NOVEMBER 2023

TEXAS
The University of Texas at Austin

CURVE FITTING GUI WITH BOUNDARY CONDITIONS

Final Project – ME396P

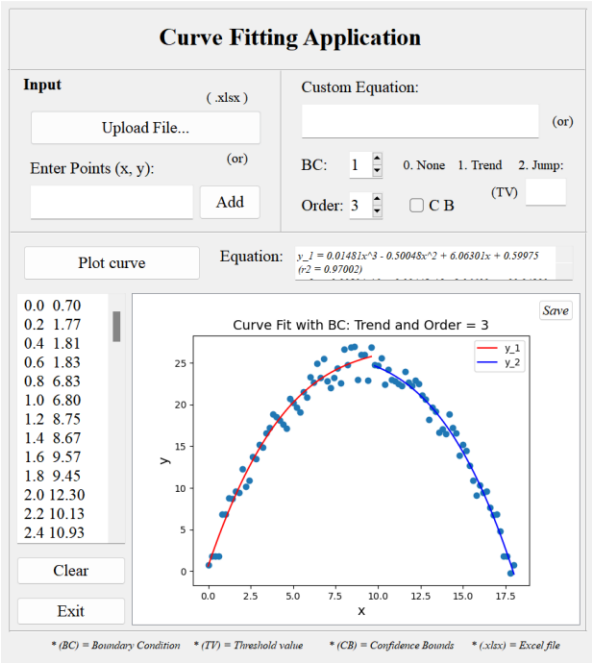
TEAM PYNAPPLES
(G03)

Anastasia Timoshenko, John Tanir, Jahnavi Valivi Reddy

The University of Texas at Austin

Project Objectives

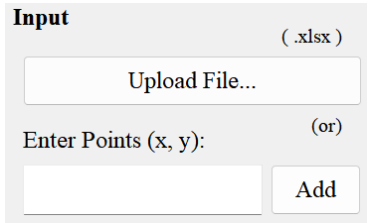
Design of a Curve-fitting GUI Tool that enables user inputs for fit parameters (order, custom equation, etc.) to fit a set of points by detecting the specified boundary conditions.

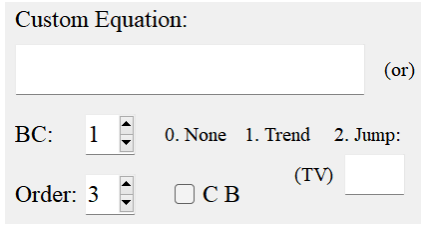


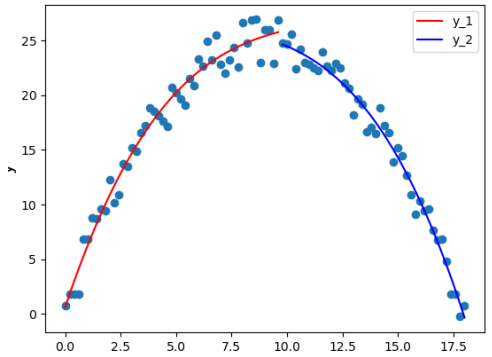
https://github.com/Jaanv99/Team-Pynapples.git

Project Requirements:

1. User Interface Design
2. Functionality:
   - Enables data input methods,
   - Selection of fit parameters,
   - Data Visualization,
   - Multi-use/Multi-click
3. Documentation







Approach

- Packages:
  - *PyQt5, Random; Numpy; Matplotlib.pyplot; Pandas; Matplotlib.backends.backend_qt5agg, Scipy.optimize (curve_fit); Sklearn.metrics (r2_score)*
- Delegation of effort:
  - GUI Design, extracting user inputs and code combination, BC 2: *Jahnavi*
  - Boundary Condition (BC 1), curve fitting, and text input algorithm: *Anastasia*
  - Evaluation of Fit (Prediction and Confidence Intervals): *John*

Approach

- Algorithms:
  - A few distinct algorithms had to be developed:
    - Boundary condition location.
    - Detecting type of Boundary condition.
    - Conversion of text input into a usable function (custom equation).
    - Curve Fitting for different types of curves (polynomials of different degrees/custom equation).

Curve Fitting

- 2 Basic Cases:
  - Polynomial fit.
    - Utilized np.polyfit(x, y, deg).
    - User selects degree and type of BC – fits based on the boundary condition.
  - Custom equation fit.
    - Utilized Scipy.optimise.curve_fit(func, x, y)
    - Converted the custom equation into a usable function with x and *nums (tuple of coeff. values) as the only inputs.
    - Performs curve fitting using the provided equation and x and y points.

Curve Fitting

- Custom function from user input.
- Utilizes exec() to convert str to variable name.
- Utilizes eval() to evaluate str expression as an equation.

