# Example-fitting

January 16, 2024

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
```

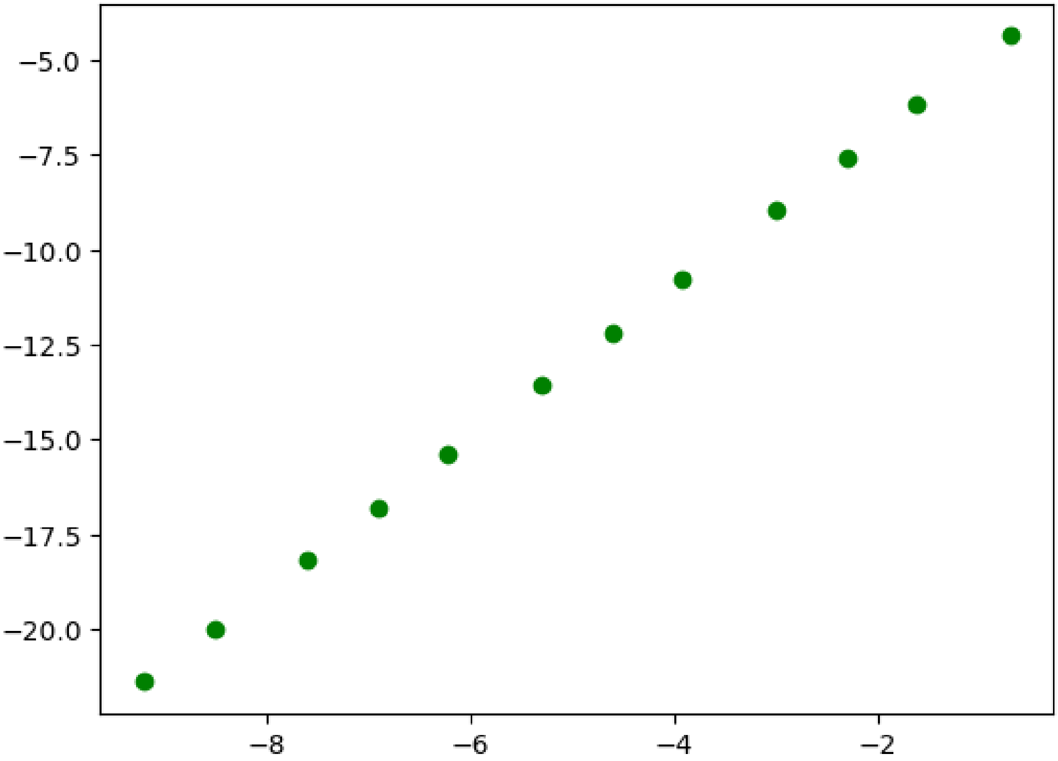
[19]:
```
def forward_difference(g,x,h):
    return (g(x+h) - g(x))/h
def central_difference(g,x,h):
    return (g(x+h) - g(x-h))/(2*h)
def f(x):
    return np.sin(x)
```

[20]:
```
hs = [0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.
 ↪0001]
```

[22]:
```
N = len(hs)
calculated_derivative_using_FD = np.zeros(N)
k = 0
x = 2*np.pi/5
for h in hs:
    #calculated_derivative_using_FD[k] = forward_difference(f,x,h)
    calculated_derivative_using_FD[k] = central_difference(f,x,h)
    k += 1
```

[23]:
```
ydata = np.asarray(np.log(abs(calculated_derivative_using_FD - np.cos(2*np.pi/
 ↪5))))
xdata = np.asarray(np.log(hs))
plt.plot(xdata, ydata,'go')
```

[23]:
```
[<matplotlib.lines.Line2D at 0x7f7fa33dfef0>]
```



[24]:

```
def eq_of_line(xs, p0, p1):
    return p0*xs+p1
```

[25]:

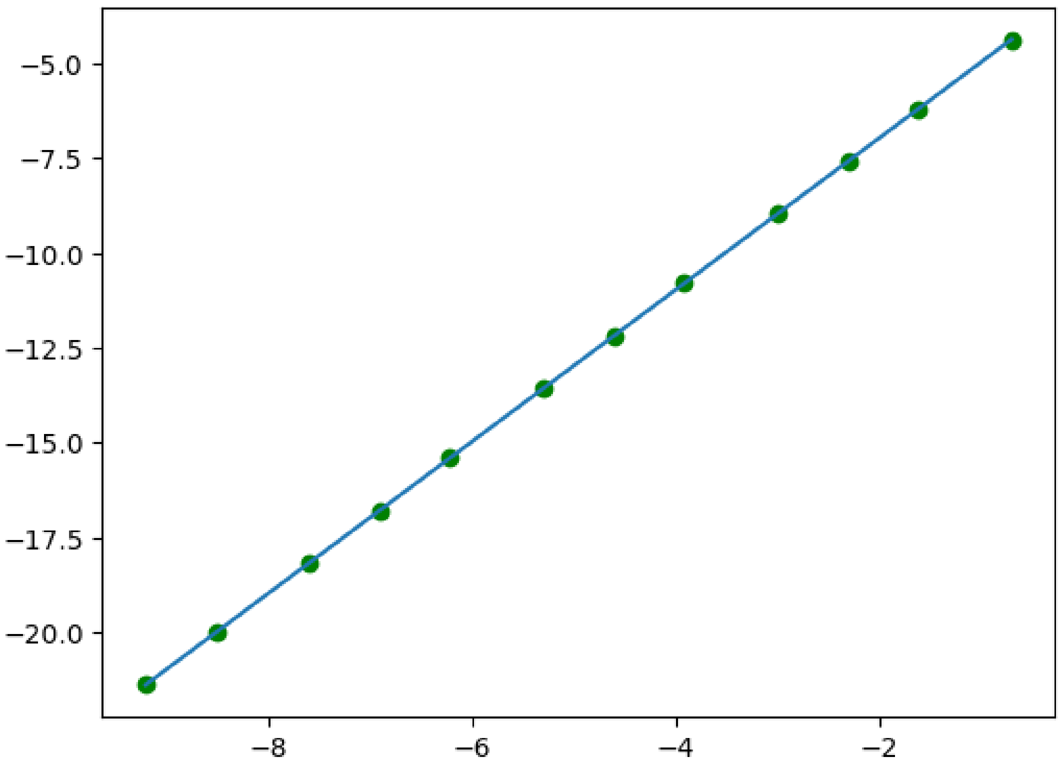
```
params, covariance = curve_fit(eq_of_line, xdata, ydata, p0=[1.0, 1.0])
```

[26]:

```
plt.plot(xdata, ydata,'go')
plt.plot(xdata, eq_of_line(xdata, params[0], params[1]))
```

[26]:

```
[<matplotlib.lines.Line2D at 0x7f7fa34536b0>]
```



```
[27]: params
```

```
[27]: array([ 1.9992408 , -2.97112419])
```

```
[62]: calculated_derivative_using_FD
```

```
[62]: array([0.06344947, 0.21217194, 0.26098901, 0.28511679, 0.29948615,
              0.3042566 , 0.30663807, 0.30806573, 0.30854141, 0.30877922,
              0.30892189, 0.30896944])
```

```
[63]: np.cos(2*np.pi/5)
```

```
[63]: 0.30901699437494745
```

```
[ ]:
```