project1_pt2_config2

March 10, 2023

[ ]:
```python
import numpy as np
from scipy.integrate import solve_ivp, odeint
from scipy.fftpack import fft
from scipy.signal import find_peaks, hilbert
from tqdm import tqdm
import matplotlib.pyplot as plt
import pywt
# Define the FitzHugh-Nagumo model equations
def fitzhugh_nagumo_coupled(t, xy,alpha , w2, a, b, c, k1, k2):
    x, y , x1, y1= xy
    dxdt = alpha*(y + x - (x**3)/3 + (k1 + c*x1))
    dydt = -(1/alpha) * (w2*x - a + b*y)
    dx1dt = alpha*(y1 + x1 - (x1**3)/3 + (k2 + c*x))
    dy1dt = -(1/alpha) * (w2*x1 - a + b*y1)
    return [dxdt, dydt, dx1dt, dy1dt]

# Define events
# def R_idx(s1,s2):
#     h1 = hilbert(s1)
#     h2 = hilbert(s2)
#     p1 = np.angle(h1)
#     p2 = np.angle(h2)
#     p_diff = p1-p2
#     R = np.abs(np.mean(np.exp(1j*p_diff)))
#     return R

def R_idx(s1, s2):
    #use the Morlet wavelet
    wavelet = "cmor1.5-1.0"
    scales = np.arange(1,40) #range of scales
    coi = [2,39] # cone of influence

    #compute the cwt
    cwt_s1, freq_s1 = pywt.cwt(s1, scales, wavelet, sampling_period=40)
    cwt_s2, freq_s2 = pywt.cwt(s2, scales, wavelet, sampling_period=40)

    #find the phases
```

```
    p1 = np.angle(cwt_s1)
    p2 = np.angle(cwt_s2)
    p_diff= p1-p2

    #find the dominating freq of frequency s1
    p_diff = p_diff[np.argmax(np.sum(np.abs(cwt_s1),axis=1)),:]

    R = np.abs(np.mean(np.exp(1j*p_diff)))
    return R

# Define the initial conditions and parameter values
alpha = 3
a = 0.7
b = 0.8
w2 = 1
x0 = 0
y0 = 0
x10 = 0
y10 = 0
xy0=[x0, x10, y0, y10]
c_vals = np.linspace(-0.5,0,num=30)
k1_vals = np.linspace(-0.85,-0.3, num=30)
k2_vals = np.linspace(-0.85,-1.4, num=30)
t_span = [0, 200]
t_eval = np.linspace(0,200,10000)
```

[ ]:
```
Rmap = np.zeros((c_vals.shape[0], k1_vals.shape[0]))
for i, c in tqdm(enumerate(c_vals)):
    for j, k2 in enumerate(k2_vals):
        k1 = k1_vals[len(k1_vals)-1-j]
        sol = solve_ivp(fitzhugh_nagumo_coupled, t_span, xy0,␣
↪args=(alpha,w2,a,b,c,k1,k2),t_eval=t_eval,dense_output=True,rtol=1e-8)

        s1 = sol.y[0]
        s2 = sol.y[2]

        #Apply hamming filter
        window = np.hamming(s1.shape[0])
        s1 = s1 * window
        s2 = s2 * window

