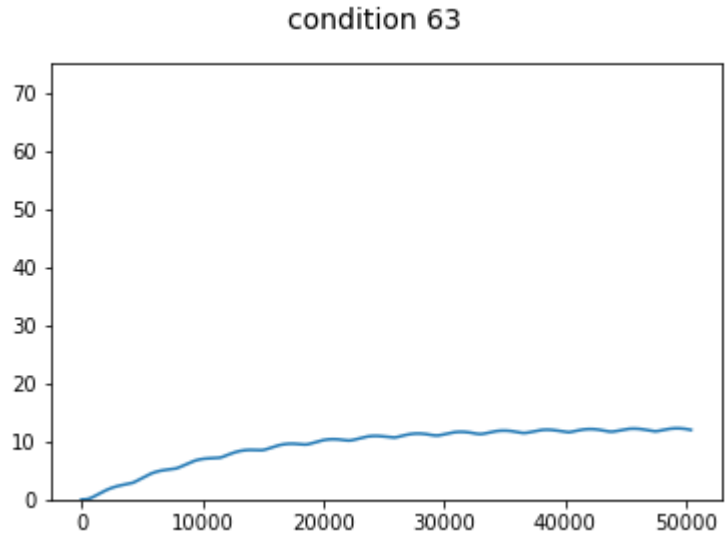

condition 58
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

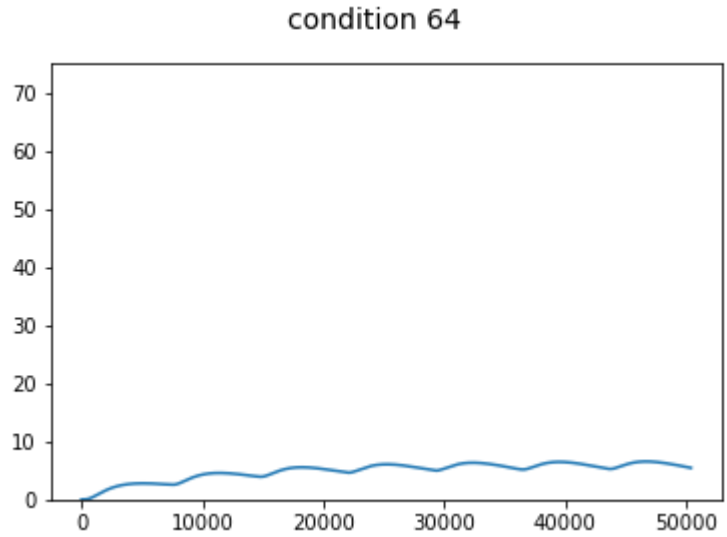

condition 59
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

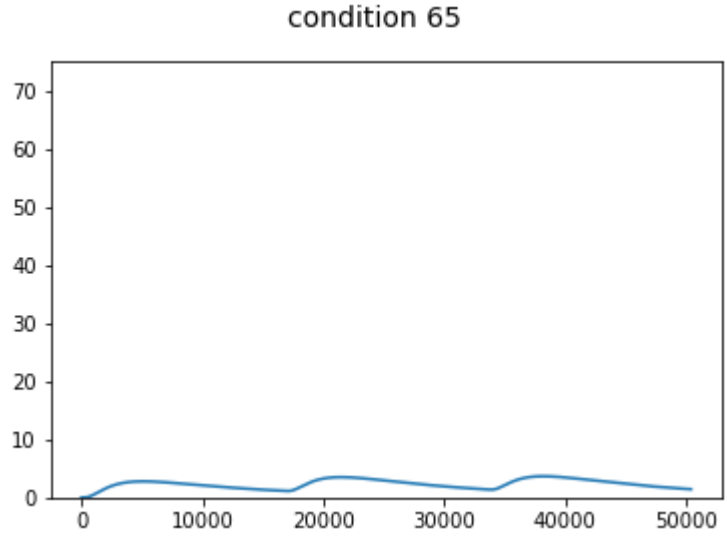

condition 60
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

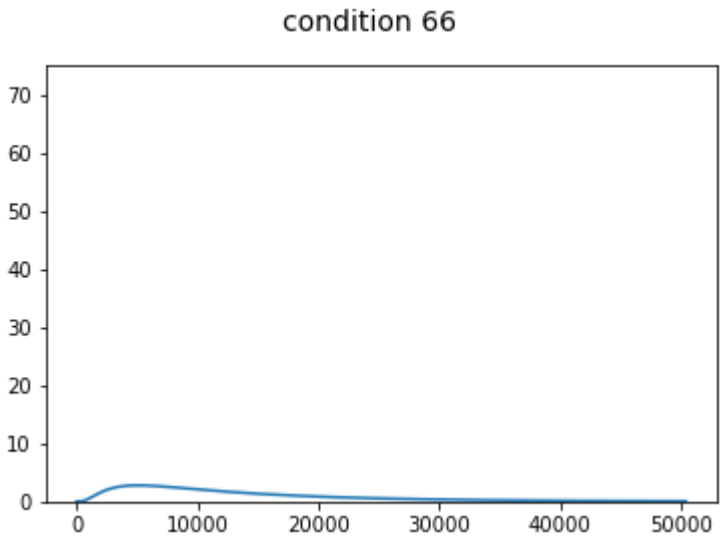

condition 61
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

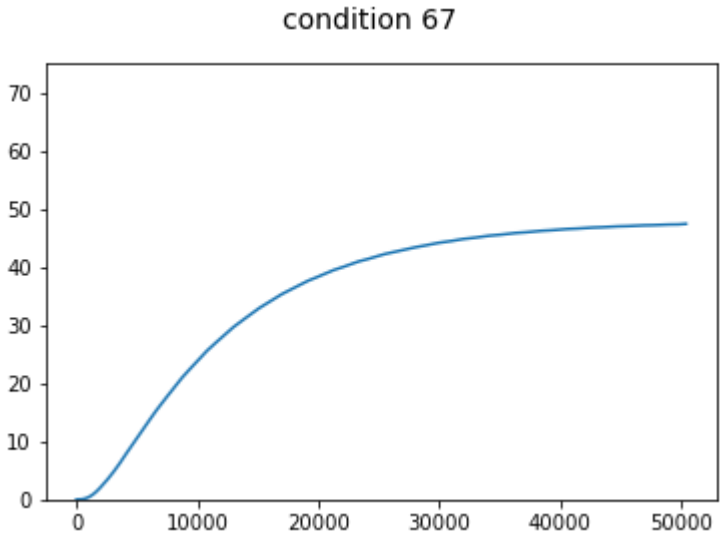

condition 62
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

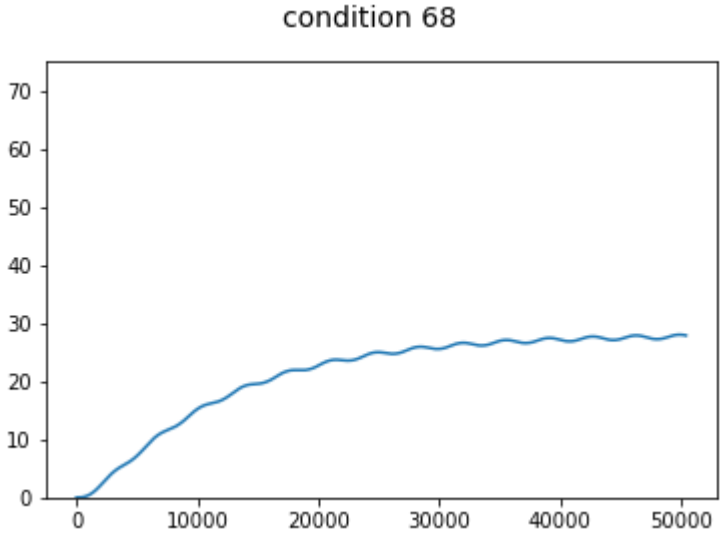

condition 63
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

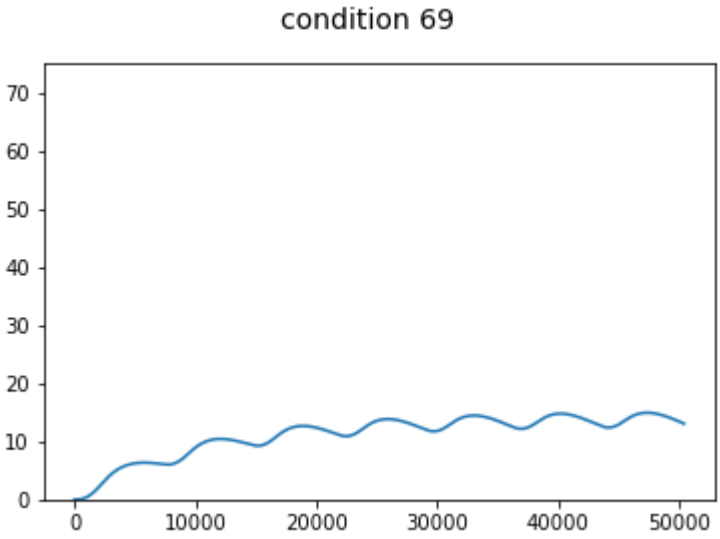

condition 64
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

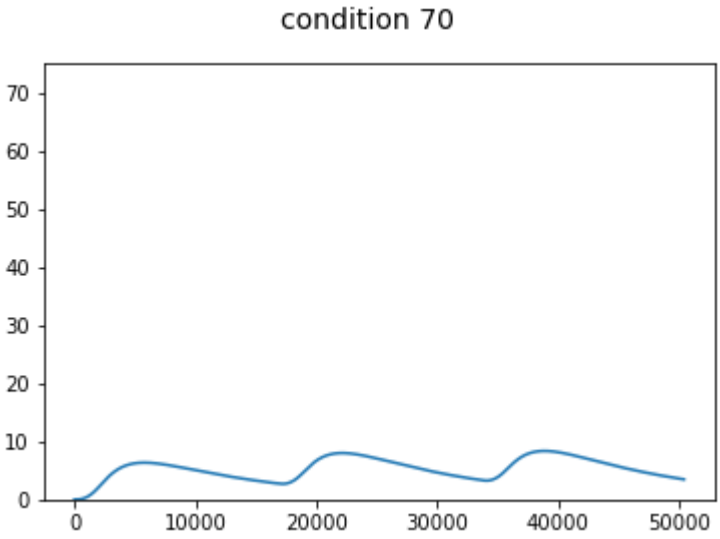

condition 65
