More Advanced Model Fitting and Plotting

PHY2004W **KDSMIL001**

24 Feb 2020

Contents

1 Appendix

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

file = open('PHY2004W Computational\CP2\DampedData1.txt', 'r')
header = file.readline()
lines = file.readlines()

i = 0
N = len(lines)
data = np.zeros((2, N))
u = [0.001]*N
p0 = [0.28, 0.04, 0.4, 30, 0]
name = ['A', 'B', 'gamma', 'omega', 'alpha']

for line in lines:
    line = line.strip()
    columns = line.split()
    data[0, i] = float(columns[0])
    data[1, i] = float(columns[1])
    i += 1

file.close()

plt.errorbar(data[0], data[1], u, fmt='_b', lw=0.5, capsize=2,
   capthick=0.5, markersize=4, markeredgewidth=0.5, label='Data')

def f(t, A, B, gamma, omega, alpha):
    return A+(B*np.exp(-gamma*t))*np.cos((omega*t)-alpha)

# Plotting my best guess
tmodel = np.linspace(0.0, 5.0, 1000)
ystart = f(tmodel, *p0)
# plt.plot(tmodel, ystart, '-g', lw=0.5, label='Initial Guess')

# Plotting the Levenberg-Marquardt best fit
popt, pcov = curve_fit(f, data[0], data[1], p0, sigma=u,
   absolute_sigma=True)
yfit = f(tmodel, *popt)
plt.plot(tmodel, yfit, '-r', lw=0.5, label='Best Fit [1.03]')

# Calculating chi squared etc
dymin = (data[1]-f(data[0], *popt))/u
min_chisq = sum(dymin*dymin)
dof = len(data[0]) - len(popt)

print('Chi Squared:', round(min_chisq, 5))
print('Number of Degrees of Freedom:', round(dof, 5))
print('Chi Squared per Degree of Freedom:', round(min_chisq/dof, 5))
```

```
print()

print('Fitted paramters with 68% C.I.:')
for i, pmin in enumerate(popt):
    print('%2i %-10s %12f +/- %10f'%(i, name[i], pmin, np.sqrt(pcov[i
        ,i])*np.sqrt(min_chisq/dof)))

print()
perr = np.sqrt(np.diag(pcov))
print('Perr:', perr)

print('Correlation matrix:')
print('           ', end='')
for i in range(len(popt)): print('%10s'%(name[i],), end=''),
print()

for i in range(len(popt)):
    print('%10s'%(name[i]), end=''),
    for j in range(i+1):
        print('%10f'%(pcov[i,j]/np.sqrt(pcov[i,i]*pcov[j,j]),), end='
            '),
    print()

plt.legend()
plt.show()
```

Appendix 1: CP2a Code

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

file = open('PHY2004W Computational\CP2\LinearWithErrors.txt', 'r')
header = file.readline()
lines = file.readlines()

i = 0
N = len(lines)
data = np.zeros((3, N))
p0 = [1, 1]

for line in lines:
    line = line.strip()
    columns = line.split()
    data[0, i] = float(columns[0])
    data[1, i] = float(columns[1])
    data[2, i] = float(columns[2])
    i += 1

file.close()
```

```
23
24 def f(x, m, c):
25     return m*x+c
26
27 popt, pcov = curve_fit(f, data[0], data[1], p0, sigma=data[2],
       absolute_sigma=True)
28 dof = len(data[1])-len(popt)
29
30 Npts = 10000
31 mscan = np.zeros(Npts)
32 cscan = np.zeros(Npts)
33 chi_dof = np.zeros(Npts)
34 c = 0
35
36 for mpar in np.linspace(0.5, 0.7, 100, True):
37     for cpar in np.linspace(0.5, 1.7, 100, True):
38         mscan[c] = mpar
39         cscan[c] = cpar
40         dymin = (data[1]-f(data[0], mpar, cpar))/data[2]
41         chi_dof[c] = sum(dymin*dymin)/dof
42         c += 1
43
44 plt.figure()
45
46 ncols = 1000
47
48 plt.tricontourf(mscan, cscan, chi_dof, ncols)
49 plt.colorbar()
50 plt.show()
```

Appendix 2: CP2b Code