        #Calculate R index
        R = R_idx(s1,s2)

        #Save to response map
        Rmap[i, j] = R
```

```
30it [04:41,  9.39s/it]
```

[ ]:
```
Rmap
```

[ ]:
```
array([[0.55803654, 0.60344618, 0.64770309, 0.69520032, 0.73376998,
        0.7708746 , 0.80707306, 0.83692955, 0.87390651, 0.90100057,
        0.92831629, 0.94937646, 0.96653385, 0.97838493, 0.98876709,
        0.98545689, 0.98064539, 0.97733276, 0.97364563, 0.96833622,
        0.94960052, 0.93082249, 0.91429835, 0.89092635, 0.86432882,
        0.82769601, 0.79837146, 0.76360075, 0.7265619 , 0.68787079],
       [0.50605488, 0.55575445, 0.60065903, 0.65721331, 0.70130831,
        0.74472609, 0.78894731, 0.82574979, 0.86032989, 0.89191348,
        0.92023208, 0.94382658, 0.9606217 , 0.97252625, 0.9779365 ,
        0.97697291, 0.97825145, 0.9805252 , 0.97473046, 0.95542613,
        0.94355255, 0.92683874, 0.90323032, 0.87566429, 0.84498062,
        0.81132213, 0.77589558, 0.72561629, 0.69119951, 0.64704565],
       [0.44603565, 0.5078304 , 0.56082211, 0.61355768, 0.66447395,
        0.71279162, 0.76253206, 0.80469129, 0.84218868, 0.88265195,
        0.91293842, 0.93595693, 0.95310718, 0.96422628, 0.9689539 ,
        0.97724693, 0.98496782, 0.97800531, 0.96553339, 0.95651383,
        0.94015876, 0.9176443 , 0.89112848, 0.86113634, 0.82688066,
        0.78248474, 0.74500313, 0.70145022, 0.65057514, 0.59933605],
       [0.36716934, 0.43301965, 0.49850701, 0.56952652, 0.62560264,
        0.67895374, 0.73149302, 0.78007452, 0.82597111, 0.86488217,
        0.8976915 , 0.92312457, 0.94116556, 0.95373253, 0.97028041,
        0.98361784, 0.97664672, 0.97415072, 0.96827295, 0.95328034,
        0.93190684, 0.90590609, 0.87560301, 0.84099188, 0.79932496,
        0.75143086, 0.70830826, 0.65640027, 0.6049191 , 0.54845858],
       [0.29039737, 0.35933219, 0.43341614, 0.49897592, 0.56199717,
        0.63278595, 0.69799075, 0.75354794, 0.80220778, 0.8435477 ,
        0.87759936, 0.90482384, 0.92779423, 0.954272  , 0.97224343,
        0.97259526, 0.9792645 , 0.97579001, 0.96431765, 0.94555713,
        0.92279979, 0.89235003, 0.85673964, 0.81620721, 0.77095451,
        0.72100758, 0.6660989 , 0.60783472, 0.54697522, 0.47962475],
