project1_pt3_corr

March 10, 2023

[ ]:
```
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
import nolds
import pickle
# 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]

alpha = 3
a = 0.7
b = 0.8
w2 = 1
x0 = 0
y0 = 0
x10 = 0
y10 = 0
xy0=[x0, x10, y0, y10]

n =30
c_vals = np.linspace(-0.3,0,num=n)
k1_vals = np.array([-0.3 for i in range(n)])
k2_vals = np.linspace(-0.65,-0.9, num=n)
t_span = [0, 200]
t_eval = np.linspace(0,100,5000)
```

[ ]:
```
lyap_map_x1 = np.zeros((c_vals.shape[0], k1_vals.shape[0]))
lyap_map_x2 = np.zeros((c_vals.shape[0], k1_vals.shape[0]))
```

```
for i, c in enumerate(c_vals):
    for j, k2 in tqdm(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
        lyap_x1 = nolds.corr_dim(s1, emb_dim=10)
        lyap_x2 = nolds.corr_dim(s2, emb_dim=10)

        #Save to response map
        lyap_map_x1[i, j] = lyap_x1
        lyap_map_x2[i, j] = lyap_x2
```

```
30it [02:42,  5.43s/it]
30it [02:41,  5.38s/it]
30it [02:42,  5.41s/it]
30it [02:42,  5.41s/it]
30it [02:42,  5.42s/it]
30it [02:43,  5.44s/it]
30it [02:43,  5.46s/it]
30it [02:35,  5.17s/it]
30it [02:27,  4.91s/it]
30it [02:27,  4.92s/it]
30it [02:27,  4.91s/it]
30it [02:26,  4.89s/it]
30it [02:26,  4.88s/it]
30it [02:26,  4.89s/it]
30it [02:34,  5.14s/it]
30it [02:41,  5.40s/it]
30it [02:41,  5.37s/it]
30it [02:41,  5.39s/it]
30it [02:35,  5.19s/it]
30it [02:43,  5.45s/it]
30it [02:43,  5.44s/it]
30it [02:39,  5.31s/it]
30it [02:38,  5.28s/it]
30it [02:38,  5.28s/it]
30it [02:38,  5.28s/it]
```

```
30it [02:39,  5.31s/it]
30it [02:39,  5.32s/it]
30it [02:38,  5.28s/it]
30it [02:25,  4.84s/it]
30it [02:25,  4.84s/it]
```

[ ]:
```python
with open("lyapmapx1.pkl", 'wb') as f:
    pickle.dump(lyap_map_x1, f)

with open("lyapmapx2.pkl", 'wb') as f:
    pickle.dump(lyap_map_x2, f)
```

