Міністерство освіти і науки України

Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

ЗВІТ

з лабораторної роботи №3

з дисципліни «Методи Оптимізації та Дослідження Операцій» на тему:

«Поліноміальна апроксимація та методи точкового оцінювання»

Виконав

Студент О. А. Онищенко

Прийняли

Викладач Л. Ю. Дейнега

2024

ПОЛІНОМІАЛЬНА АПРОКСИМАЦІЯ ТА МЕТОДИ ТОЧКОВОГО ОЦІНЮВАННЯ

Мета роботи

Вивчити методи пошуку оптимуму з використанням квадратичної апроксимації й точкового оцінювання; навчитися застосовувати метод Пауела для оптимізації об'єктів керування.

Постановка задачі

Розробити программну реалізацію методу пошуку оптимуму з використанням квадратичної апроксимації й точкового оцінювання.

Функція:

$$(x-2)^2$$

Інтервал: $[-1{,}3]$

Результати виконання

Код програми

```
import numpy as np
import matplotlib.pyplot as plt
from rich.console import Console
from rich.traceback import install
from scipy.optimize import minimize_scalar

install()
console = Console()


def f(x):
    return (x - 2) ** 2
```

```python
def quadraticApproximation(p1, p2, p3):
    v1, v2, v3 = f(p1), f(p2), f(p3)
    return (
        0.5
        * ((v2 - v1) * (p3**2 - p1**2) - (v3 - v1) * (p2**2 - p1**2))
        / ((v2 - v1) * (p3 - p1) - (v3 - v1) * (p2 - p1))
    )


def searchPowell(p1, step, epsilon):
    points = [p1]
    while True:
        p2 = p1 + step
        v1, v2 = f(p1), f(p2)
        p3 = p1 + 2 * step if v1 > v2 else p1 - step
        v3 = f(p3)
        vMin, pMin = min((v1, p1), (v2, p2), (v3, p3))
        pPrime = quadraticApproximation(p1, p2, p3)

        points.append(pPrime)

        if abs(vMin - f(pPrime)) < epsilon and abs(pMin - pPrime) <
epsilon:
            return pMin, vMin, points
        p1, p2, p3 = sorted(
            [
                pMin,
                pPrime,
                pMin + step if pMin == pPrime else pMin - step,
            ]
        )


def main() -> None:
    bounds: tuple[float, float] = (-1, 3)

    with console.status("Optimizing...", spinner="point"):
        pMin, vMin, points = searchPowell(-1, 0.1, 1e-6)
        resPowell = f"{pMin:.2f}"
        resScalar: float = f"{minimize_scalar(f, bounds).x:.2f}"

    console.print(f"Powell's Method: {resPowell}")
    console.print(f"Scalar Method: {resScalar}")
    console.print()
    console.print(
        "[green bold]Correct answer found![/green bold]"
        if resPowell == resScalar
        else "[red bold]Doesn't match![/red bold]"
    )
```

```
    def drawResults():
        x = np.linspace(-1, 3, 100)
        plt.figure(figsize=(10, 6))
        plt.plot(x, f(x), label="f(x) = (x-2)^2")
        plt.plot(points, [f(p) for p in points], "ro-", label="Powell's
Method")
        plt.plot(pMin, vMin, "go", label="Powell's Minimum")
        plt.title("Visualization of Powell's Method for function (x-
2)^2")
        plt.xlabel("x")
        plt.ylabel("f(x)")
        plt.legend()
        plt.show()

    drawResults()


if __name__ == "__main__":
    main()
```

Результати роботи програми

```
Powell's Method: 2.00
Scalar Method: 2.00

Correct answer found!
```