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

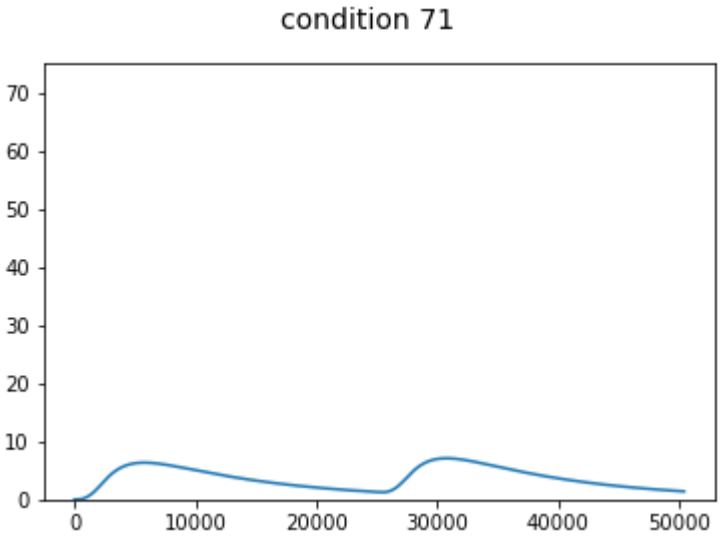

condition 66
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

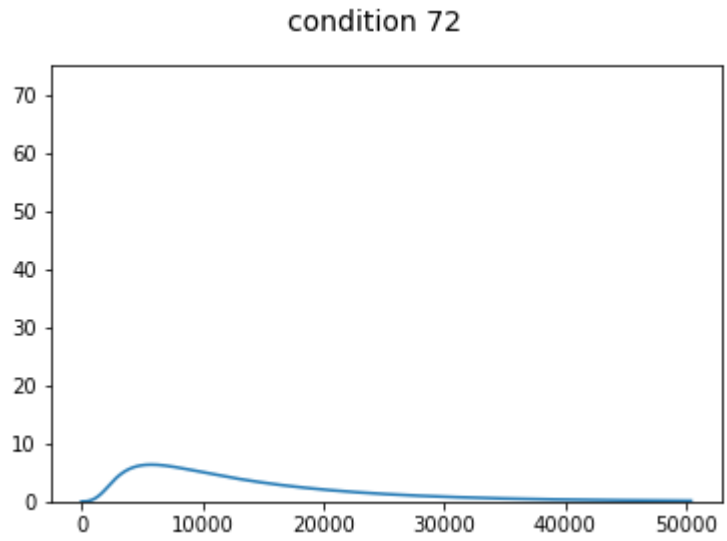

condition 67
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

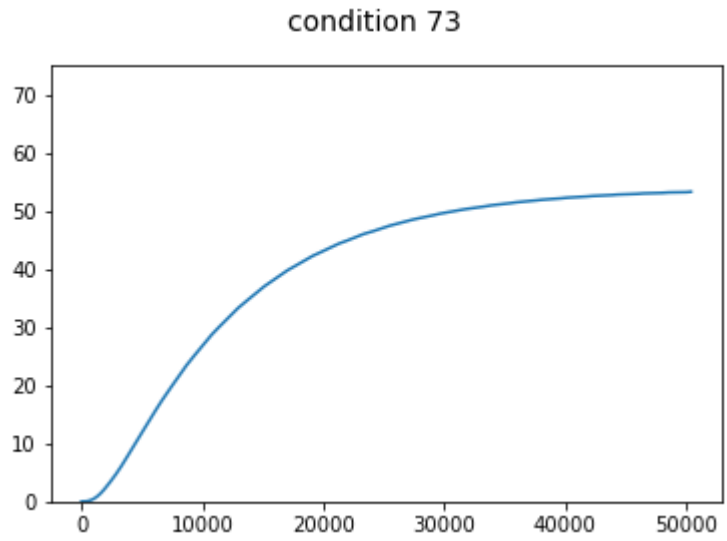

condition 68
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

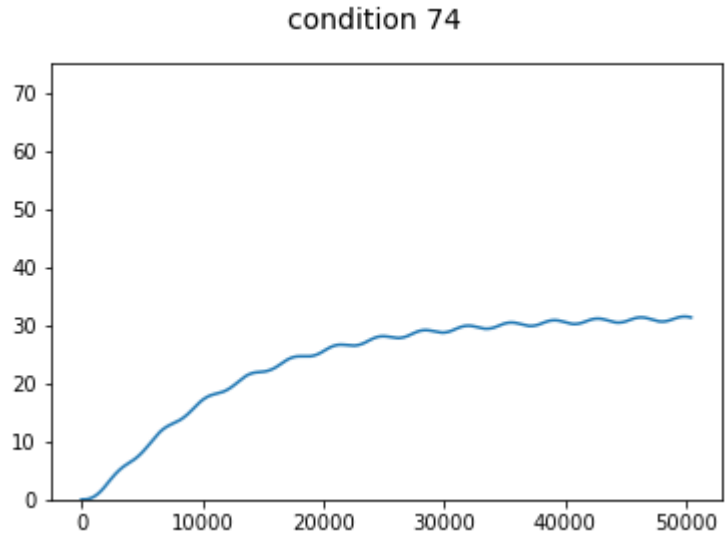

condition 69
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

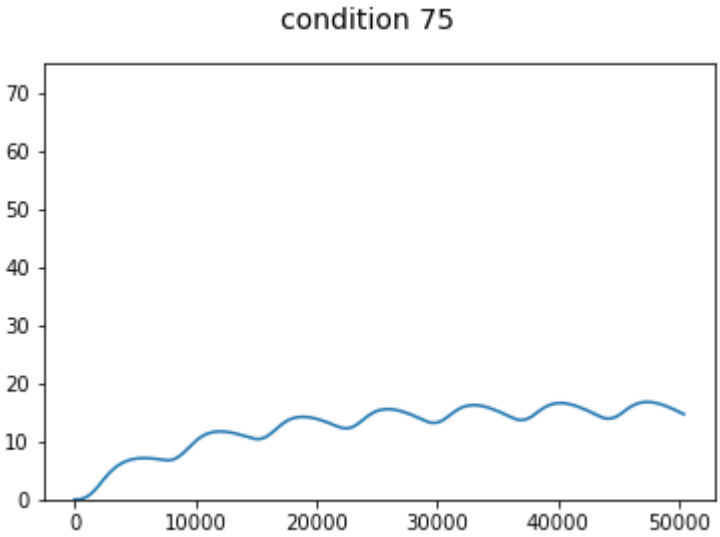

condition 70
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

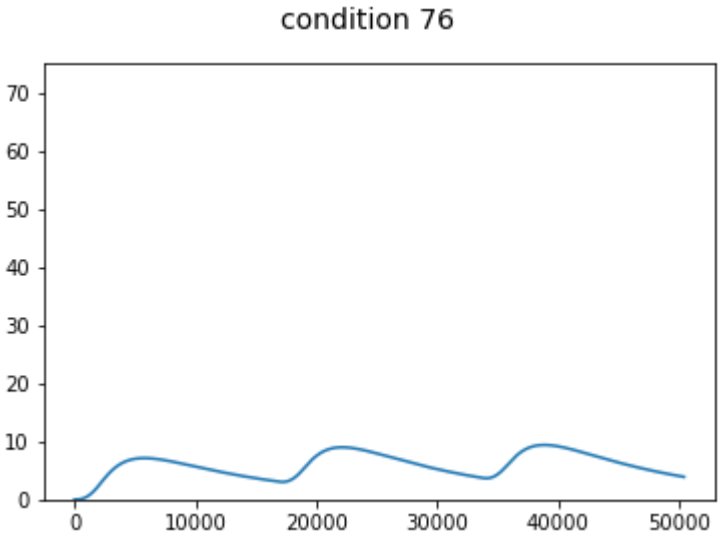

condition 71
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

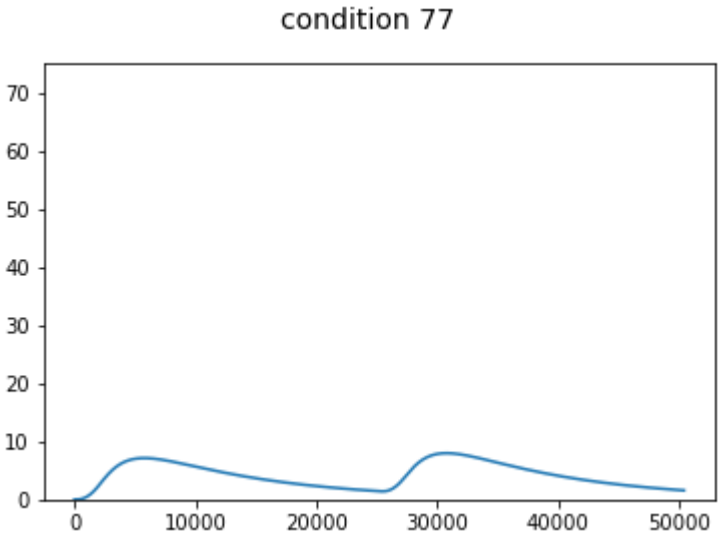

condition 72
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