       [0.2276328 , 0.29463403, 0.36445537, 0.43459827, 0.50884225,
        0.57822386, 0.64560336, 0.70511919, 0.75337017, 0.80325903,
        0.85017326, 0.88681851, 0.92562235, 0.94785343, 0.96196822,
        0.97636045, 0.97833614, 0.97267858, 0.9594779 , 0.93879049,
        0.91064628, 0.87532353, 0.83354991, 0.78460241, 0.73146919,
        0.67545985, 0.61122541, 0.54582907, 0.48584913, 0.41659851],
       [0.15169416, 0.22493098, 0.30910146, 0.3803632 , 0.45489652,
        0.52698766, 0.59604044, 0.6608891 , 0.71962332, 0.76547086,
        0.81496316, 0.87008644, 0.89899751, 0.93413648, 0.95921185,
        0.97237549, 0.97585694, 0.96923227, 0.95224896, 0.9253551 ,
        0.88859018, 0.85023383, 0.80071122, 0.74731698, 0.6905079 ,
        0.63038514, 0.56156729, 0.48782423, 0.42314708, 0.35419205],
       [0.08905417, 0.16038389, 0.2351419 , 0.3247669 , 0.41134883,
```

```
 0.48286622, 0.55209417, 0.61603494, 0.67264297, 0.73588933,
 0.79972027, 0.8414643 , 0.89027792, 0.92533221, 0.95164913,
 0.96890512, 0.9739357 , 0.96560486, 0.94536196, 0.91560491,
 0.87676172, 0.8312286 , 0.77902233, 0.71950274, 0.65398279,
 0.58699176, 0.51758627, 0.44646261, 0.37311536, 0.30050856],
[0.09360503, 0.09833451, 0.16436928, 0.25870467, 0.33886309,
 0.42375587, 0.50587804, 0.57348527, 0.64809879, 0.71916876,
 0.77297456, 0.83192937, 0.87999509, 0.91986558, 0.94926017,
 0.96693174, 0.97193643, 0.96400003, 0.94185858, 0.91134863,
 0.86907168, 0.81839739, 0.76014375, 0.69585141, 0.62673884,
 0.55387231, 0.47815672, 0.4002573 , 0.31157163, 0.22105232],
[0.17115344, 0.09904544, 0.10862474, 0.17970356, 0.2656137 ,
 0.3555021 , 0.43213108, 0.52536701, 0.60390171, 0.67410394,
 0.75091605, 0.81295134, 0.86743951, 0.91202766, 0.94447245,
 0.96390805, 0.96938988, 0.96068501, 0.93874093, 0.90434769,
 0.85902868, 0.80388317, 0.74021176, 0.67066352, 0.58304629,
 0.50517669, 0.4166939 , 0.32742786, 0.24028441, 0.14818444],
[0.28052315, 0.16360853, 0.09479263, 0.1131596 , 0.18760407,
 0.27114761, 0.37538081, 0.46080078, 0.55374155, 0.63427   ,
 0.71361654, 0.7884776 , 0.85228537, 0.90112698, 0.93839227,
 0.95977437, 0.96608524, 0.95723323, 0.93372824, 0.89464632,