[ ]:
```python
lyap_map_x1
```

[ ]:
```
array([[1.00462039, 1.00302269, 1.0000049 , 0.99558897, 0.99044912,
        0.98506261, 0.97955635, 0.97419131, 0.96895127, 0.96398491,
        0.95936443, 0.95524417, 0.95142515, 0.94801304, 0.94460527,
        0.94114318, 0.93769189, 0.93386786, 0.92975541, 0.92499757,
        0.91960975, 0.91223771, 0.90271474, 0.89477426, 0.88817952,
        0.8826741 , 0.87812446, 0.87441891, 0.87104376, 0.86790875],
       [0.99360149, 1.00132384, 1.00524788, 1.00343711, 1.00241632,
        1.00054921, 0.99708367, 0.99226723, 0.98686391, 0.98129251,
        0.97562415, 0.97013778, 0.964963  , 0.96015616, 0.95577056,
        0.95178161, 0.94830012, 0.94492224, 0.94135521, 0.93786098,
        0.93409636, 0.92992967, 0.92502364, 0.91962914, 0.91168399,
        0.9021742 , 0.89430174, 0.88760584, 0.8818805 , 0.87710853],
       [0.97060357, 0.9751253 , 0.98005041, 0.98597911, 0.99346384,
        1.00122498, 1.00336946, 1.0013834 , 1.00030073, 0.99778983,
        0.99350664, 0.98819218, 0.982545  , 0.9767655 , 0.97105584,
        0.96568654, 0.96069698, 0.95618201, 0.95209405, 0.94855678,
        0.94511737, 0.94162666, 0.93800723, 0.93426326, 0.92994497,
        0.92491924, 0.91963506, 0.91061528, 0.90119332, 0.89340855],
       [0.95636398, 0.95944972, 0.96303683, 0.96698983, 0.97125564,
        0.97594639, 0.98127195, 0.98776047, 0.99584638, 1.00266151,
        1.00128932, 0.99972862, 0.99797489, 0.99440363, 0.98922385,
        0.98345418, 0.97754382, 0.97172981, 0.9662145 , 0.96105092,
        0.95644125, 0.95236179, 0.94884606, 0.94536486, 0.94178606,
        0.9382457 , 0.93436673, 0.9299677 , 0.92467679, 0.91899199],
       [0.95022478, 0.9514719 , 0.95316429, 0.95536488, 0.9582255 ,
        0.9614908 , 0.96517078, 0.96930388, 0.97375999, 0.97872447,
        0.98478749, 0.99241842, 1.00021627, 1.00189455, 0.99940685,
        0.99798806, 0.99489851, 0.98990042, 0.98407595, 0.97804025,
        0.9721332 , 0.96650019, 0.96130514, 0.95668126, 0.95272647,
        0.94919863, 0.94570261, 0.9421471 , 0.93863367, 0.93459703],
       [0.94966395, 0.95038569, 0.95118408, 0.95217078, 0.95341554,
        0.95496379, 0.95693499, 0.95936937, 0.96230288, 0.96563001,
        0.96950952, 0.97382104, 0.97875802, 0.98471915, 0.99213194,
```

```
 0.99962913, 1.00305334, 0.99982448, 0.99829178, 0.99550981,
 0.99054683, 0.98451149, 0.97837677, 0.97233741, 0.96665475,
 0.96160126, 0.95713874, 0.95327208, 0.94980348, 0.94640679],
[0.95079332, 0.95219488, 0.95380677, 0.95548754, 0.95711245,
 0.95838672, 0.95957343, 0.96055976, 0.96184171, 0.96327909,
 0.96517015, 0.9674957 , 0.97027563, 0.97368333, 0.97786729,
 0.98294112, 0.98928162, 0.99607281, 1.00272613, 1.00495855,
 1.00105562, 0.99923989, 0.99628766, 0.99104931, 0.98488746,
 0.97864203, 0.97259368, 0.96713079, 0.96224893, 0.95802046],
[0.95232227, 0.95226104, 0.95296471, 0.95420135, 0.95614813,
 0.95852353, 0.96142599, 0.96454767, 0.96798648, 0.97127039,
 0.97434223, 0.97709353, 0.97952033, 0.98180604, 0.98399678,
 0.9865617 , 0.98967354, 0.99340405, 0.99768908, 1.00195593,
 1.00634986, 1.01044358, 1.00737504, 1.00331429, 1.00118863,
 0.99765841, 0.99197436, 0.98556529, 0.97931812, 0.97356977],
[0.96558231, 0.96191583, 0.95874696, 0.95669188, 0.95573991,
 0.95574243, 0.95666293, 0.95837928, 0.96091882, 0.96423158,
 0.96847513, 0.97357951, 0.97969995, 0.98681393, 0.99367883,
 0.99990621, 1.00531736, 1.00961567, 1.01320425, 1.01608112,
 1.0182937 , 1.02013823, 1.02169111, 1.01952557, 1.01414688,
 1.00977742, 1.00753357, 1.00459017, 0.99995941, 0.99385521],
[0.98144359, 0.97893011, 0.97625526, 0.97312851, 0.96957639,
 0.96556532, 0.96225887, 0.96005345, 0.95908682, 0.95911952,
 0.96008403, 0.96213773, 0.96515593, 0.9692551 , 0.97447081,
 0.98140097, 0.9899708 , 0.99825607, 1.00561173, 1.01191789,
 1.01728779, 1.02173913, 1.02544169, 1.0285267 , 1.02753126,
 1.02200371, 1.01722757, 1.01555068, 1.01432767, 1.01246673],
[0.99310803, 0.99124201, 0.98922661, 0.98702686, 0.98465038,
 0.98221025, 0.97971395, 0.97655316, 0.97278358, 0.96879687,
 0.96542551, 0.96318276, 0.9622062 , 0.96232044, 0.96347749,
 0.96577464, 0.96927392, 0.97383647, 0.9801867 , 0.98871878,
 0.99762353, 1.00565364, 1.01269977, 1.01875103, 1.02371132,
 1.02783528, 1.031433  , 1.02824603, 1.02244837, 1.01856523],
[1.00317488, 1.00124776, 0.99941612, 0.99765962, 0.995804  ,
 0.99389465, 0.99191993, 0.98986032, 0.98765031, 0.98530814,
 0.98247187, 0.9792121 , 0.97546497, 0.97139385, 0.96818056,
 0.96594622, 0.96509019, 0.96537398, 0.96680997, 0.96954344,
 0.97350578, 0.97899906, 0.98679455, 0.99592707, 1.00459322,
 1.01210959, 1.01865097, 1.02406469, 1.02863612, 1.03251322],
[1.01425677, 1.01183719, 1.00944559, 1.00727559, 1.00546559,
 1.00364435, 1.00186977, 1.00007468, 0.99827502, 0.99639922,
 0.99447231, 0.99259004, 0.99061154, 0.98804334, 0.9849803 ,
 0.9815195 , 0.97743402, 0.97338801, 0.97036241, 0.96846041,
 0.96789194, 0.96855553, 0.97061083, 0.97393106, 0.97874279,
 0.98565649, 0.9946047 , 1.00343493, 1.01143304, 1.01835102],
[1.02757502, 1.02415962, 1.02110044, 1.01840197, 1.01571699,
 1.01343394, 1.01139989, 1.00950362, 1.00775346, 1.00604499,
```

