

Python for Scientific Data Analysis

Homework - Week 6

1. Root-Finding/Minimization (LM algorithm)

Consider the function $f(x) = x^2 + -5 * x + 1.5 * \cos(x^2) + \sin(x)$

- Find the roots of this function using the Levenberg-Marquardt algorithm
- Verify your answer by calculating $f(x)$ at the value of these roots

2. Root-Finding/Minimization (Newton-Raphson)

Consider the function $2x^3 + 3x^2 - 4x - 5$

- Compute the value of this function at integers 1, 2, 3, 4, and 5.
- Based on the above give a starting guess for the integer closest to the root of this function
- Use the definition of the Newton-Raphson method, to estimate the first update of the root of this function from:

$$x_1 = x_o - f(x_o)/f'(x_o)$$

(Note: it is easiest to define two functions -- `func(x)` and `funcd(x)` -- corresponding to the function and its derivative at some value `x` and call these functions in your manual N-R first estimate

- Compute the real root estimate from the Newton-Raphson method using again your starting integer value.
- Verify that your solution is indeed a root of this function
- How close were you to the solution from just the first iteration?

3. Curve Fitting

- start with `inputdata.txt` located in the week5 homework folder
- read in these data with `np.loadtxt`
- visualize the data using a very simple matplotlib call:

i.e. `import matplotlib.pyplot as plt`

`plt.scatter([variable name for x],[variable name for y])`

What kind of function does this look like? (note: the answer is functionally simple and involves two coefficients and one variable)

- Fit the data with `curve_fit`. To do this, define a simple function whose form is based on your answer to the previous item. Report the values for the two coefficients needed to fit the data.
- Compare your solution by plotting the data (as in item 3) with the functional fit overplotted (don't worry about nice-looking formatting: just the data + function are good enough)
- Why might the fit not look perfect?