Рисунок 1.1 – Результати розрахунку мінімумів по завершенню роботи програми

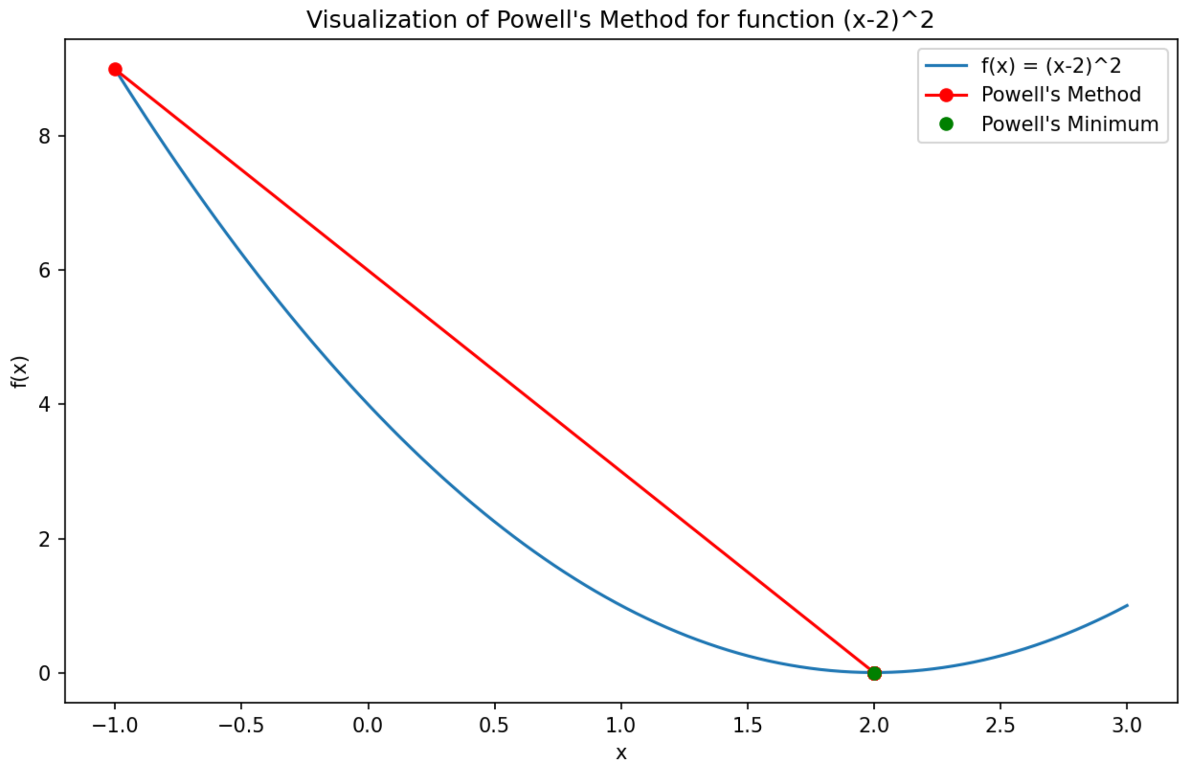

Рисунок 1.2 – Візуалізація пошуку за допомогою методу Пауела

Висновки

Таким чином, ми вивчили методи пошуку оптимуму з використанням квадратичної апроксимації й точкового оцінювання; а також навчилися застосовувати метод Пауела для оптимізації об'єктів керування.

Контрольні питання

У чому полягає основна ідея методів поліноміальної апроксимації й точкового оцінювання?

Основна ідея методів поліноміальної апроксимації полягає в тому, щоб представити складну функцію у вигляді полінома, з яким легше працювати. Це робиться шляхом знаходження полінома, який якомога ближче підходить до функції на інтервалі, що нас цікавить. Точкові методи оцінки, з іншого боку, використовуються для отримання єдиного "найкращого" прогнозу деякої величини, що нас цікавить. Наприклад, метод найменших квадратів, який мінімізує суму квадратів залишків, може бути використаний для оцінки параметрів поліноміальної апроксимації.

Сформулюйте необхідні умови ефективної реалізації методу пошуку, заснованого на поліноміальній апроксимації.

Для того, щоб метод пошуку, заснований на поліноміальній апроксимації, був ефективним, зазвичай потрібно виконати декілька умов:

- Функція, що апроксимується, повинна бути добре керованою, тобто гладкою і неперервною на інтервалі, що нас цікавить.
- Степінь полінома повинен бути обраний належним чином. Занадто низький степінь полінома може погано відображати поведінку функції, в той час як занадто високий степінь полінома може надмірно підганяти дані.
- Точки, в яких оцінюється функція (для побудови полінома), слід вибирати з розумом. Вони повинні покривати інтервал, що нас цікавить, але водночас не бути надто близько одна до одної, щоб уникнути числової нестабільності.

По заданих точках і відповідних значеннях функції виведіть формули для оцінки параметрів апроксимуючого квадратичного полінома й оцінки координати точки оптимуму.

За заданими точками $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ та відповідними значеннями функції можна оцінити параметри a, b та c апроксимуючого квадратичного полінома $y = ax^2 + bx + c$ за допомогою системи рівнянь:

$$ax_1^2 + bx_1 + c = y_1,$$
$$ax_2^2 + bx_2 + c = y_2,$$
$$ax_3^2 + bx_3 + c = y_3.$$

Цю систему можна розв'язати за допомогою таких методів, як виключення Гауса або правило Крамера.

Маючи параметри a, b і c, ми можемо знайти координати точки оптимуму (тобто мінімум або максимум полінома), прирівнявши похідну полінома до нуля і розв'язавши для x, що дає $x_{\text{opt}} = -\frac{b}{2a}$.