 0.84742249, 0.7826511 , 0.71219873, 0.63191133, 0.54552754,
 0.45485567, 0.35881383, 0.25832889, 0.16038541, 0.07420118],
[0.38946143, 0.27539834, 0.16415669, 0.07673299, 0.11754486,
 0.21061986, 0.29281145, 0.39184154, 0.49216286, 0.58850902,
 0.67783353, 0.76215363, 0.83150408, 0.88835389, 0.92956575,
 0.95491044, 0.96278619, 0.95345139, 0.92755818, 0.88594795,
 0.82988926, 0.76068731, 0.67891202, 0.59167914, 0.48752361,
 0.38894665, 0.28551613, 0.18257642, 0.0922114 , 0.08704041],
[0.50400281, 0.40666144, 0.28622859, 0.17812716, 0.08524233,
 0.12397443, 0.22142599, 0.32301745, 0.43153021, 0.53833405,
 0.63886406, 0.72969132, 0.80838346, 0.87279298, 0.92029268,
 0.9494854 , 0.95953952, 0.95062649, 0.92194019, 0.87351869,
 0.81021476, 0.72970308, 0.64149893, 0.54205669, 0.43601482,
 0.32601983, 0.21535525, 0.11272139, 0.06720113, 0.18096212],
[0.59296062, 0.51960108, 0.43362499, 0.29891625, 0.18444619,
 0.09804454, 0.13183141, 0.2356649 , 0.36431497, 0.48289329,
 0.59271898, 0.69347057, 0.78275553, 0.85570504, 0.90965851,
 0.94437653, 0.95762485, 0.94735013, 0.91401881, 0.86026486,
 0.78786822, 0.69960206, 0.59840599, 0.48956102, 0.37416071,
 0.24962025, 0.11839882, 0.06329747, 0.16567983, 0.2872266 ],
[0.65989267, 0.61890109, 0.52406109, 0.41830776, 0.29774468,
 0.17475701, 0.09505518, 0.16242975, 0.28365026, 0.4102506 ,
 0.53461675, 0.64946033, 0.75241271, 0.83609305, 0.89986912,
 0.94012537, 0.95440099, 0.94210826, 0.90419048, 0.84290223,
 0.76141972, 0.66146629, 0.54496288, 0.42040491, 0.28659171,
 0.15075173, 0.04917492, 0.15002775, 0.28132376, 0.40856197],
```

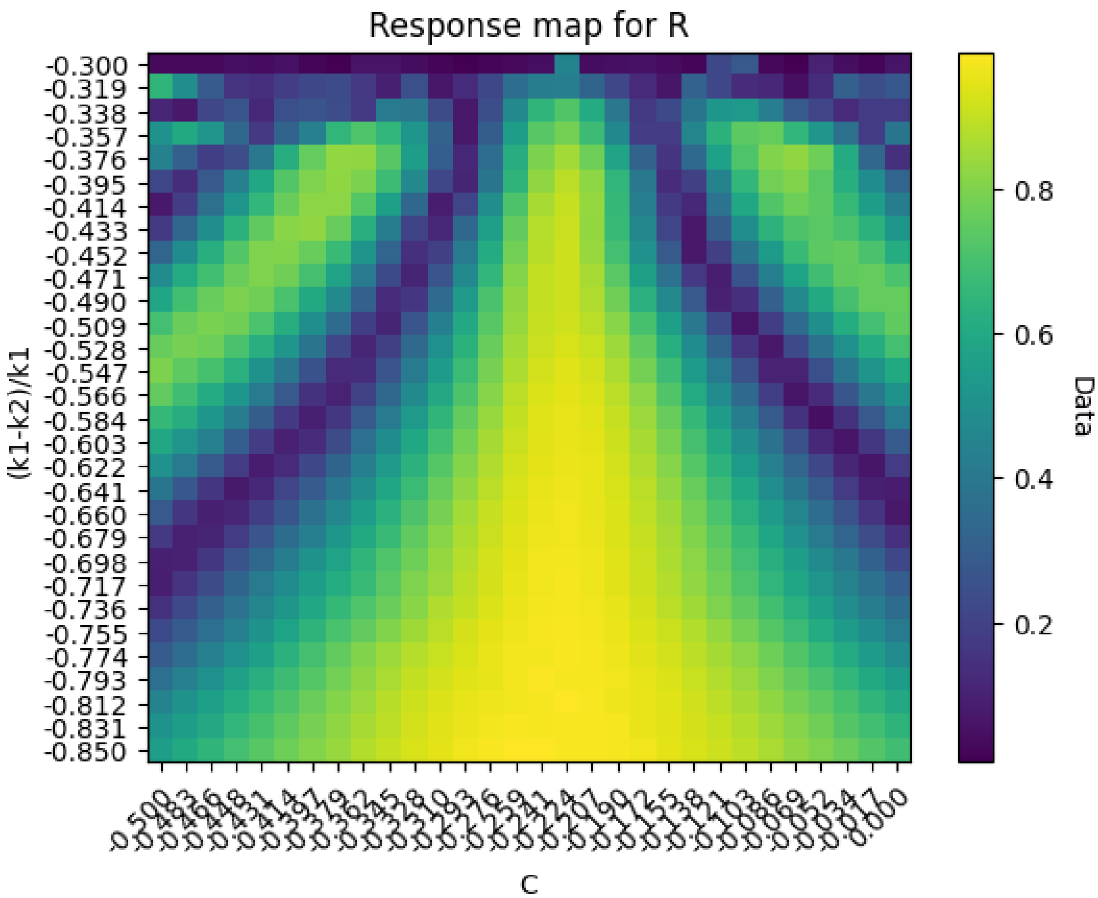
```
[0.75310759, 0.6878802 , 0.61060746, 0.5236594 , 0.4095055 ,
 0.2793011 , 0.15219178, 0.10339767, 0.2003353 , 0.33933787,
 0.47492546, 0.5992143 , 0.71451994, 0.81198535, 0.88595329,
 0.93254304, 0.94875739, 0.93376974, 0.88853402, 0.81700662,
 0.72307873, 0.60828352, 0.48161397, 0.34414086, 0.18743287,
 0.06296875, 0.1281103 , 0.26947833, 0.40020491, 0.51167581],
[0.79321593, 0.74491118, 0.70153272, 0.62884616, 0.50948865,
 0.39289728, 0.25479658, 0.12831298, 0.13604141, 0.26657078,
 0.41089856, 0.54942263, 0.67525236, 0.78582633, 0.8687737 ,
 0.92165181, 0.94008964, 0.92203842, 0.86861964, 0.78490669,