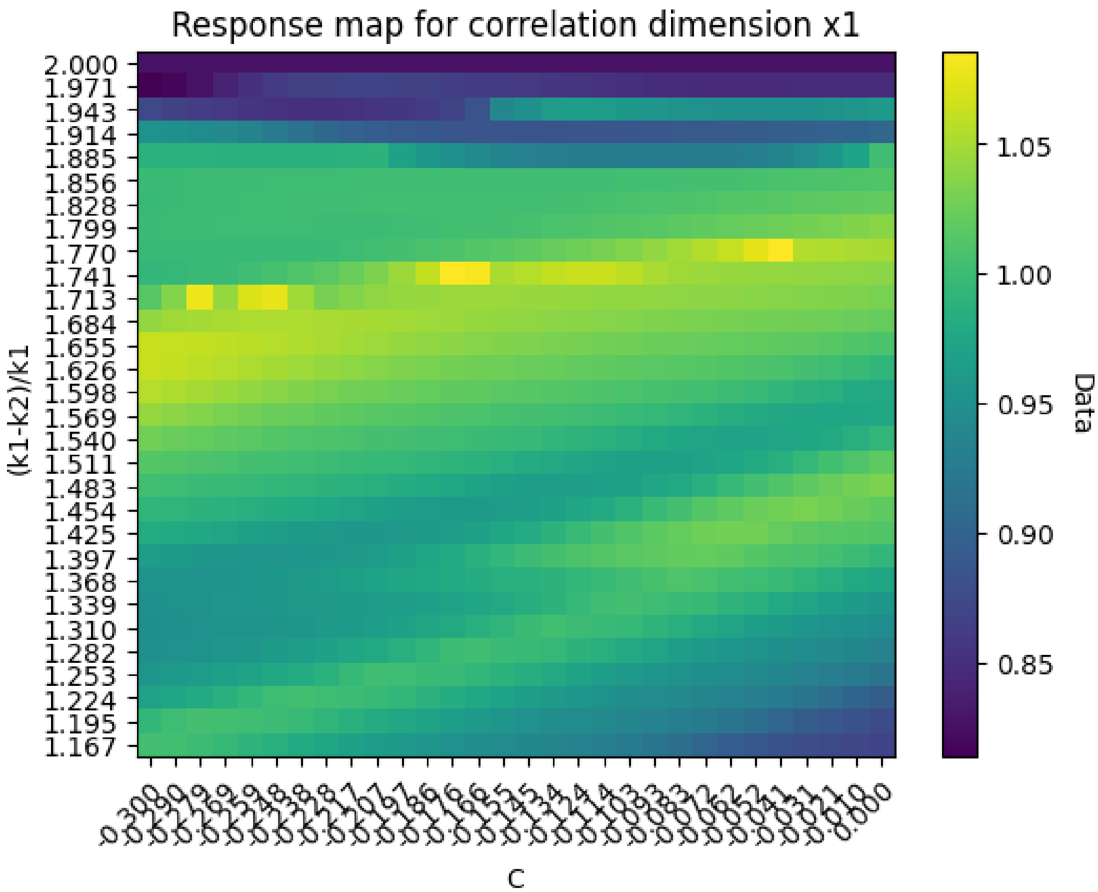
```
 1.00433499, 1.002632  , 1.00083981, 0.99918267, 0.99749826,
 0.99549135, 0.99306776, 0.99033318, 0.98703164, 0.98305775,
 0.9786186 , 0.97495519, 0.97228051, 0.9709844 , 0.97097789,
 0.9722453 , 0.97497269, 0.97920738, 0.98532825, 0.99400335],
[1.04292933, 1.03908937, 1.03512615, 1.03129476, 1.02791128,
 1.02486034, 1.02215028, 1.01963069, 1.0174982 , 1.01548456,
 1.0136802 , 1.01201638, 1.01034761, 1.00858239, 1.00700376,
 1.00541217, 1.00381017, 1.00219409, 1.00026404, 0.99803966,
 0.99541495, 0.99217626, 0.9883351 , 0.98369008, 0.97954982,
 0.97630494, 0.9742678 , 0.9736921 , 0.97442684, 0.97661928],
[1.05524925, 1.05230874, 1.04897739, 1.04535605, 1.04159252,
 1.03790508, 1.03438717, 1.03117515, 1.02823307, 1.02560266,
 1.02349555, 1.02144545, 1.01962353, 1.01798378, 1.01630585,
 1.01463358, 1.01306499, 1.01151115, 1.01013867, 1.00862038,
 1.00700813, 1.00524192, 1.00310204, 1.00034074, 0.99703125,
 0.9930088 , 0.98838232, 0.98394993, 0.9803788 , 0.97784125],
[1.06268875, 1.06069648, 1.05836739, 1.05573856, 1.05280892,
 1.04971697, 1.0464075 , 1.04303514, 1.03963728, 1.03650571,
 1.03376388, 1.03134795, 1.02921406, 1.02727654, 1.02547131,
 1.02382309, 1.02225995, 1.0206565 , 1.01926486, 1.01762922,
 1.01630051, 1.01504304, 1.01363754, 1.01206107, 1.0101435 ,
 1.00791666, 1.00497169, 1.00147747, 0.99734848, 0.99267891],
[1.06282484, 1.06222686, 1.06122038, 1.0598025 , 1.05807944,
 1.05600535, 1.05368301, 1.05120852, 1.04865445, 1.0459079 ,
 1.04329726, 1.04081706, 1.03858979, 1.03646061, 1.03456296,
 1.03281696, 1.03114857, 1.02955032, 1.02804204, 1.02656133,
 1.02513789, 1.02376481, 1.02248629, 1.02129061, 1.02005877,
 1.01872219, 1.01709015, 1.01492564, 1.01234861, 1.00922035],
[1.04186152, 1.04666697, 1.04956605, 1.05126215, 1.052027  ,
 1.05227662, 1.05198773, 1.05133929, 1.05036721, 1.04922831,
 1.04785748, 1.04646601, 1.04488137, 1.04333681, 1.04181825,
 1.04039091, 1.03893342, 1.03753331, 1.03618139, 1.03485536,
 1.03349809, 1.03227953, 1.03109498, 1.02985618, 1.02885015,