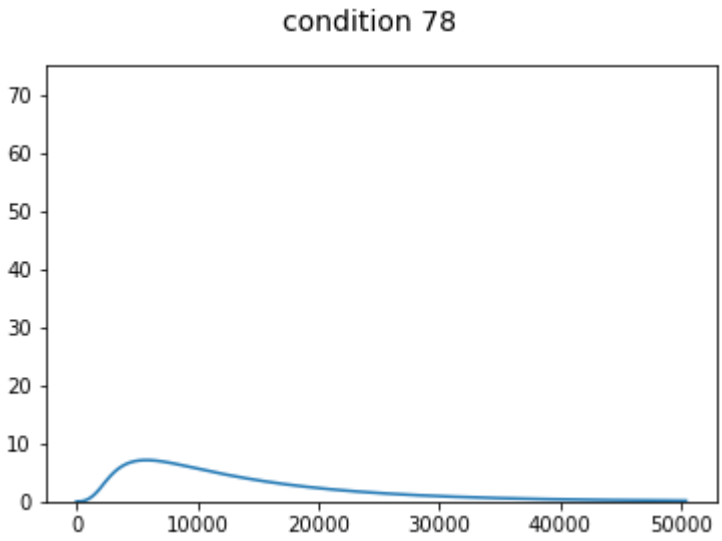

condition 73
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

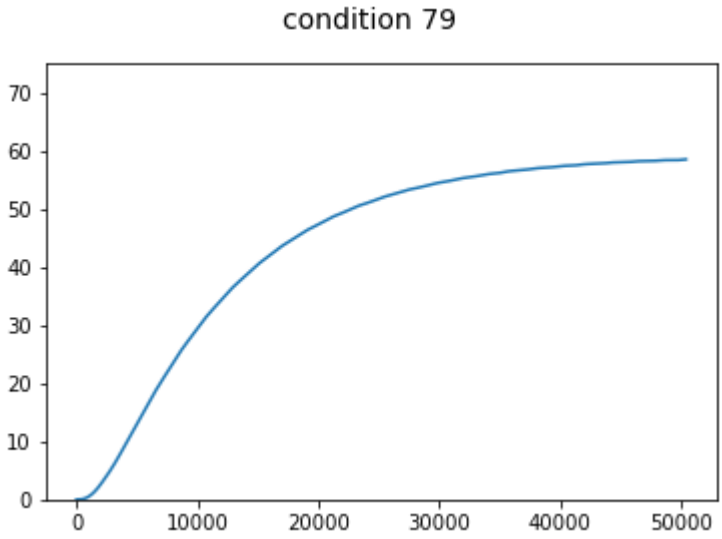

condition 74
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

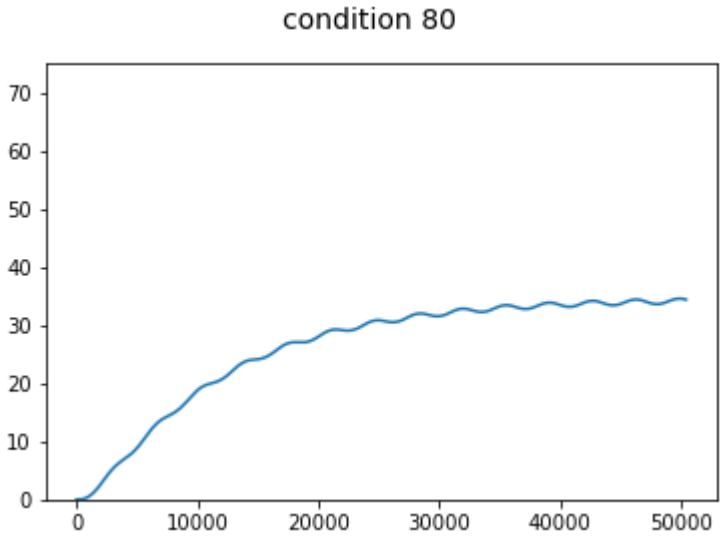

condition 75
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

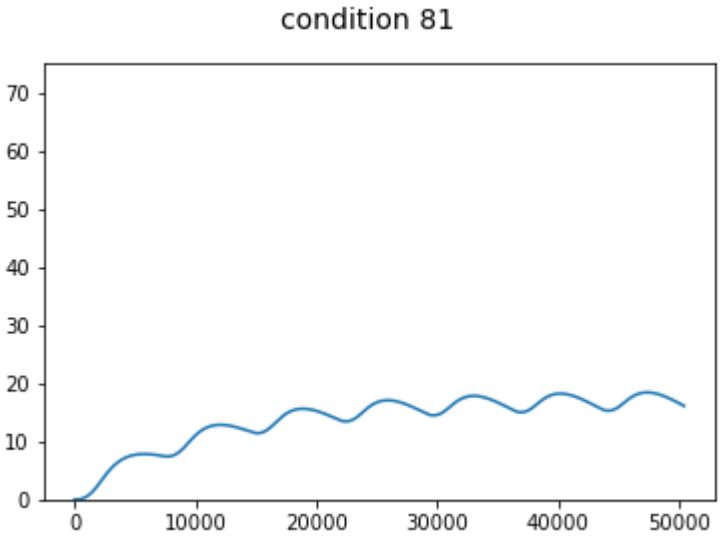

condition 76
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

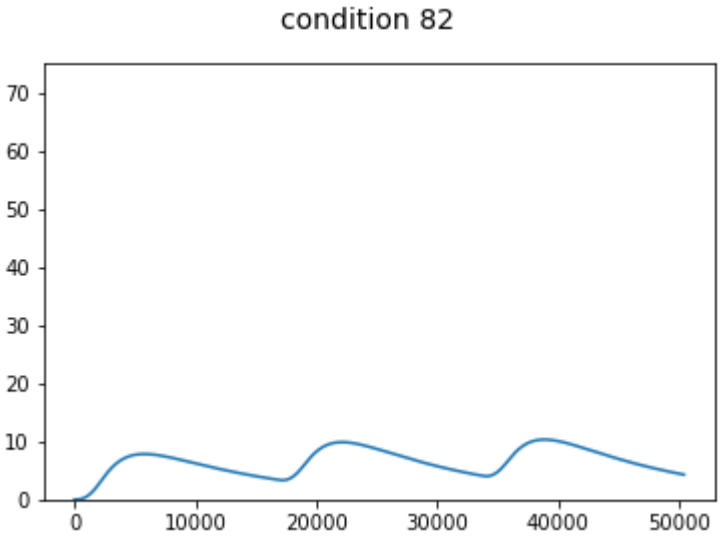

condition 77
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

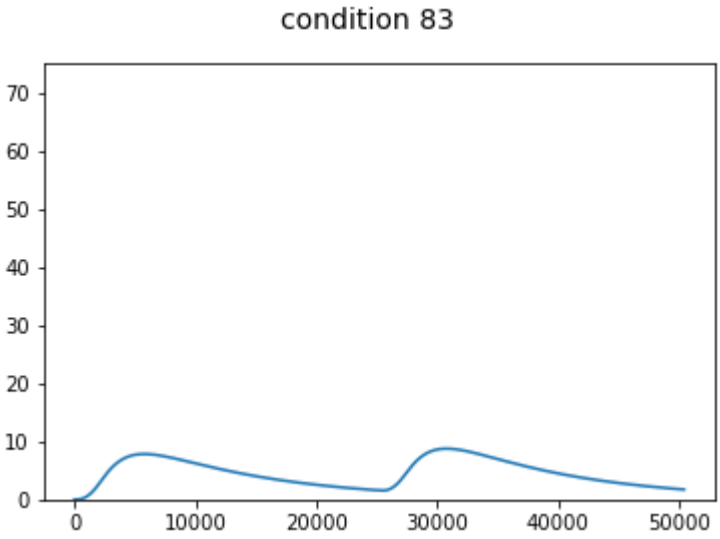

condition 78
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

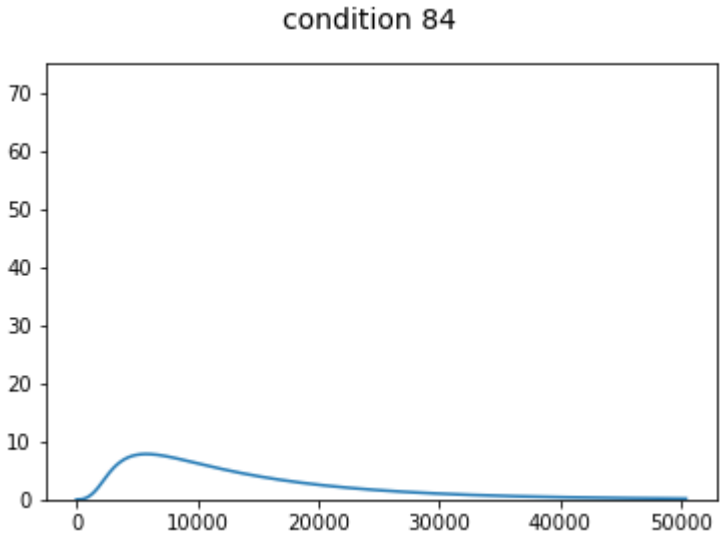

condition 79
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

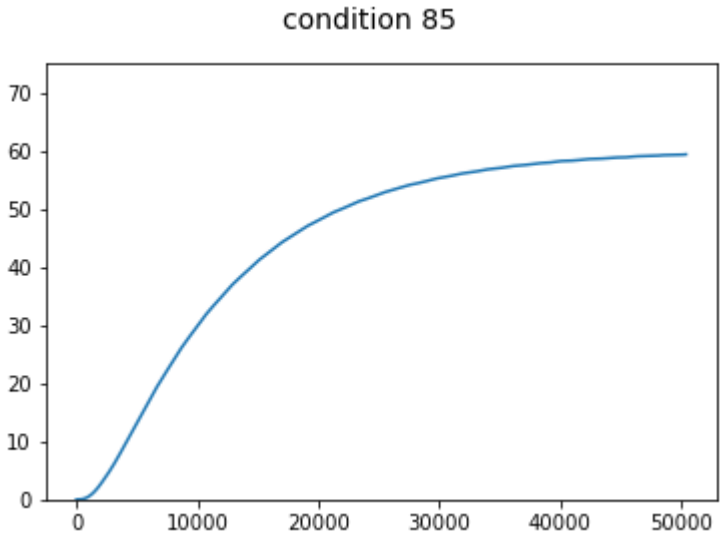