 0.68036266, 0.55054929, 0.4071496 , 0.25120253, 0.10142473,
 0.09961208, 0.2482488 , 0.39315982, 0.5136456 , 0.60518966],
[0.75885831, 0.78655834, 0.77251427, 0.70805172, 0.62740943,
 0.51947024, 0.38613122, 0.21501389, 0.11201251, 0.19379317,
 0.34540064, 0.49853337, 0.63842042, 0.75913108, 0.85176357,
 0.91025317, 0.92893136, 0.90635316, 0.84488566, 0.74944911,
 0.62666596, 0.48280482, 0.32295282, 0.15642903, 0.07480566,
 0.22409504, 0.37258982, 0.49862072, 0.61578427, 0.69729206],
[0.69417217, 0.76306336, 0.79308236, 0.77457935, 0.71244385,
 0.61982902, 0.4953426 , 0.34921834, 0.19073715, 0.10755322,
 0.26495322, 0.43037148, 0.59984584, 0.73529645, 0.83812364,
 0.90088731, 0.91801838, 0.88798106, 0.81387638, 0.70194153,
 0.56163762, 0.39941779, 0.22308249, 0.06313364, 0.19098313,
 0.34941092, 0.49769566, 0.61438915, 0.69717799, 0.74931047],
[0.58805682, 0.69156447, 0.76076182, 0.79391228, 0.77569172,
 0.70857514, 0.60495915, 0.46846667, 0.3081571 , 0.13190666,
 0.16243737, 0.34893828, 0.53634934, 0.70511717, 0.82599439,
 0.89816969, 0.9129659 , 0.87073228, 0.77604699, 0.63179248,
 0.47065005, 0.28104253, 0.09806563, 0.14133484, 0.30856395,
 0.46851038, 0.60509656, 0.69742573, 0.75583444, 0.7556078 ],
[0.48196765, 0.60597341, 0.69301382, 0.76514889, 0.80062412,
 0.77598527, 0.69848504, 0.57383799, 0.41358416, 0.22938585,
 0.10959086, 0.26694696, 0.47070808, 0.65815255, 0.80711244,
 0.89747342, 0.91477801, 0.85869112, 0.73570199, 0.56471499,
 0.37158758, 0.17101788, 0.09112954, 0.26217522, 0.4325775 ,
 0.5776261 , 0.68732379, 0.7488647 , 0.75653643, 0.71100094],
[0.32720031, 0.4847732 , 0.62171532, 0.72444784, 0.79884169,
 0.8014369 , 0.76200209, 0.66807318, 0.51455844, 0.32779961,
 0.14043572, 0.19281048, 0.4075235 , 0.61252439, 0.77795636,
 0.88451136, 0.91335304, 0.83910415, 0.69646651, 0.50275881,
 0.28073862, 0.07514647, 0.19381006, 0.38422988, 0.54260517,
 0.66682162, 0.7403554 , 0.75407415, 0.70878552, 0.61340763],
[0.17322038, 0.33152461, 0.49083526, 0.63614648, 0.749607  ,
 0.81485064, 0.82703926, 0.76090939, 0.62236321, 0.4174062 ,
 0.20969056, 0.12199786, 0.33466527, 0.56307859, 0.75220078,
 0.87942739, 0.91154931, 0.83056937, 0.67017998, 0.45574128,
