

```
In [79]: import numpy as np
import pandas as pd
import cvxpy as cp
```

(a) Optimum of LS problem

Let $g(x) = ||\tilde{y} - y||_2^2$, then we have $\nabla g(x) = -2H^T \tilde{y} - Hx = 0$ and $\nabla^2 g(x) = 2H^T H > 0$, which leads to the solution of question:

$$\hat{x} = (H^T H)^{-1} H^T \tilde{y}$$

which is the optimum of the solution.

(b) return the matrix of H

```
In [80]: def matrix_return(t, theta):
    H = []
    for i in range(len(theta)):
        H.append(np.cos(theta[i]*t))
    return np.transpose(H)
```

(c)

Aronson's sequence

```
In [88]: y_tilde_data = pd.read_csv('mathias_distress_call_1.csv')
y_tilde = y_tilde_data.values
y_tilde_data_2 = pd.read_csv('mathias_distress_call_2.csv')
y_tilde_2 = y_tilde_data_2.values

dt = 1.0/8192
t = np.arange(0, len(y_tilde)*dt, dt)
theta = [104, 111, 116, 122, 126, 131, 136, 142, 147, 158, 164, 169,
         174, 181, 183, 193, 199, 205, 208, 214, 220, 226, 231, 237,
         243, 249, 254]

H = matrix_return(t, theta)
x_hat_i = np.linalg.solve(np.dot(np.transpose(H), H), np.dot(np.transpose(H), y_tilde))
print('residual error:', np.linalg.norm(y_tilde - np.dot(H, x_hat_i)))

round_ = np rint(x_hat_i)
print('(integer) amplitudes of existing frequencies:', round_[round_.nonzero()])
```

residual error: 2046.298989299373
(integer) amplitudes of existing frequencies: []

Mathias own musical scale

```
In [89]: theta = np.arange(311.127, 311.127+311.127, 311.127/27)
H = matrix_return(t, theta)
x_hat_ii = np.linalg.solve(np.dot(np.transpose(H), H), np.dot(np.transpose(H), y_tilde))
print('residual error:', np.linalg.norm(y_tilde - np.dot(H, x_hat_ii)))

round_ = np rint(x_hat_ii)
print('(integer) amplitudes of existing frequencies:', round_[round_.nonzero()])
```

residual error: 1963.530567863447
(integer) amplitudes of existing frequencies: [4.]

"Fibonacci-on-steroids" sequence

```
In [90]: theta = np.zeros(27)
theta[0] = 150
theta[1] = 175
for i in range(25):
    theta[i+2] = np.ceil(0.5*theta[i+1])+np.ceil(0.8*theta[i])
H = matrix_return(t,theta)
x_hat_iii = np.linalg.solve(np.dot(np.transpose(H),H),np.dot(np.transpose(H),y_tilde))
#H_inv = np.linalg.inv(np.dot(H.transpose(),H));
#x_hat_iii = np.dot(np.dot(H_inv,H.transpose()),y_tilde);
print('residual error:', np.linalg.norm(y_tilde-np.dot(H,x_hat_iii)))

round_ = np rint(x_hat_iii)
print('(integer) amplitudes of existing frequencies:', round_[round_.nonzero()])
```

residual error: 983.5818503735173

(integer) amplitudes of existing frequencies: [2. 6. 3. 4. 1. 7. 6. 1.]

Based on the corresponding solutions, we know "Fibonacci-on-steroids" sequences works the best. We find from the solution that some values are very close to zero. We round the solution and find that this solution is sparse and only $a_5, a_7, a_8, a_{18}, a_{19}, a_{20}, a_{21}, a_{27}$ are nonzero, these values are printed below. Therefore, the message only contains the frequency of $\theta_5, \theta_7, \theta_8, \theta_{18}, \theta_{19}, \theta_{20}, \theta_{21}, \theta_{27}$

(d)

Based on encoding rules, and to make the message meaningful, I believe the message is "Sehr Gut"

(e) With CVXPY package and Lasso

Aronson's sequence

```

In [85]: theta = [104, 111, 116, 122, 126, 131, 136, 142, 147, 158, 164, 169,
                  174, 181, 183, 193, 199, 205, 208, 214, 220, 226, 231, 237,
                  243, 249, 254]

H = matrix_return(t,theta)

lambd_value = 10
x = cp.Variable(27)
q = np.dot(H.transpose(),y_tilde_2[:,0]);
Q = np.dot(H.transpose(),H);

obj = cp.Minimize((1/2)*cp.quad_form(x,Q) - q.T@x+ lambd_value * cp.norm(x,1))
prob = cp.Problem(obj)
prob.solve()

print("status:", prob.status)
print("optimal value", prob.value)
print("optimal var", x.value)
print('residual error:', np.linalg.norm(y_tilde_2-np.dot(H,np.array(x.value).reshape(27,1))))

round_ = np rint(x.value)
print('(integer) amplitudes of existing frequencies:\n',round_[round_.nonzero()])

```

```

status: optimal
optimal value -23903.73007045189
optimal var [-0.21487499 -0.00847243  0.07280993  0.09947972 -0.17233566  0.00236998
 -0.12879463 -0.24122263 -0.20385179 -0.5824162  -0.57086698 -0.31759509
 -0.08591571  0.09933246 -0.21166452  0.09572439 -0.69108564  0.00534629
 -0.53413733 -0.50411963 -0.00159194  0.0788171  -0.12479244 -0.46301546
 -0.11044539 -0.15596631 -0.26432722]
residual error: 9520.504306316883
(integer) amplitudes of existing frequencies:
[-1. -1. -1. -1. -1.]

```

Mathias own musical scale

```
In [86]: theta = np.arange(311.127, 311.127+311.127,311.127/27)
H = matrix_return(t,theta)

lambd_value = 100000
x = cp.Variable(27)
q = np.dot(H.transpose(),y_tilde_2[:,0]);
Q = np.dot(H.transpose(),H);

obj = cp.Minimize((1/2)*cp.quad_form(x,Q) - q.T@x+ lambd_value * cp.norm(x,1))
prob = cp.Problem(obj)
prob.solve()
print("status:", prob.status)
print("optimal value\n", prob.value)
print("optimal var\n", x.value)
print('residual error:', np.linalg.norm(y_tilde_2-np.dot(H,np.array(x.value).reshape(27,1))))

round_ = np rint(x.value)
print('(integer) amplitudes of existing frequencies:\n',round_[round_.nonzero()])
```

```
status: optimal
optimal value
-4167009.5027802205
optimal var
[ 1.00148617e-22  5.71277706e-23 -6.77085100e-23 -1.94400973e-22
 9.23301964e+00  6.45509653e+00  7.90186269e+00 -1.07886027e-22
-6.74941380e-23 -4.27053895e-23  3.27150849e+00  1.02465767e+01
-2.21082613e-22 -1.99783475e-22  3.70866257e+00  5.17341632e+00
-7.12947565e-23  9.64738103e-01 -6.37475920e-23 -3.70987915e-23
 2.38566045e-23  3.65297332e-23  1.79625021e-23 -2.75872356e-23
-8.29337059e-23  2.16721066e+00  7.11053805e+00]
residual error: 8432.519527668373
(integer) amplitudes of existing frequencies:
[ 9.  6.  8.  3. 10.  4.  5.  1.  2.  7.]
```

"Fibonacci-on-steroids" sequence

```

In [87]: theta = np.zeros(27)
theta[0] = 150
theta[1] = 175
for i in range(25):
    theta[i+2] = np.ceil(0.5*theta[i+1])+np.ceil(0.8*theta[i])
H = matrix_return(t,theta)

lambd_value = 1
x = cp.Variable(27)
q = np.dot(H.transpose(),y_tilde_2[:,0]);
Q = np.dot(H.transpose(),H);

obj = cp.Minimize((1/2)*cp.quad_form(x,Q) - q.T@x+ lambd_value * cp.norm(x,1))
prob = cp.Problem(obj)
prob.solve()

print("status:", prob.status)
print("optimal value", prob.value)
print("optimal var", x.value)
print('residual error:', np.linalg.norm(y_tilde_2-np.dot(H,np.array(x.value).reshape(27,1))))

round_ = np rint(x.value)
print('(integer) amplitudes of existing frequencies:\n',round_[round_.nonzero()])

```

```

status: optimal
optimal value -19543.285139751504
optimal var [-0.40217729  0.04187345 -0.52785243  0.16658523  0.28453751  0.0762843
 0.40895657  0.30252246 -0.3425983  -0.0218973  -0.05640451 -0.2139084
 0.3563077  0.24023466  0.41632495  0.2223009  0.19751906  0.43440569
 0.36303533  0.08260647  0.13351879  0.1013268  0.17676502 -0.16813495
 0.05141703  0.24695511  0.00631517]
residual error: 9520.968011623941
(integer) amplitudes of existing frequencies:
[-1.]

```

(i)

I would say we take the $\lambda = 10000$ is a good value, because with this value there is no alias in the message. The reason that the one norm regularizer helps de-noise the signal is that under this regularizer, the solution tends to be sparse, which means it will filter some unimportant signals. i.e. setting the corresponding weights to zero.

(ii)

I would choose the second sequence

(iii)

The message Mathias broadcasted was "RZKOPF GEL", which I believed made no sense.

(f)

If we are going to maintain the variable number to 27, I think there is no easy fix to this problem. But if we can augment the variable to larger dimension, so we allow multiple appearance for each letter and space, then it is possible to encode the message.