condition 80
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

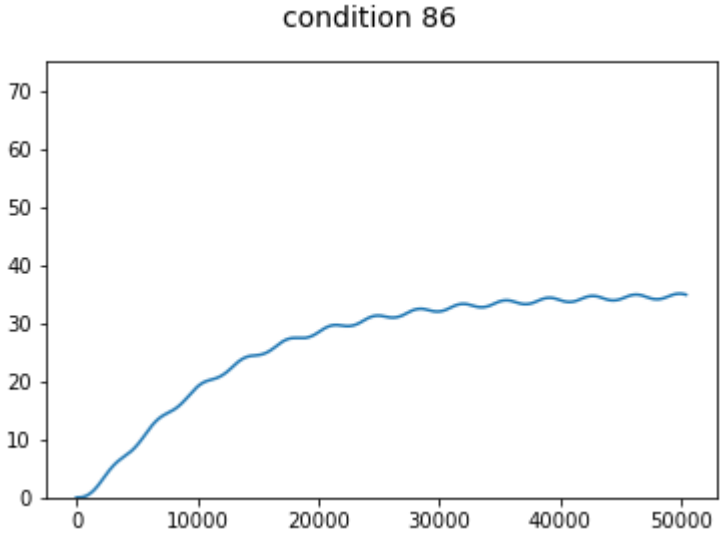

condition 81
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

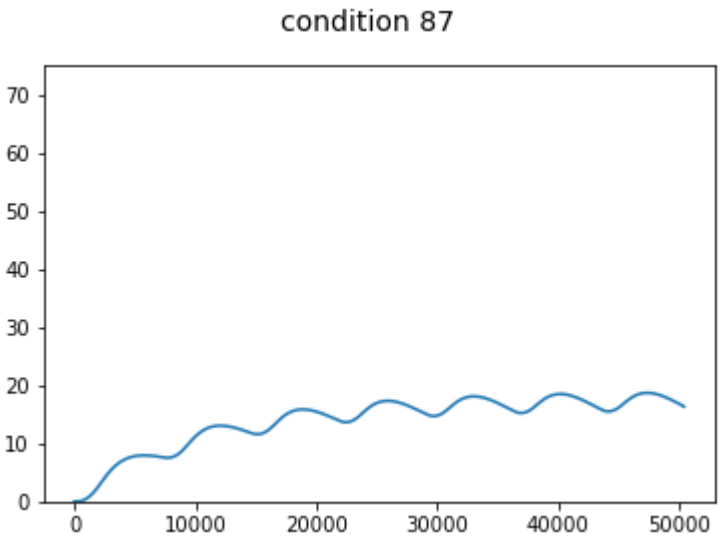

condition 82
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

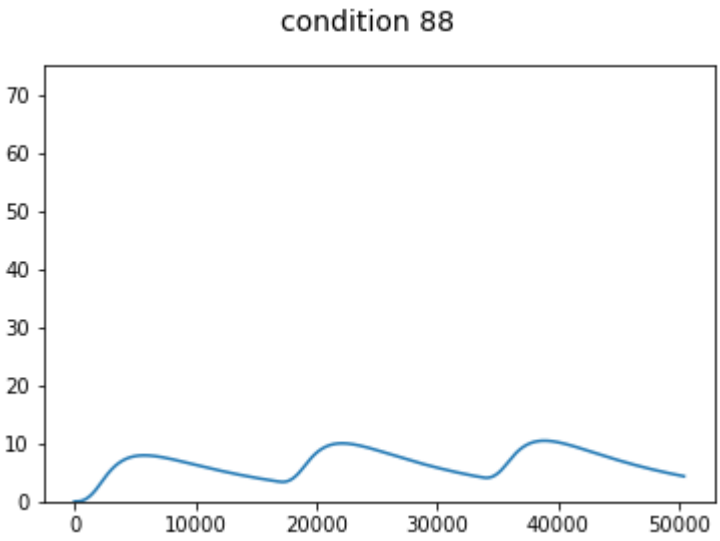

condition 83
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

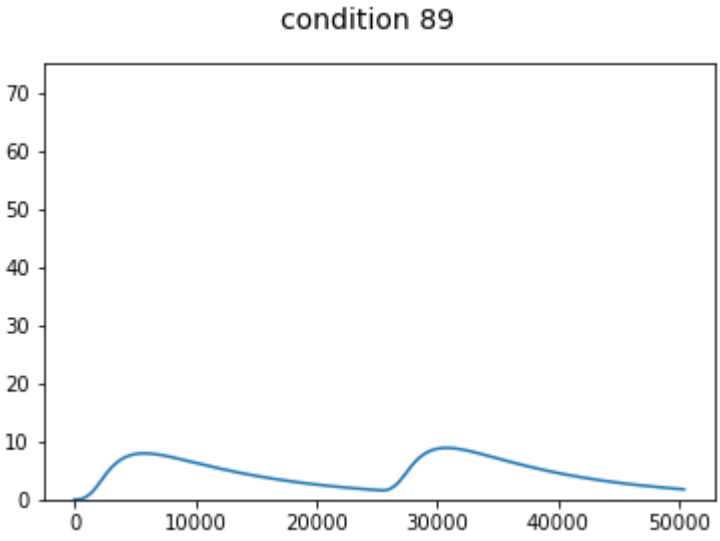

condition 84
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

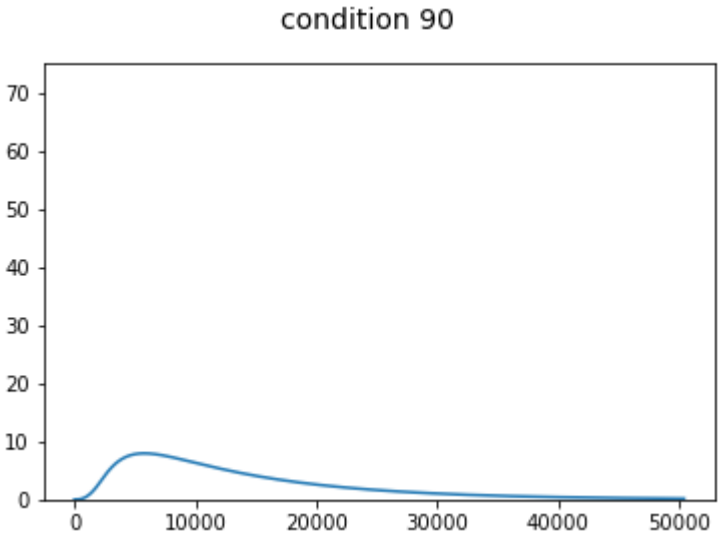

condition 85
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

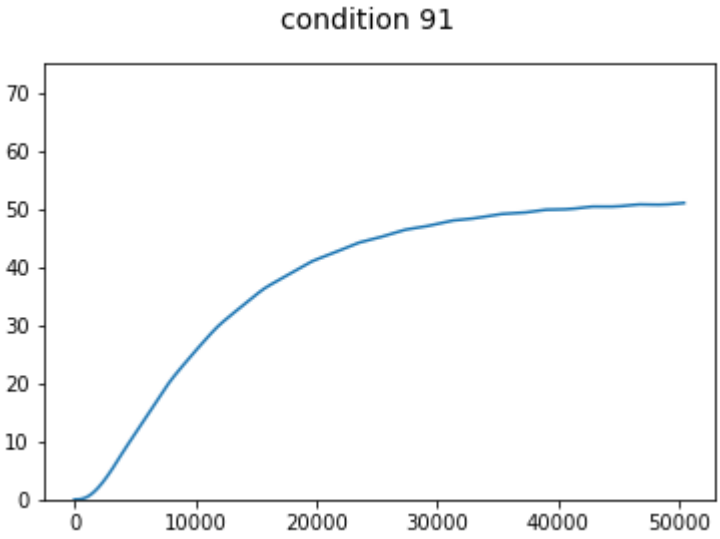

condition 86
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

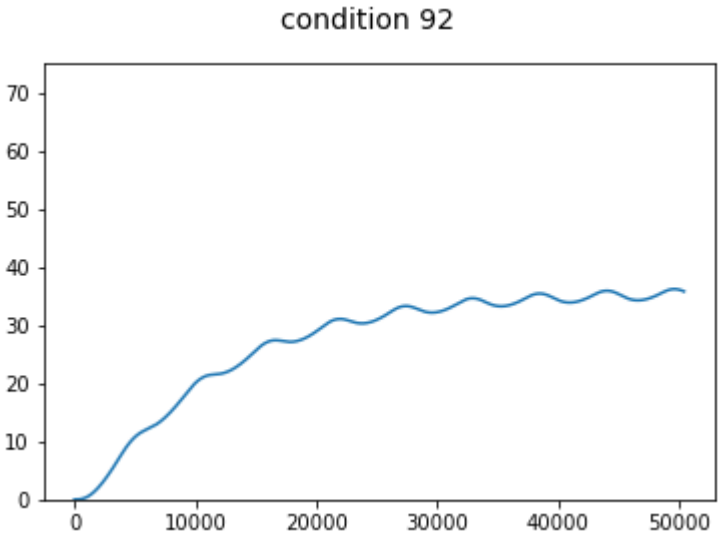

condition 87
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

