In [109]:

```python
import numpy as np
import pandas as pd
import cvxpy as cp
import math
```

(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 $\nabla^2 g(x) = 2H^TH > 0$, which leads to the solution of question:

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

which is the optimum of the solution.

(b) return the matrix of H

In [116]:

```python
def matrix_return(t,theta):
    H = np.outer(t,theta)
    H = np.cos(H)
    return H
```

(c)

Aronson’s sequence

In [117]:

```
y_tilde_data = pd.read_csv('mathias_distress_call_1.csv',header=None)
y_tilde = y_tilde_data.values
y_tilde_data_2 = pd.read_csv('mathias_distress_call_2.csv',header=None)
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: 2046.6129776377231
(integer) amplitudes of existing frequencies: []
```

Mathias own musical scale

In [118]:

```
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: 1971.2745375190368
(integer) amplitudes of existing frequencies: [4.]
```

"Fibonacci-on-steroids" sequence

In [125]:

```
theta = np.zeros(27)
theta[0] = 150
theta[1] = 175
for i in range(2,27):
    theta[i] = math.ceil(0.5*theta[i-1])+math.ceil(0.8*theta[i-2])

H = matrix_return(t,theta)
x_hat_iii = 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_iii)))

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

```
residual error: 100.96291599345125
(integer) amplitudes of existing frequencies: [2. 6. 3. 4. 1. 8. 7. 5.]
```

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 θ_5 , θ_7 , θ_8 , θ_{18} , θ_{19} , θ_{20} , θ_{21} , θ_{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 [126]:

```
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 = 0.01
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 -23229.716229886206
optimal var [-0.21521703 -0.00353822  0.08368956  0.10221334 -0.16853694  0.01085701
 -0.1206886  -0.23036186 -0.20126676 -0.58099642 -0.57316031 -0.31312355
 -0.07366938  0.09966962 -0.20783639  0.09966721 -0.6814735   0.01021345
 -0.52643423 -0.48178886 -0.00666119  0.0925495  -0.11018446 -0.44973632
 -0.10788049 -0.15544431 -0.26628757]
residual error: 9521.232829210603
(integer) amplitudes of existing frequencies:
 [-1. -1. -1. -1.]
```

Mathias own musical scale

In [128]:

```
theta = np.arange(311.127, 311.127+311.127,311.127/27)
H = matrix_return(t,theta)

lambd_value = 0.01
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
 -12814917.773039307
optimal var
 [ 4.85675538  0.12491349  2.10439819 -0.0642973  14.04120424 11.23070402
 12.66863005  3.43940203  0.27610538 -0.42143239  8.15230994 15.10680501
  0.27170356 -0.10013353  8.5317525   9.97387158  0.16070435  5.79752298
  1.06782768 -0.55382922  0.21608757 -0.56366787  3.83147286 -0.12496136
  0.12715705  7.02689868 11.95693671]
residual error: 8066.628552844164
(integer) amplitudes of existing frequencies:
 [ 5.  2. 14. 11. 13.  3.  8. 15.  9. 10.  6.  1. -1. -1.  4.  7. 12.]
```

"Fibonacci-on-steroids" sequence

In [129]:

```
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 = 0.01
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 -19423.112508260954
optimal var [-0.39262496  0.04397385 -0.51970336  0.18338171  0.30182342  0.08864298
  0.42671417  0.30490034 -0.33818958 -0.05398927 -0.04919512 -0.22962772
  0.40636206  0.21341208  0.42150183  0.23243899  0.180572    0.30452787
  0.25112767  0.04696447  0.12193685 -0.00703604  0.19412716 -0.01149291
  0.23975734  0.33252731 -0.1887173 ]
residual error: 9521.63262176316
(integer) amplitudes of existing frequencies:
 [-1.]
```

(i)

I would say we take the $\lambda = 0.01$ is a good value, becasue with this value there is no alias in the message, which means the interpretation is meaningful. In the sametime, this value is small so it won't harm the least square results a lot. Ther 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 "SCHWARZKOPF GEL", which I believe is one kind of gel.

(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 auagment the variable to larger dimension, so we all multiple appearance for each letter and space, then it is possible to encode the message.

In [ ]: