Finite Differences vs. Floating Point

1分

In this problem, you're given a function `f` and its derivative `df` as a function. For a large number of different values of the grid spacing h, you will use second-order centered finite differences to compute an approximation of the derivative:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

For each of the point counts given in `n_values`, compute the finite difference approximation to f' everywhere except at the two endpoints. Compute the relative error in the ∞-norm and plot the result, using the starter code given.

What do you observe?

INPUT:

- `f`, a (reasonably wiggly) function.
- `df`, the derivative of `f`.
- `n_values`, a list of point counts to try. For each entry n in this list, compute the second order finite differences on the grind between $[0, 1]$ with n equispaced points.

OUTPUT:

- `a`, `b`, the final ends of your bracket.

评分代码 (点击查看)

起始代码 (点击查看)

回答*

```
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as plt

h_values = []
rel_err_values = []
for n in n_values:
    x = np.linspace(0, 1, n).astype(np.float32)
    h = x[1] - x[0]
    h_values.append(h)

    # Evaluate 2nd centered order finite differences of f at x.
    # Compute error against df at x in the infinity norm.

```

按F9以打开/关闭全屏模式. 在 用户信息 (/profile/) 中设置编辑器模式.

保存回答 提交用于评分的回答

(您仍然可以在提交**本问题**后修改回答)