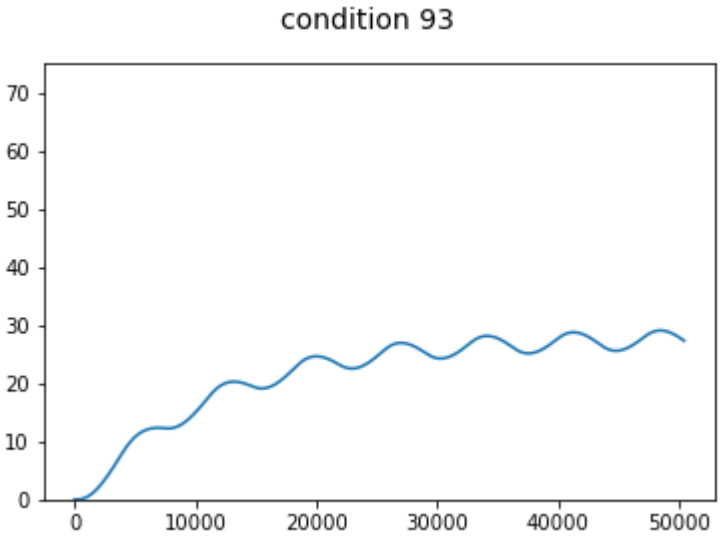

condition 88
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

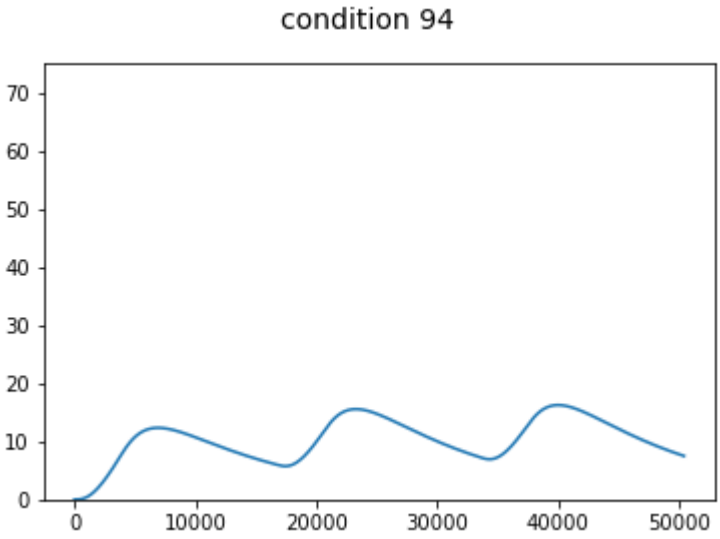

condition 89
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

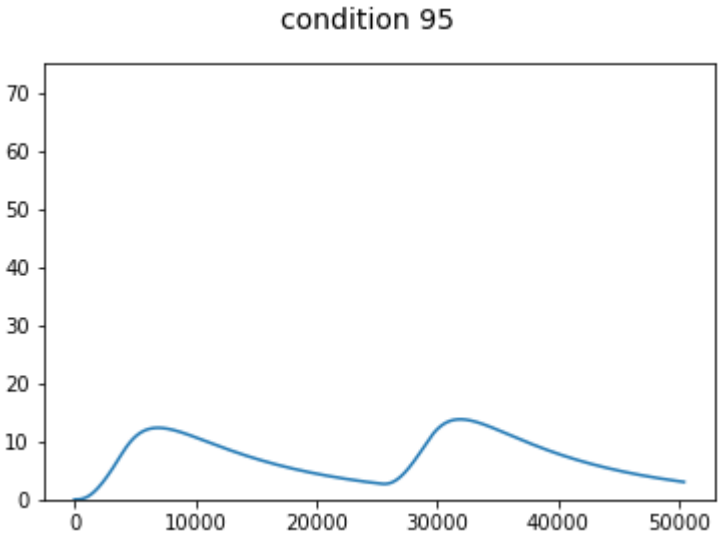

condition 90
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

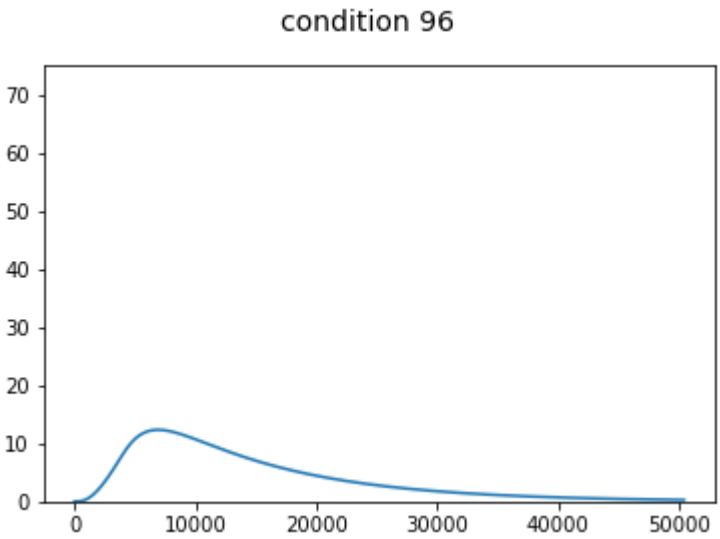

condition 91
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

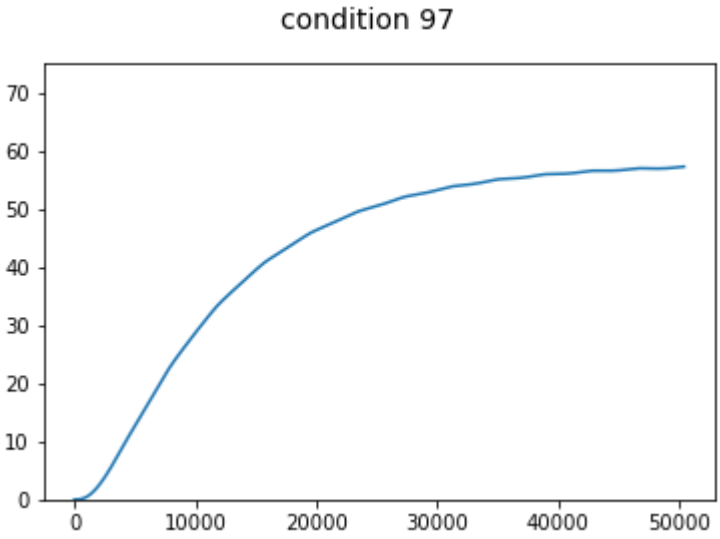

condition 92
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

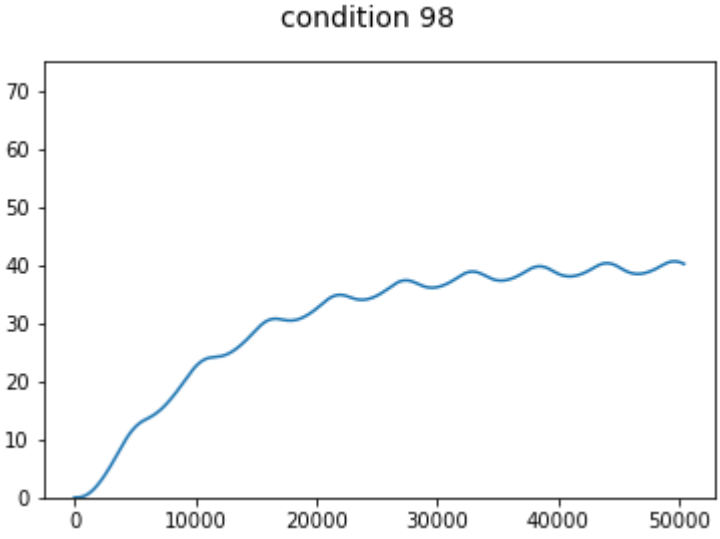

condition 93
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

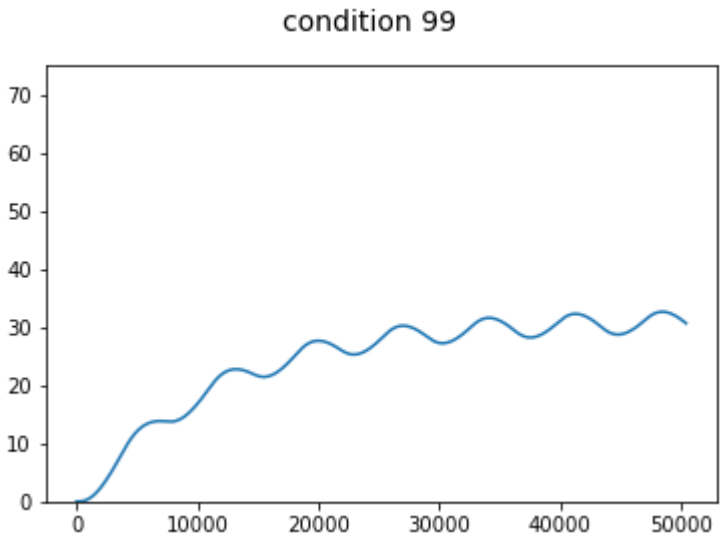

condition 94
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

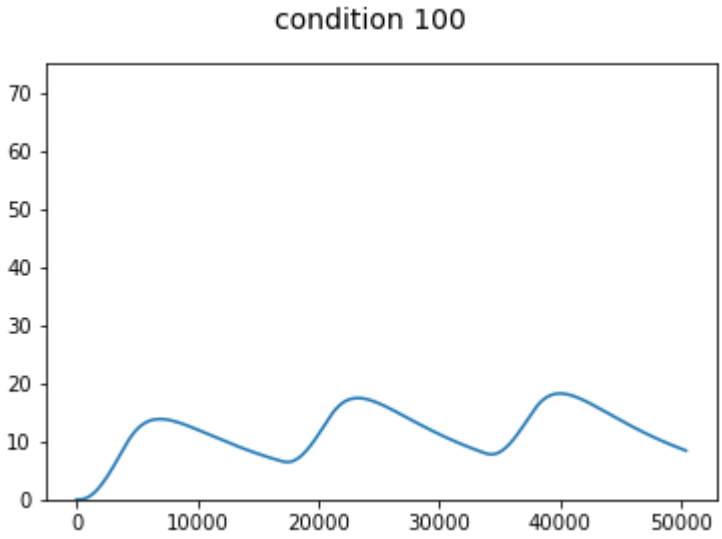