 1.02781226, 1.02670664, 1.02537528, 1.02363128, 1.02166769],
[1.01526604, 1.03537701, 1.07948542, 1.04289248, 1.07138675,
 1.07690147, 1.04750506, 1.03010268, 1.03583889, 1.04037393,
 1.04297184, 1.04419523, 1.04462573, 1.04462931, 1.04444691,
 1.04409008, 1.04348936, 1.04286104, 1.04218666, 1.04135983,
 1.04060626, 1.03975903, 1.03903092, 1.03813154, 1.03731471,
 1.03631318, 1.03533366, 1.03430667, 1.03299867, 1.03156733],
[0.99363452, 0.99457861, 0.99656311, 0.99928398, 1.00275164,
 1.00660799, 1.01128877, 1.0166333 , 1.02321207, 1.0323628 ,
 1.04505756, 1.06056983, 1.08545721, 1.08226464, 1.05170446,
 1.05527897, 1.06073604, 1.06446428, 1.06373242, 1.0583128 ,
 1.05133335, 1.04683679, 1.04495813, 1.04388072, 1.04313122,
 1.04231941, 1.04151988, 1.04074356, 1.03968546, 1.03869624],
[0.99861478, 0.99854472, 0.9980584 , 0.99738101, 0.99684948,
```

```
 0.99717573, 0.99813082, 0.99959722, 1.00126273, 1.0032006 ,
 1.00524767, 1.00759762, 1.01008427, 1.01271149, 1.0155852 ,
 1.01870902, 1.02213339, 1.02586728, 1.03010244, 1.03550649,
 1.04141229, 1.04764627, 1.05425561, 1.06188866, 1.07524489,
 1.08589176, 1.05484108, 1.05338601, 1.0507978 , 1.05042339],
[0.99943741, 0.9998152 , 1.00035205, 1.00081011, 1.00099011,
 1.00104649, 1.00096742, 1.00077804, 1.00075136, 1.00075078,
 1.00123924, 1.00206124, 1.00316793, 1.00433125, 1.00569236,
 1.00708583, 1.00854474, 1.01015198, 1.01188897, 1.01376244,
 1.0156899 , 1.01766459, 1.0197635 , 1.02183134, 1.02394301,
 1.0261576 , 1.02852864, 1.03130008, 1.03452085, 1.03749687],
[0.9979701 , 0.99879089, 0.99975486, 1.00047259, 1.00113739,
 1.00169948, 1.00217322, 1.0025706 , 1.00276324, 1.00301987,
 1.00316878, 1.00324105, 1.00331019, 1.00334461, 1.00343639,
 1.00378917, 1.00441173, 1.00521172, 1.0061325 , 1.00721423,
 1.00846984, 1.00976051, 1.01110543, 1.01252132, 1.01391643,
 1.0153701 , 1.01678866, 1.01819589, 1.01956889, 1.02088007],
[0.99906444, 0.99951534, 0.99999594, 1.00038931, 1.00079263,
 1.00107378, 1.0014249 , 1.00163032, 1.00181324, 1.00207016,
 1.00221704, 1.00225009, 1.00230511, 1.00238881, 1.0023807 ,
 1.00237834, 1.00235851, 1.00252062, 1.00292506, 1.00341023,
 1.00390206, 1.00451847, 1.0052773 , 1.00604543, 1.00694085,
 1.00780442, 1.00883923, 1.0098904 , 1.01090496, 1.01197303],
[0.98785739, 0.98738982, 0.98708678, 0.98678513, 0.98660758,
 0.98651193, 0.98648082, 0.98681497, 0.98875444, 0.98937892,
 0.96948533, 0.95905578, 0.95060541, 0.94299016, 0.93685707,
 0.93251833, 0.93029083, 0.92928704, 0.92884186, 0.9285419 ,
 0.92836568, 0.92824816, 0.92841895, 0.92913024, 0.93116657,
 0.93613924, 0.94472661, 0.95658523, 0.97154307, 1.00281299],
[0.9526131 , 0.94981426, 0.94586111, 0.94141255, 0.93516969,
 0.92533174, 0.91478662, 0.90577745, 0.8993904 , 0.89448834,
 0.89074372, 0.88809974, 0.88644278, 0.88543186, 0.88487534,
 0.8845637 , 0.88462514, 0.88498551, 0.88573209, 0.88665107,
 0.88781116, 0.88916746, 0.89062626, 0.89227024, 0.89386055,
 0.89562595, 0.89744157, 0.89936793, 0.90129827, 0.90350005],
[0.87330515, 0.86761918, 0.86270174, 0.85864176, 0.85549462,
 0.85328915, 0.85186496, 0.85137233, 0.85217704, 0.8543875 ,
 0.85753988, 0.86195688, 0.8697045 , 0.88349015, 0.94036939,
 0.95027967, 0.96321384, 0.96608696, 0.96346118, 0.95882531,
 0.95519646, 0.95260384, 0.95115819, 0.95058208, 0.95014117,
 0.94998803, 0.95032857, 0.95203107, 0.95552429, 0.95966036],
[0.81359868, 0.81857369, 0.8264224 , 0.83916493, 0.85159715,
 0.85982808, 0.86495979, 0.86773588, 0.86890176, 0.86858964,
 0.86732174, 0.86584521, 0.86422772, 0.86238413, 0.86040185,
 0.85838094, 0.85628669, 0.85426885, 0.85234512, 0.85065203,
 0.84934792, 0.84831847, 0.84744344, 0.84687533, 0.84682528,
 0.847168  , 0.84778031, 0.84819642, 0.84831416, 0.84818113],
```

```
[0.82721443, 0.82721443, 0.8272144 , 0.82721432, 0.82721452,
 0.82721445, 0.82721416, 0.82721414, 0.82721425, 0.82721434,
 0.82721437, 0.82721442, 0.82721432, 0.82721427, 0.82721444,
 0.8272144 , 0.82721445, 0.8272144 , 0.82721438, 0.8272144 ,
 0.82721441, 0.82721439, 0.82721442, 0.82721445, 0.82721441,
 0.82721446, 0.82721444, 0.82721444, 0.82721424, 0.82721433]])
```