 0.21072274, 0.07456525, 0.29542411, 0.49175253, 0.64249969,
```

```
        0.71630427, 0.7367499 , 0.70601354, 0.63697205, 0.52706915],
       [0.07298887, 0.18400532, 0.36780792, 0.51778914, 0.65684461,
        0.76125484, 0.81747692, 0.81660643, 0.7314257 , 0.56001937,
        0.33585291, 0.08387007, 0.21279458, 0.49096315, 0.71581494,
        0.86067078, 0.90326162, 0.83369261, 0.66180565, 0.42765686,
        0.17212674, 0.11481541, 0.38014992, 0.59265079, 0.72268795,
        0.77362351, 0.74316045, 0.6631549 , 0.55692905, 0.42006827],
       [0.21961604, 0.13507023, 0.2804867 , 0.43404206, 0.59481197,
        0.72415667, 0.79816833, 0.82634591, 0.79540305, 0.68035955,
        0.47019924, 0.21599329, 0.1074279 , 0.38971142, 0.64702149,
        0.82710689, 0.88727612, 0.82318965, 0.65371465, 0.40125227,
        0.13167218, 0.19258586, 0.44545678, 0.65833034, 0.77881308,
        0.80227339, 0.73924882, 0.62330842, 0.48430165, 0.3257328 ],
       [0.43953095, 0.30849319, 0.1884952 , 0.23801676, 0.4017489 ,
        0.61776409, 0.75624297, 0.82900359, 0.82654712, 0.73565048,
        0.55505392, 0.30139643, 0.11507237, 0.35613325, 0.61746001,
        0.79449885, 0.85114807, 0.77101507, 0.57928149, 0.31812296,
        0.15468689, 0.34261071, 0.56839326, 0.71661859, 0.80093305,
        0.82709803, 0.77154804, 0.6125548 , 0.32929732, 0.14133869],
       [0.51299458, 0.60472668, 0.51801778, 0.33246686, 0.17027304,
        0.31972674, 0.43436531, 0.65370418, 0.71953975, 0.65465337,
        0.51948433, 0.32053849, 0.0758917 , 0.29608619, 0.55237752,
        0.73332695, 0.78395424, 0.68143477, 0.46338962, 0.18375212,
        0.18012843, 0.45234341, 0.63899741, 0.74261357, 0.75541436,
        0.6564875 , 0.52051126, 0.37063532, 0.17975605, 0.38690832],
       [0.11350597, 0.0728471 , 0.21627319, 0.26550437, 0.11749096,
        0.24549635, 0.26490142, 0.23948057, 0.17564524, 0.4116616 ,
        0.39617962, 0.23227855, 0.07423774, 0.24218777, 0.47029823,
        0.64748159, 0.7206109 , 0.61217842, 0.39524712, 0.17522019,
        0.22711545, 0.38893956, 0.52361427, 0.5396537 , 0.42772208,
        0.28597896, 0.21355321, 0.12232527, 0.18041561, 0.17683771],