condition 95
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

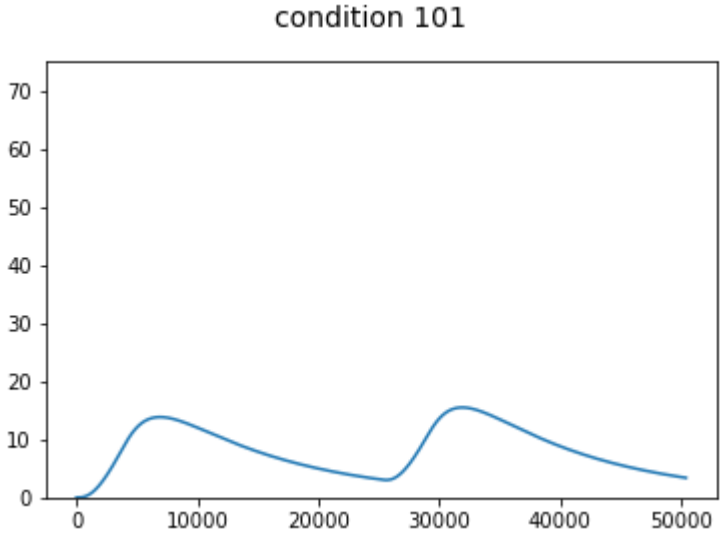

condition 96
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

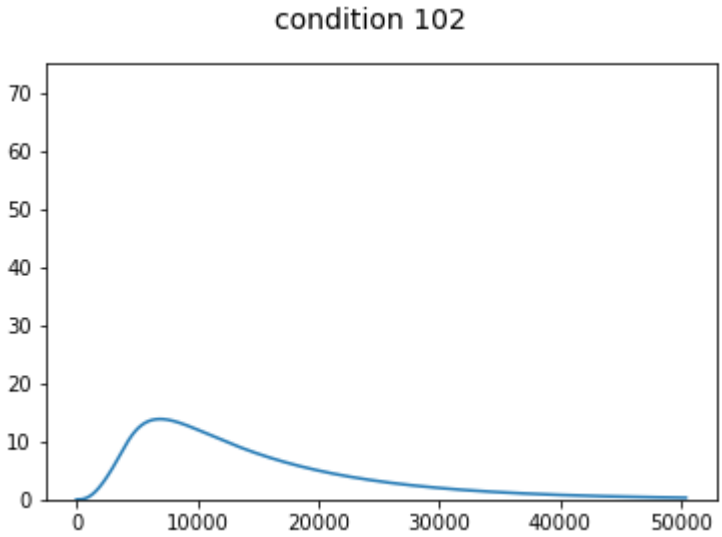

condition 97
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

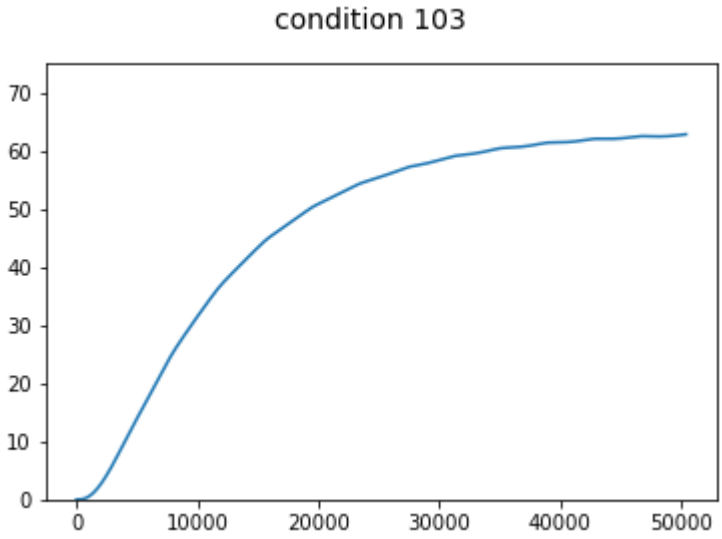

condition 98
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

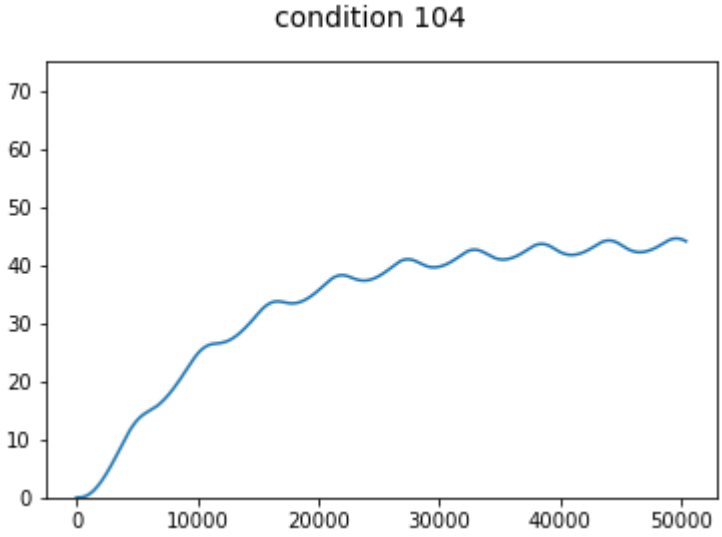

condition 99
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

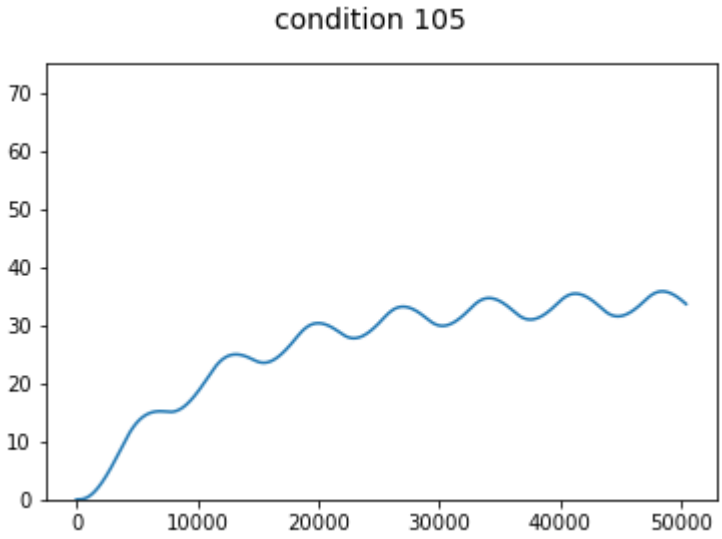

condition 100
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

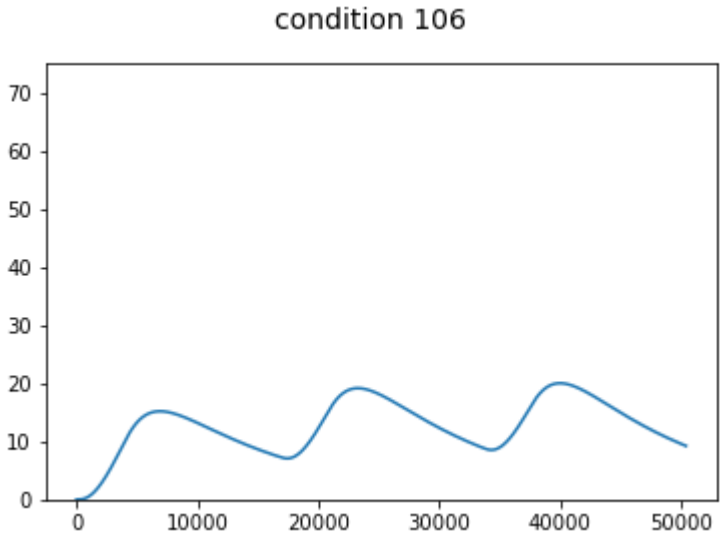

condition 101
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

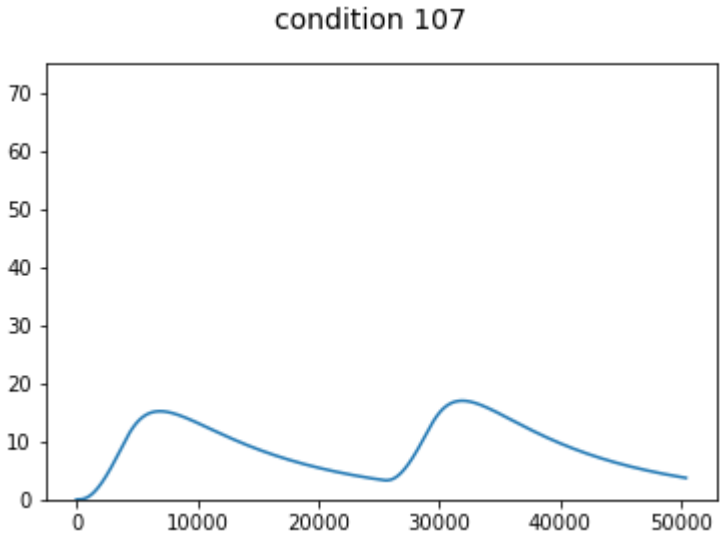

condition 102