[ ]:
```
K = []
C = []
for i, c in tqdm(enumerate(c_vals)):
    C.append(c)
for j, k2 in enumerate(k2_vals):
    K.append((k2+0.3)/(-0.3))



# plt.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)
fig, ax = plt.subplots()
im = ax.imshow(lyap_map_x1, 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 correlation dimension x1")
# Show plot
plt.show()
```

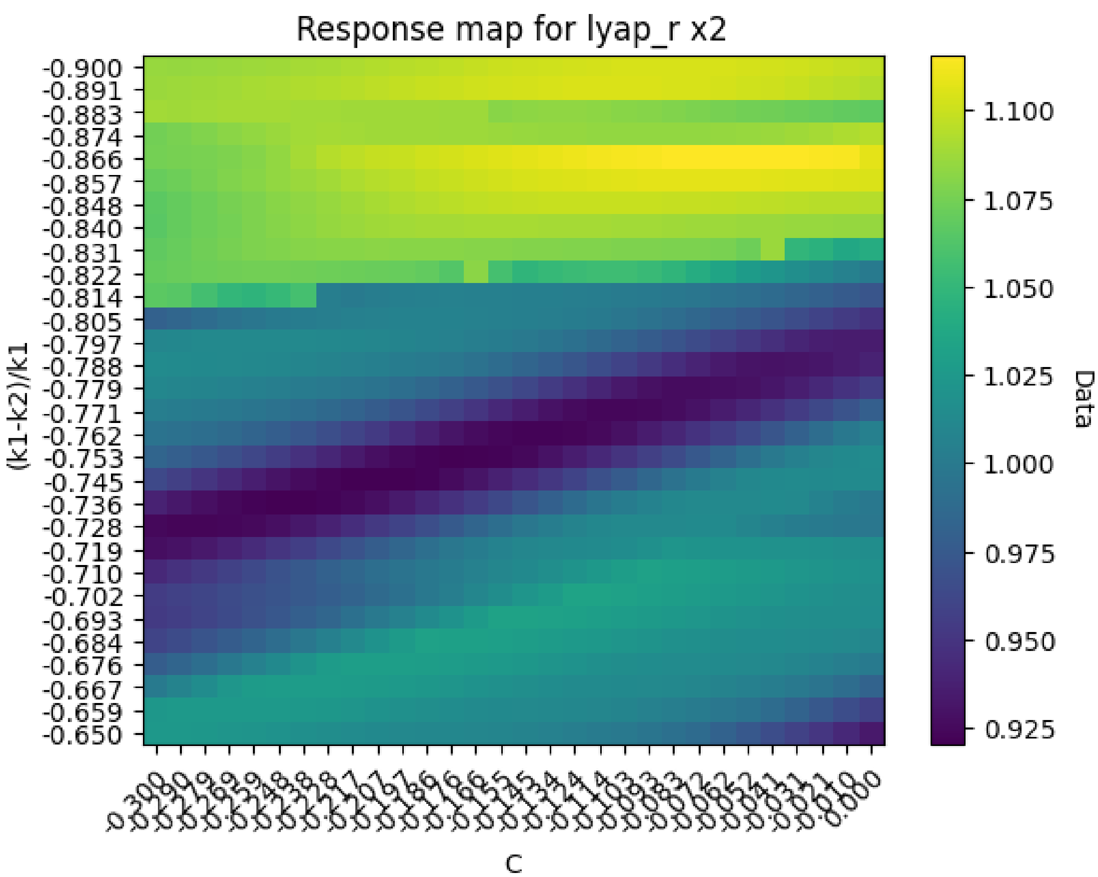
```
30it [00:00, 154581.23it/s]
```



[ ]:
```python
K = []
C = []
for i, c in tqdm(enumerate(c_vals)):
    C.append(c)
# for j, k2 in enumerate(k2_vals):
#     K.append((k2+0.3)/(-0.3))
for k2 in k2_vals:
    K.append(k2)

# plt.imshow(Rmap, cmap='viridis', origin='lower', aspect='auto',)
fig, ax = plt.subplots()
im = ax.imshow(lyap_map_x2, 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 correlation dimension x2")
# Show plot
plt.show()
```

