

ENR145 Assignment #8: curve fitting in python

Due: 3/13/26 9:00 pm

Try out curve fitting tool in python:

Down load and check this:

Module 2: Multiphysics Simulation (3 weeks)

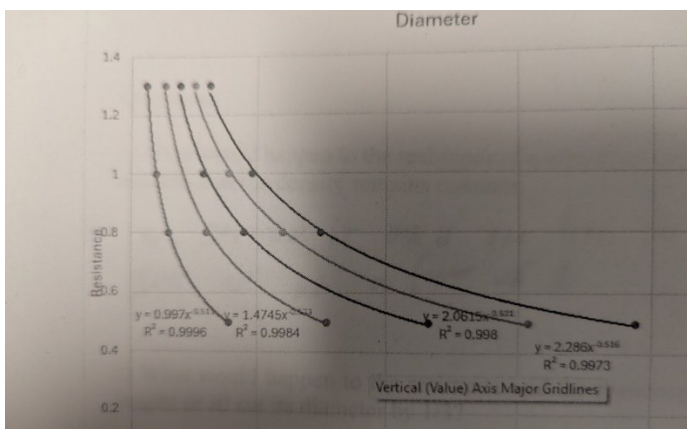
Week 7: [Slides #8](#) [Assignment #6](#)

Week 8: [Slides #9](#) [Slides #10](#) [Curve fitting speedrun](#)

Week 9: [\[Slides\]](#) [\[Prelabs\]](#) [\[Assignment\]](#)

```
import numpy as np
import matplotlib.pyplot as plt # for plot
from scipy.optimize import curve_fit # for curve fitting
from sklearn.metrics import r2_score # to calculate the R-Squared
# run this box to import the library
```

Now re-purpose the codes to do the following curve fitting:



With both *inverse fit* and *inverse square* fit.

Here're the data:

```
# Data for Curve 1
```

```
x_data_1 = np.array([1.3, 1.0, 0.8, 0.5])
```

```
y_data_1 = np.array([0.60, 0.99, 1.54, 3.84])
```

```
# Data for Curve 2
```

```
x_data_2 = np.array([1.3, 1.0, 0.8, 0.5])
```

```
y_data_2 = np.array([1.27, 2.10, 3.23, 7.94])
```

```
# Data for Curve 3
```

```
x_data_3 = np.array([1.3, 1.0, 0.8, 0.5])
```

```
y_data_3 = np.array([2.42, 4.01, 6.15, 15.16])
```

```
# Data for Curve 4
```

```
x_data_4 = np.array([1.3, 1.0, 0.8, 0.5])
```

```
y_data_4 = np.array([2.99, 4.96, 7.65, 19.02])
```

To prove your effects, submit a slide with screenshot of crucial steps, and the curve fitting graph generated in python. Then, describe you finding about the curve fitting in a take home message.

Generative AI can provide the boiler plate codes very fast. I am expecting you spend less than **10 min** getting the graph done.