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

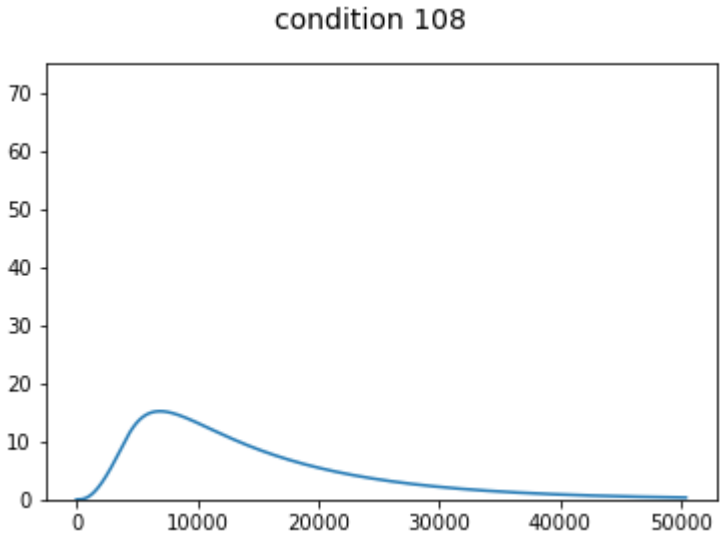

condition 103
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

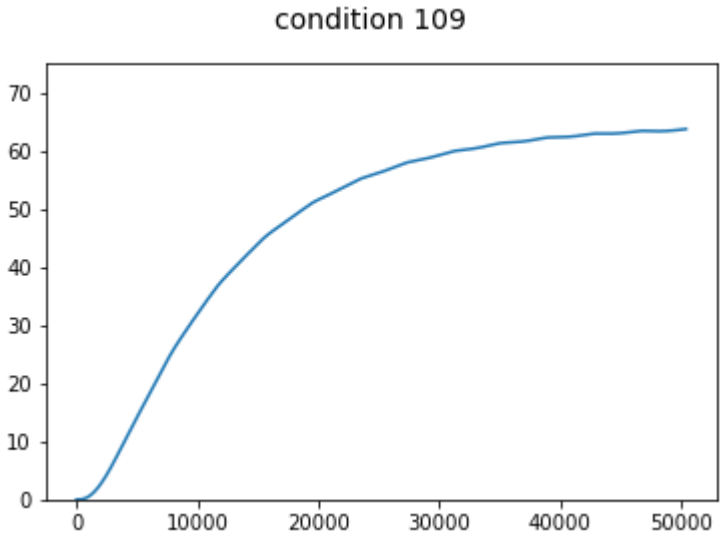

condition 104
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

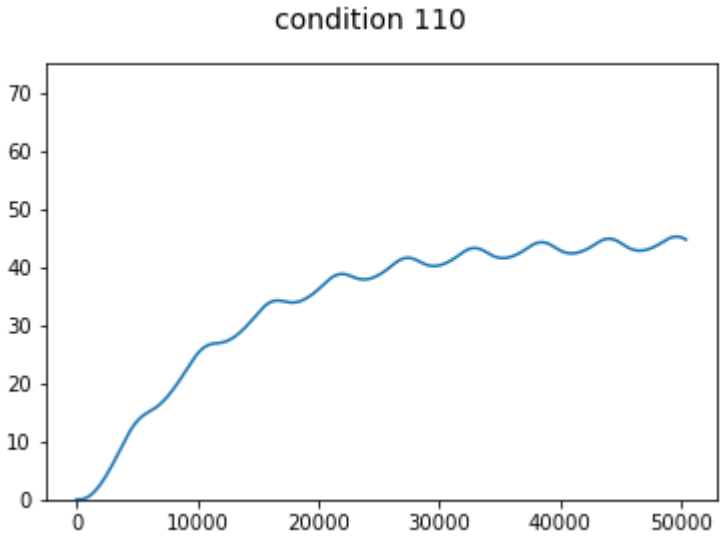

condition 105
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

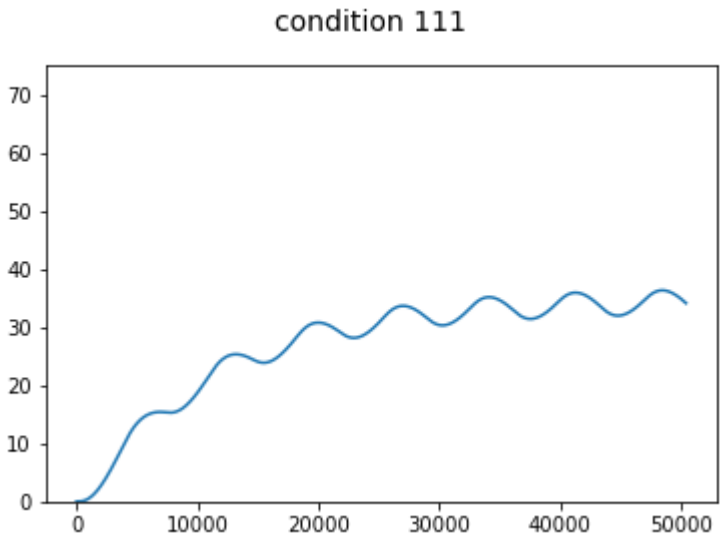

condition 106
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

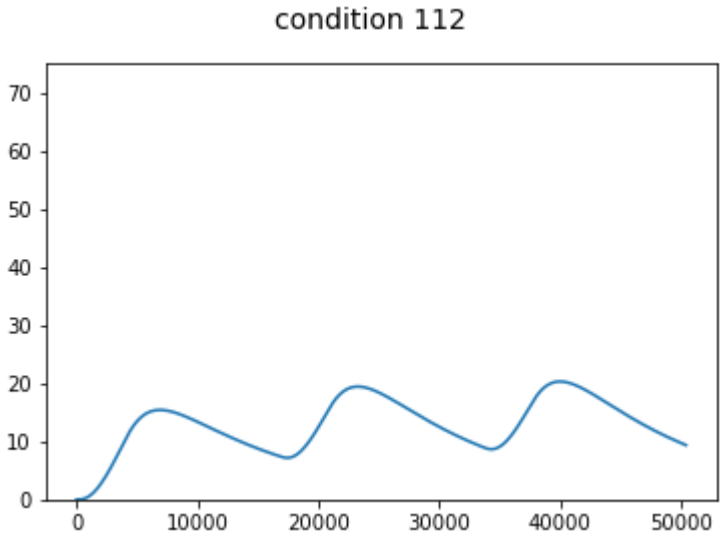

condition 107
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

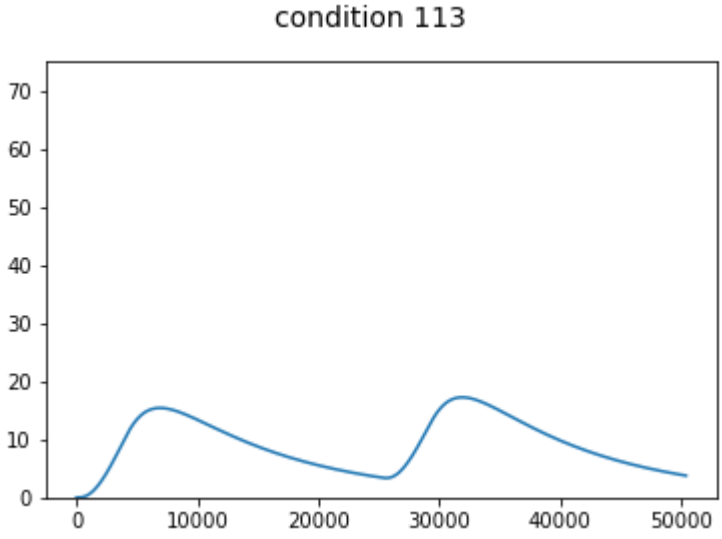

condition 108
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

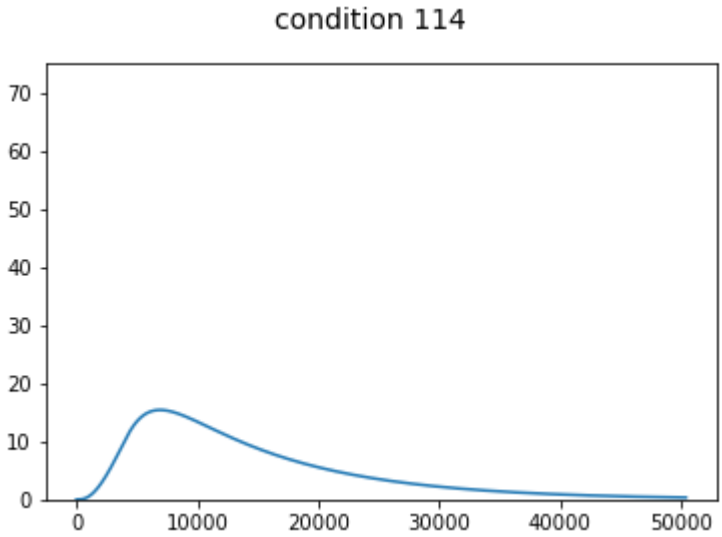

condition 109
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