30it [00:00, 197533.94it/s]



[ ]:

```
```

[ ]:

```
k1 = -0.3
k2 = -0.872
c = -0.200
```

```
# 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]
```

[ ]:
```
nolds.lyap_r(s1)
```

```
/Users/timgu/opt/anaconda3/envs/ece1786/lib/python3.8/site-
packages/nolds/measures.py:240: RuntimeWarning: signal has very low mean
frequency, setting min_tsep = 1250
  warnings.warn(msg.format(min_tsep), RuntimeWarning)
```

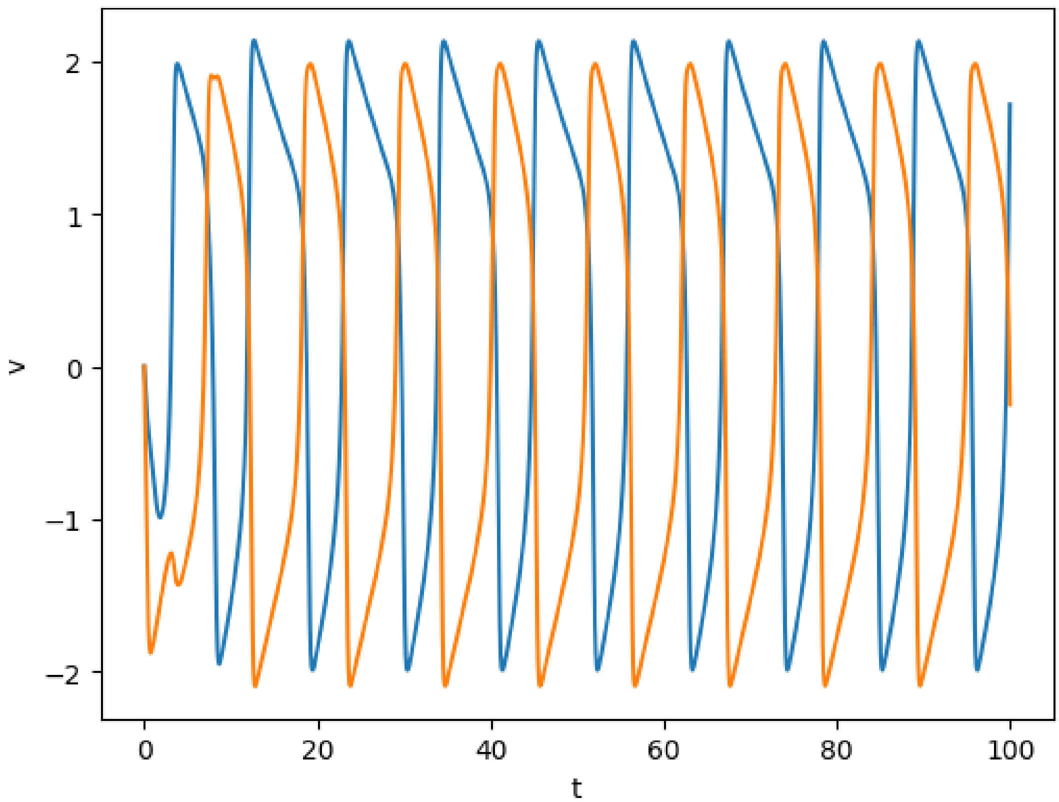
[ ]: 0.0018533649301170285

[ ]:
```
nolds.lyap_r(s2)
```

```
/Users/timgu/opt/anaconda3/envs/ece1786/lib/python3.8/site-
packages/nolds/measures.py:240: RuntimeWarning: signal has very low mean
frequency, setting min_tsep = 1250
  warnings.warn(msg.format(min_tsep), RuntimeWarning)
```

[ ]: 0.0022231453343441072

[ ]:
```
# 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
1
0
−1
−2
v
0
20
40
60
80
100
t