

README

Project 1 - John Akujobi

Numerical Differentiation Error Analysis Web App

Overview

This interactive web app is built with Streamlit and is designed to analyze and visualize errors in numerical differentiation methods. The project focuses on approximating the derivative of the function at using two common methods:

- **Forward Difference:**
- **Central Difference:**

For each method, the app computes:

- **Actual Error:** The difference between the numerical derivative and the exact derivative ().
- **Truncation Error:** The error from approximating the derivative by neglecting higher-order terms in the Taylor series.
- **Rounding Error:** The error introduced by the finite precision of computer arithmetic.

It then visualizes these errors on log-log plots so that you can see how they change as the step size varies. Also, the app calculates the optimal values for minimizing the total error.

Features

- **Interactive Controls:**
Adjust the range for (expressed as 10^x), the number of points, and the machine epsilon (ϵ) via the sidebar.
- **Theoretical Background:**
An expandable section explains the mathematics behind the forward and central difference methods, including the expected orders of truncation and rounding errors.
- **Visualization:**
Log-log plots display the actual error, truncation error, and rounding error for each method, making the power-law relationships clear.
- **Optimal Error Calculation:**
The app computes and displays the optimal step size that minimizes the total error for both differentiation methods.

- **Comparison Table:**

A summary comparing the properties of the forward and central difference methods is provided.

Installation

Prerequisites

- **Python 3.13**
- **Pip** (Python package installer)

Dependencies

The project requires the following Python libraries:

- `numpy`
- `matplotlib`
- `streamlit`

You can install the dependencies using pip:

```
pip install numpy matplotlib streamlit
```

Usage

1. **Download the Repository:**

Save the provided code

2. **Run the Application:**

Open a terminal, navigate to the directory containing `streamlit_app`, and run:

```
streamlit run streamlit_app.py
```

3. **Interact with the App:**

- Use the sidebar to set the minimum and maximum exponent for (e.g.,), choose the number of points, and adjust the machine epsilon.
 - View the log-log plots that display the actual error, truncation error, and rounding error for both the forward and central difference methods.
 - Expand the "Theoretical Background" section to review the underlying mathematics.
 - Check the optimal values and comparison table for a summary of the error analysis.
-

Explanation of the Code

1. Configuration & Styling

- `configure_page()` : Sets up the page title, layout, and custom CSS for a clean, professional look. It also loads MathJax to render LaTeX equations.

2. Theoretical Background

- `show_theory()` : Displays an expandable section with an explanation of the forward and central difference methods, including their truncation and rounding errors.

3. User Controls

- `get_user_inputs()` : Creates sidebar widgets (sliders and number inputs) for the user to define the range of values, the number of data points, and the machine epsilon ().

4. Error Calculations

- `calculate_errors()` : Loops over an array of values to compute:
 - The approximate derivative using both forward and central differences.
 - The actual error (difference from the exact derivative).
 - Estimated truncation error (using Taylor series approximations).
 - Estimated rounding error (based on the machine epsilon).

5. Visualization

- `create_error_plot()` : Uses Matplotlib to generate log-log plots of the errors. These plots help illustrate the relationships between the error components and the step size .

6. Optimal Value Calculation

- `calculate_optimal_values()` : Computes the optimal for minimizing total error:
 - For forward difference:
 - For central difference:

7. Main App Flow

- `main()` : Orchestrates the overall app functionality by:
 - Configuring the page.
 - Displaying the theoretical background.
 - Accepting user inputs.
 - Calculating errors and generating plots.
 - Displaying optimal values and a comparison table summarizing the two methods.