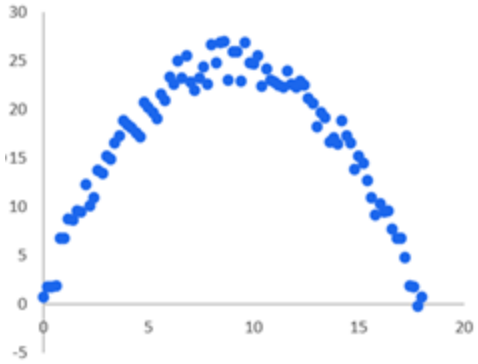
- In [1]: runfile(...Final Project')
- Out [1]: 18

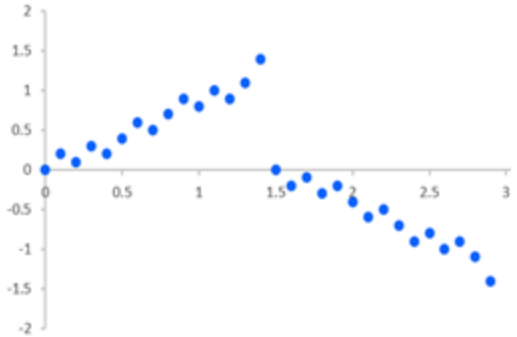
```
text = 'y = m*x^2 + c*x + b'

text = text.replace('^', '**')

def func(x, *nums):
    # Make a list of possible coefficent names without x and e:
    alph = list(map(chr, range(97, 123)))
    alph.remove('e')
    alph.remove('x')

    # Pulls in the text input from global. Need to be careful not to change the text variable
in the code.
    global text

    # Split off the 'y = ' from the text:
    eq = text.split('= ')[1]

    i = 0

    # For each character in the text, if the character is in the alphabet list (coefficient n
assign the next value in the coefficients tuple (*nums) to it and create a cariable.
    for char in eq:
        if char in alph:
            exec(f"{char} = {nums[i]}")
            i = i + 1

    # Use eval on the text to evaluate the equation and return the result.
    result = eval(eq)
    return result

xval = 3

solved = func(xval, 1, 2, 3)

print(solved)
```

Boundary Conditions

Two types of boundary conditions:

- Maximum/minimum point on a curve.
  - Utilized moving average to "smooth" the data.
  - Max/min point of the "smooth" data used as BC.
- Discontinuous point on the curve.
  - Significant gaps in the data were identified.
  - The "threshold value" (TV) determines the BC.
  - TV is user-specified or auto-generated (from standard deviation of errors).





Boundary Conditions

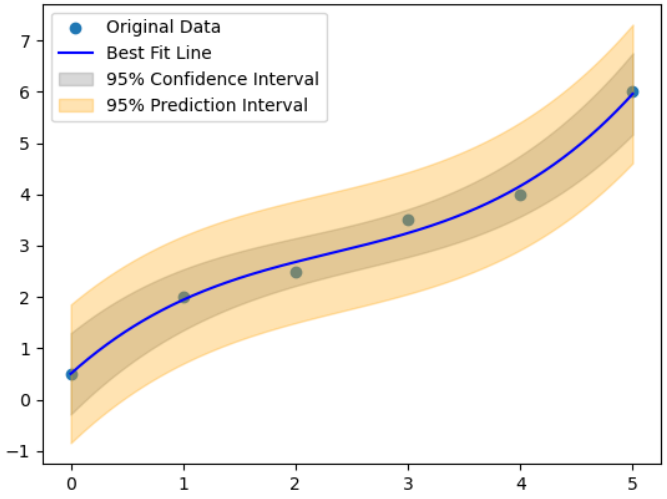
- Utilized pandas, numpy to run moving average and standard deviations:
  - Moving average isn't perfectly accurate.
  - Multiple averages improve the fit.
  - Perfect fits and BC identification remain challenging.

```
# Make the x and y sets into numpy arrays:
x_arr = np.array(x)
y_arr = np.array(y_noisy)

# Convert them to a pandas dataframe:
data = pd.DataFrame({'x': x_arr, 'y': y_arr})

# Apply moving average using average data rolling from pandas:
data_smooth = data.rolling(window = 5).mean()
data_smooth2 = data_smooth.rolling(window = 8).mean()

# Plot the smoothed data sets:
plt.plot(x, data_smooth['y'], 'r-', label = 'Smooth Data 1')
plt.plot(x-1, data_smooth2['y'], 'b-', label = 'Smooth Data 2')

# Find the maximum point - the boundary condition - in the smoothed data set,
and plot it to check:
max_id = list(data_smooth2[['y']].idxmax())[0]

plt.plot(list(data_smooth2['x'])[max_id], list(data_smooth2['y'])[max_id],
'xr', ms = 10)
plt.legend()

# Split the original data set using the new max_id index:
x_l = x[:max_id+1].copy()
x_r = x[max_id:].copy()
```