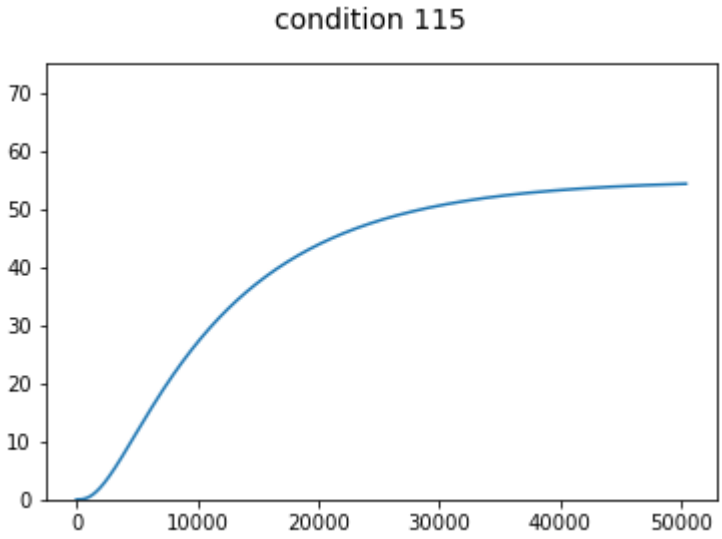

condition 110
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

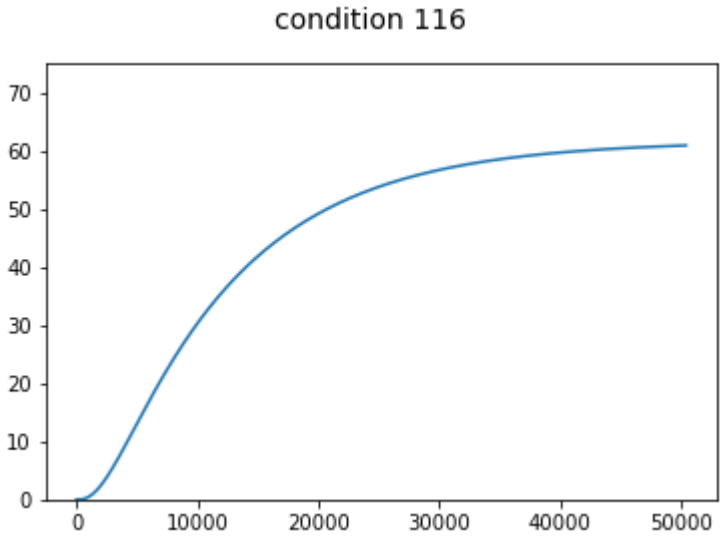

condition 111
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

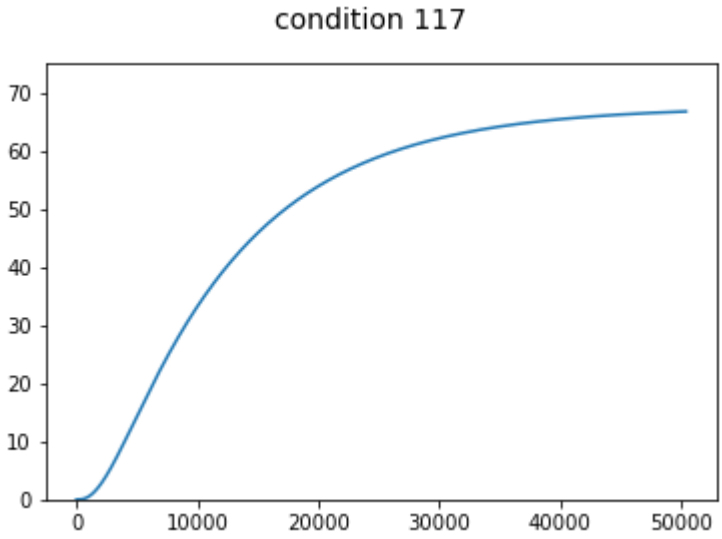

condition 112
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

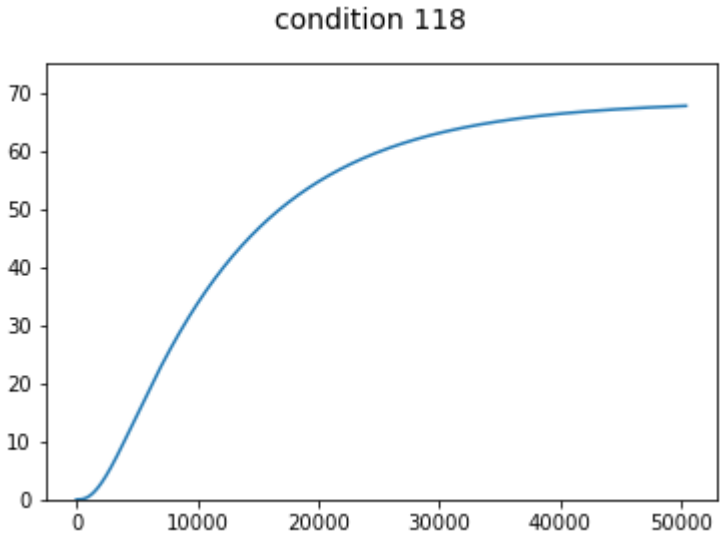

condition 113
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000


condition 114
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000


condition 115
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000


condition 116
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000


condition 117
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000


condition 118
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000

In [ ]:

In [ ]: