

Solution Review: Parameters of an FID Signal

This lesson discusses the solution to parameters of an FID signal.

WE'LL COVER THE FOLLOWING ^

- Solution
- Explanation

Solution

Here is the FID signal equation for reference:

$$y(t) = [\sin(2\pi f_o t)]e^{-\alpha t}$$

```
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

def RMSE(y1, y2):
    return np.mean((np.square(np.subtract(y1, y2))))

# function for FID signal
def func(x, freq, alpha):
    return np.sin(2 * freq * np.pi * x) * np.exp(-alpha * x)

popt, pcov = curve_fit(func, time, data, p0=(500, 50)) # p0 are initial estimates
print("freq = ", popt[0], "alpha = ", popt[1])
yfit = func(time, *popt) # equivalent to popt[0], popt[1]

# figure and axes settings
fig = plt.figure(figsize=(16, 16))
axes = fig.add_axes([0.1, 0.3, 0.8, 0.4])
axes.set_xlabel('time', fontsize='20')
axes.set_ylabel('signal', fontsize='20')
axes.tick_params(axis="both", labelsize=18)
axes.set_xlim(0, max(time))
axes.set_ylim(-1, 1)

# plotting raw data
axes.plot(time, data, '.', color='b', label='observed')

# plotting fitted data
axes.plot(time, yfit, 'r', label='fit')
```

```
# plotting decay curves
axes.plot(time, np.exp(-1 * popt[1] * time), 'g', linestyle='--', linewidth=3, label='decay c
axes.plot(time, -1 * np.exp(-1 * popt[1] * time), 'g', linestyle='--', linewidth=2)

# setting legend
axes.legend(loc=0, fontsize='xx-large')

# print RMSE
print('RMSE =', RMSE(data, yfit))

# saving the figure
fig.savefig('output/fid.png')
```



Explanation

- In lines 8 - 9, we have defined the function corresponding to the FID signal equation given in the exercise.
- As shown in line 11, the initial guesses are 500 for f_0 and 50 for α . The optimized parameter for f_0 is 999.99 and for α , 75.48. Optimized parameters are very close to the actual parameters.
- Using the optimized parameters, we compute the fitted signal in line 13 and plot it in line 28.
- In lines 31 and 32, we plot the decay curves which are given by the equation $e^{-\alpha t}$ and $-e^{-\alpha t}$.

With this solution review, we are done with scientific algorithms. Let's move on to random variables.