Evaluation of Fit

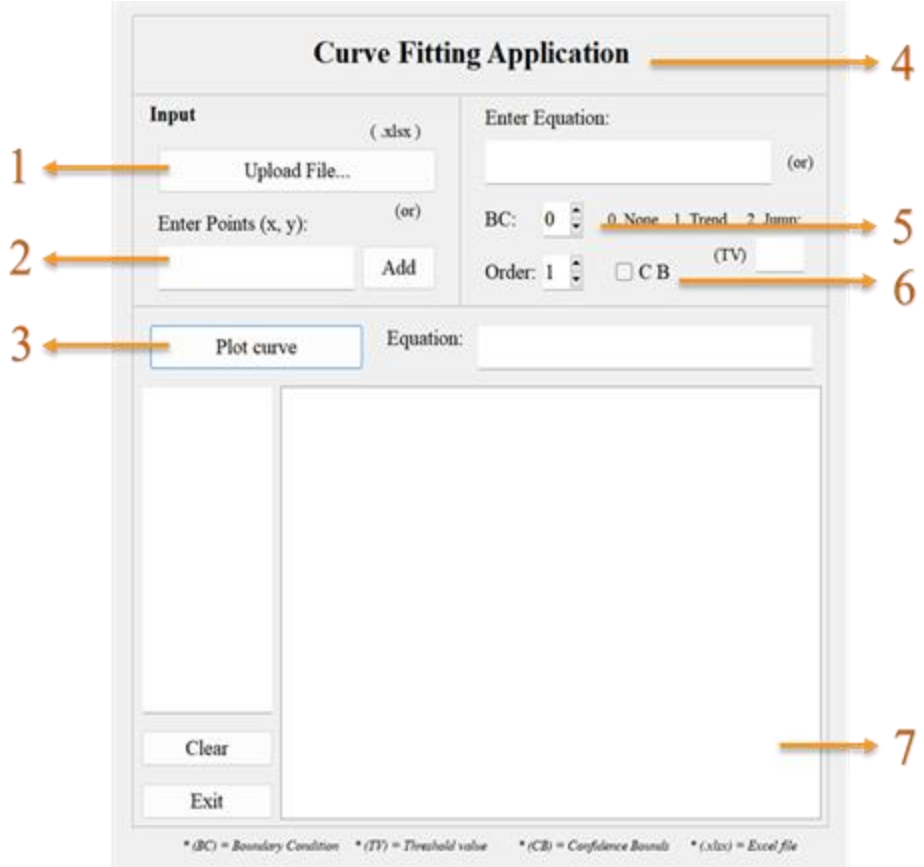
Techniques:

- *Sum of squares errors*, *error of the estimate*, and *r^2* were calculated.
- Checks how closely the data matches the fitted polynomial
- Statistical analysis: To determine the *confidence and prediction intervals*.
- Confidence interval: Provides an interval estimate with a measure of reliability about the mean
- Prediction interval is for reliability about future observations.



GUI Design:

	Object	Functioning
1.	QPushButton (auto default)	*self.Button.clicked.connect(self.fun)*
2.	QTextEdit	*self.Text.toPlainText()* *self.Text.setText()*
3.	QPushButton (default)	*Same as 1*
4.	Qlabel	*Labelling*
5.	QSpinBox	*self.SpinBox.value()*
6.	QCheckBox	*self.CheckBox.state.connect(self.fun)*
7.	QGraphicsView	*self.graphicsView.setScene(self.scene)*



Project Results:

(Demo)

Go to code .....

Project Links:

- Github - https://github.com/Jaanv99/Team-Pynapples.git
- Code Documentation and required files in the repository

References

- https://learnpython.com/blog/average-in-matplotlib/
- https://learnpython.com/blog/filter-rows-select-in-pandas/
- https://stackoverflow.com/questions/74364841/how-to-find-peaks-in-a-noisy-signal-or-estimate-its-number
- https://www.geeksforgeeks.org/how-to-calculate-moving-average-in-a-pandas-dataframe/
- https://stackoverflow.com/questions/13728392/moving-average-or-running-mean
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html
- https://pandas.pydata.org/docs/getting_started/intro_tutorials/03_subset_data.html
- https://stackoverflow.com/questions/29949757/creating-pandas-dataframe-between-two-numpy-arrays-then-draw-scatter-plot
- https://www.geeksforgeeks.org/get-the-index-of-maximum-value-in-dataframe-column/
- https://stackoverflow.com/questions/16060899/alphabet-range-in-python
- https://stackoverflow.com/questions/9685946/math-operations-from-string
- https://www.geeksforgeeks.org/convert-string-into-variable-name-in-python/
- https://doc.qt.io/qtforpython-6/tutorials/datavisualize/index.html
- https://stackoverflow.com/questions/30964669/qt-designer-qgraphicsview-load-image
- https://www.appsloveworld.com/coding/python3x/6/how-to-get-confidence-intervals-from-curve-fit?expand_article=1

Thank you!

Questions?