       [0.65022593, 0.49180405, 0.29073806, 0.15973675, 0.1412758 ,
        0.18477253, 0.22144688, 0.235745  , 0.16827913, 0.09388896,
        0.24734938, 0.0700797 , 0.12978079, 0.21895707, 0.36080239,
        0.4205856 , 0.42728877, 0.32234487, 0.21436253, 0.12854322,
        0.06750267, 0.31710993, 0.21869177, 0.13088621, 0.11516516,
        0.04562173, 0.13723565, 0.30705303, 0.24338068, 0.27354345],
       [0.03174703, 0.03365054, 0.03079089, 0.04588986, 0.04125009,
        0.0538085 , 0.02427504, 0.00701898, 0.05986911, 0.05709828,
        0.03868912, 0.02176965, 0.00836957, 0.02297835, 0.03090911,
        0.04433213, 0.44867476, 0.03773827, 0.04623577, 0.05318908,
        0.04064399, 0.02353479, 0.2182139 , 0.27991494, 0.028883  ,
        0.01059461, 0.10530175, 0.04340131, 0.01987775, 0.06422843]])
```

[ ]:
```python
K = []
C = []
for i, c in tqdm(enumerate(c_vals)):
```

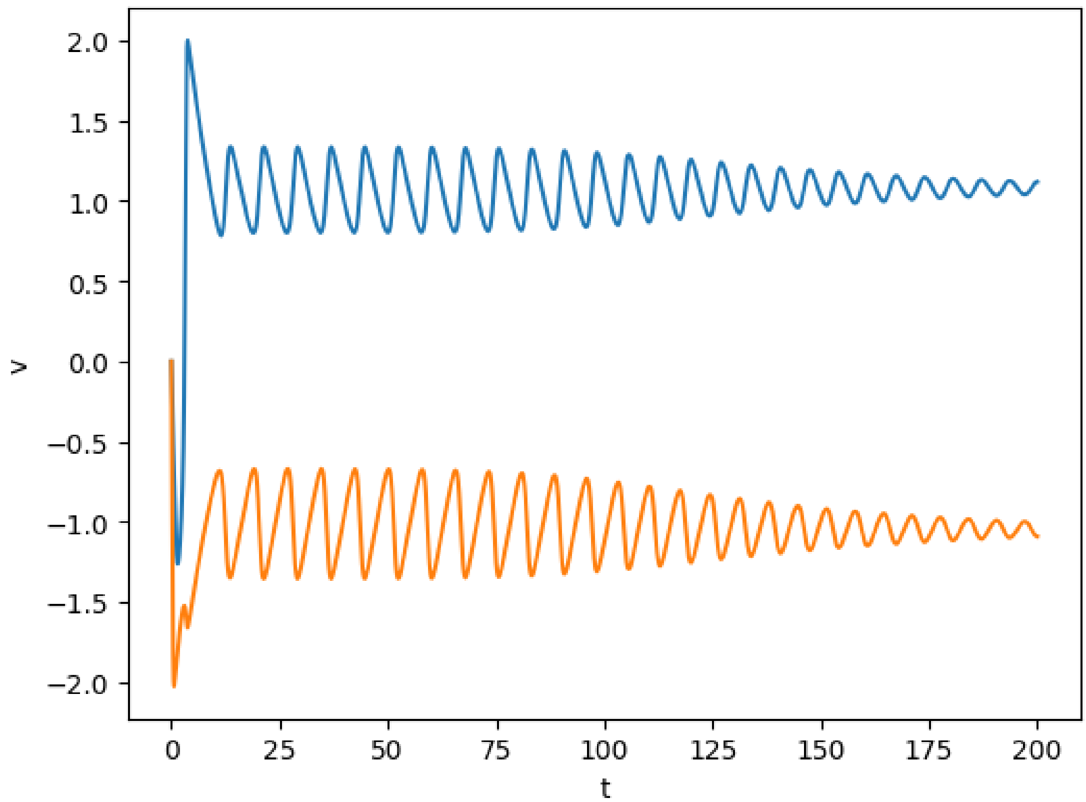
```
    C.append(c)
# for i,_ in enumerate(k2_vals):
#     K.append((k1_vals[i]-k2_vals[i])/(k1_vals[i]))
for k1 in k1_vals:
    K.append(k1)

# plt.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)
fig, ax = plt.subplots()
im = ax.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)

# Set tick labels
ax.set_xticks(np.arange(len(C)))
ax.set_yticks(np.arange(len(K)))
ax.set_xticklabels([f"{c:.3f}" for c in C])
ax.set_yticklabels([f"{k:.3f}" for k in K])

plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")
cbar = ax.figure.colorbar(im, ax=ax)
cbar.ax.set_ylabel('Data', rotation=-90, va="bottom")

# Add axis labels
ax.set_xlabel('C')
ax.set_ylabel('(k1-k2)/k1')
ax.set_title("Response map for R")
# Show plot
plt.show()
```

```
30it [00:00, 151601.35it/s]
```



[ ]:
```python
k1 = -0.40092
k2 = -1.7-k1
c = -0.207
# k1 = -0.3
# k2 = -0.68
# c = -0.3
sol = solve_ivp(fitzhugh_nagumo_coupled, t_span, xy0, args=(alpha,w2,a,b,c,k1,k2), t_eval=t_eval, dense_output=True,rtol=1e-8)
t = sol.t
s1 = sol.y[0]
s2 = sol.y[2]
```

[ ]:
```python
# plt.plot(t[0:100], (np.abs(np.fft.fft(v))**2)[0:100])
plt.plot(t, s1)
plt.plot(t, s2)
plt.xlabel('t')
plt.ylabel('v')
plt.show()
```


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
v
0
25
50
75
100
125
150
175
200
t