

Національний Технічний Університет України
«Київський Політехнічний Інститут»
Науково-навчальний комплекс
«Інститут Прикладного та системного аналізу»
Кафедра системного проектування

Лабораторна робота №5
з курсу **«МЕТОДИ ОПТИМІЗАЦІЇ»**
на тему
“Дослідження методів одновимірного пошуку”

Виконав
студент III курсу
групи ДА-42
Приставський Костянтин
варіант 18

Київ – 2016

Варіант №18

Завдання:

1) Сформувати індивідуальну квадратичну функцію $F5$ виду:

$$F5 = x^2 + (p+q)x + pq$$

де p – перша, а q – друга цифра в номері варіанту. Якщо номер варіанту менше 10, то $p=0$.

Дослідити квадратичну $F5$ та індивідуально задану функцію однієї змінної $F6$ $F(x) = \sin^4 x + \sqrt{\text{abs}(x)}$, обрану за варіантом, в приблизних межах $[-10.0; 10.0]$.

4.2) За допомогою графіків функцій, визначити початкові інтервали пошуку $[a^{(0)}, b^{(0)}]$, які містять мінімуми функцій (локальний або глобальний).

4.3) Скласти програму пошуку мінімуму функції однієї змінної методом золотого перетину. Результати розрахунку мінімуму двох функцій $F5$ та $F6$ з точністю $\varepsilon = 1e-3$ зведені у таблицю.

4.4) Розрахувати значення мінімуму функцій $F5$, $F6$ аналітично та порівняти з результатами п.4.3.

квадратичної $F5$ та індивідуальної функцій $F6$ на ПЕОМ (табл. 5.2);

5.8) висновки.

Хід роботи

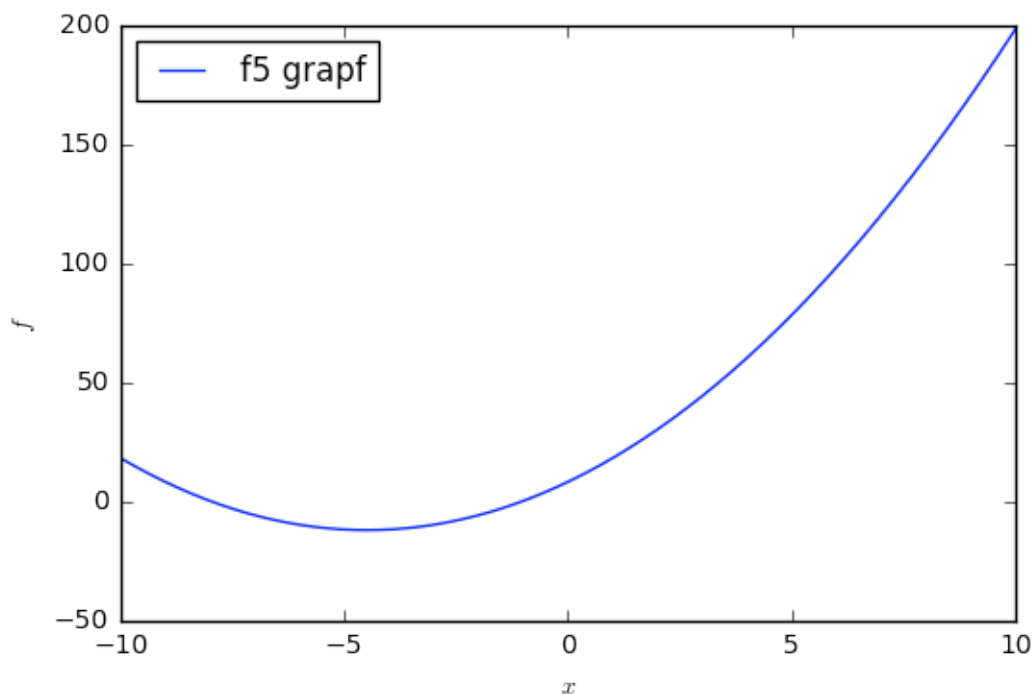
1. Сформулюємо квадратичну функцію

$$F5(x) = x^2 + (p + q)x + pq$$

В нашому випадку, задаємо f5 як:

```
def f5(x):  
    return x**2+9*x +8
```

Будуємо графік функції:

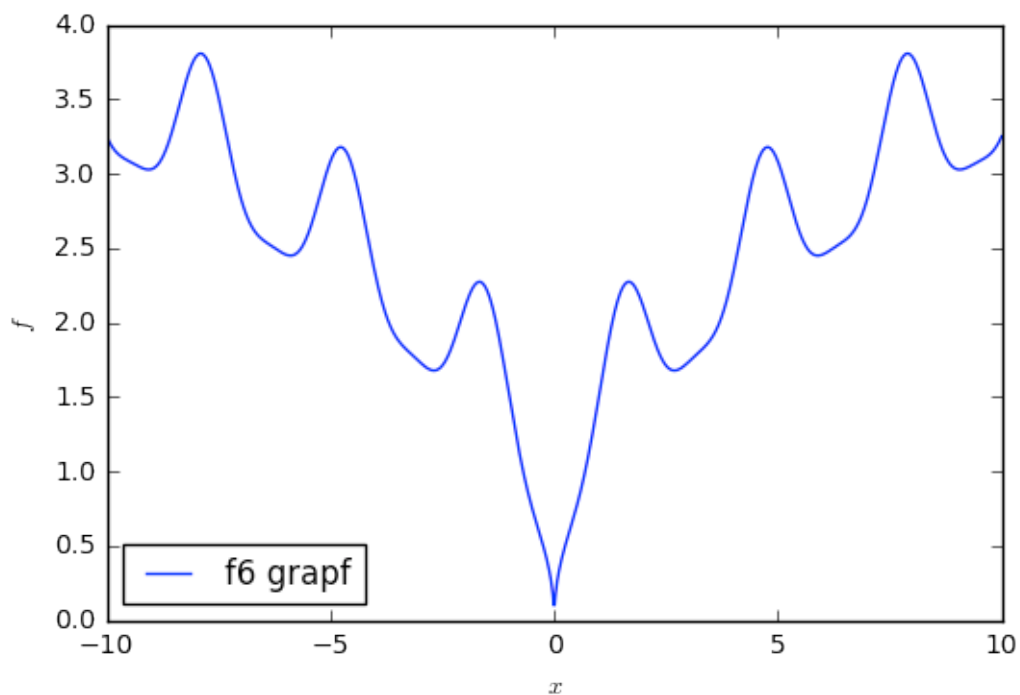


2. Формулювання та графік індивідуальної функції

Задаємо функцію f6:

```
def f6(x):  
    return np.sin(x)**4 + np.sqrt(np.abs(x))
```

Будуємо графік функції:

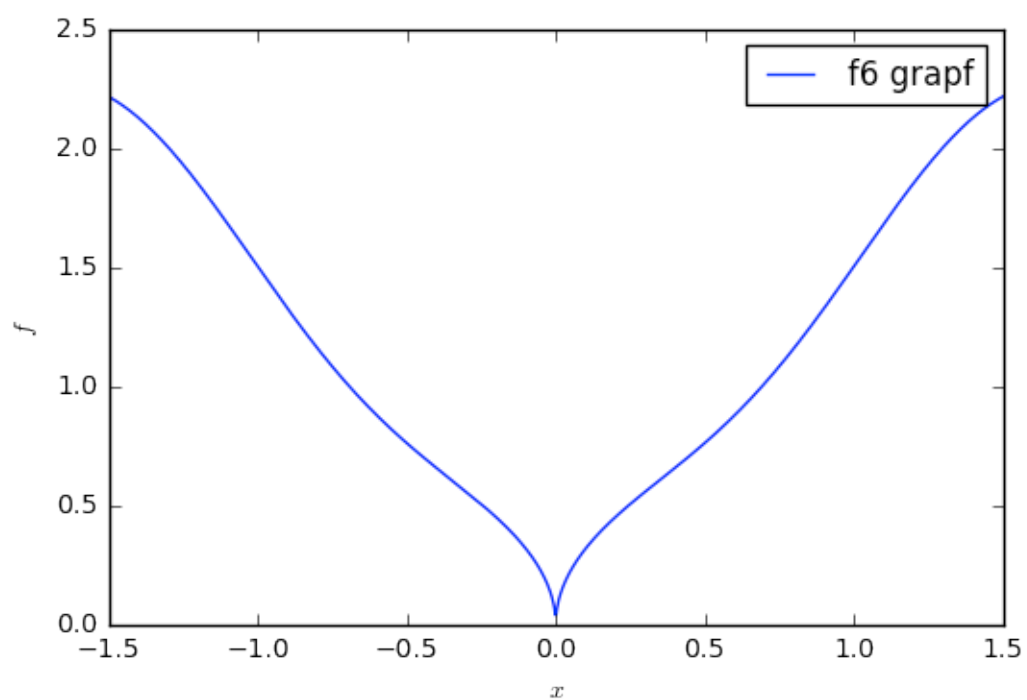
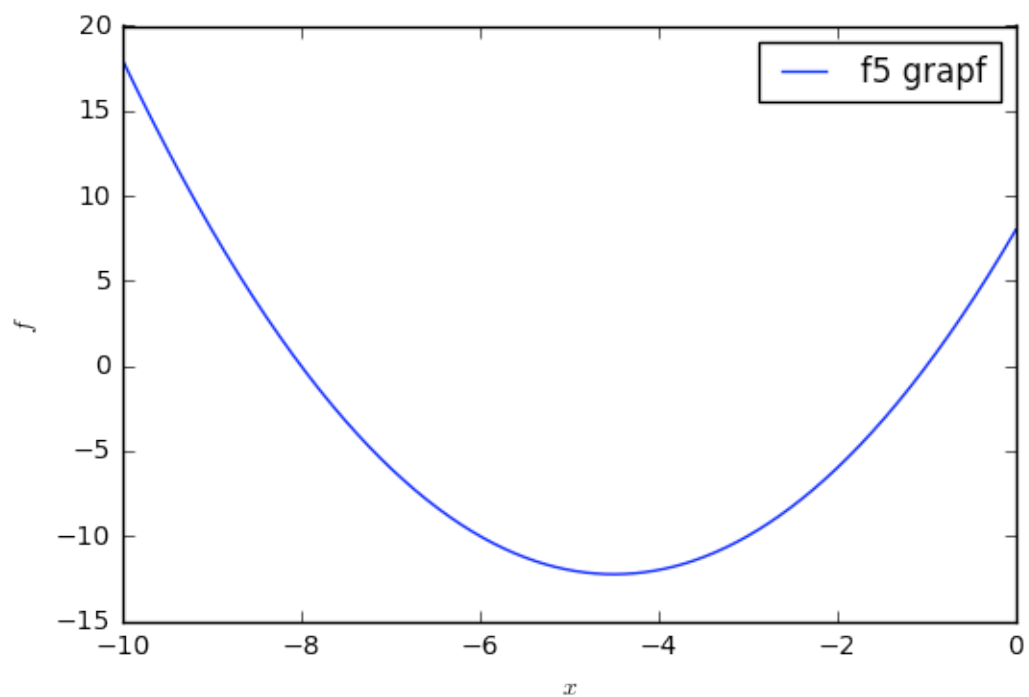


3. Визначення інтервалів пошуку за допомогою графіків функцій.

Мінімум квадратичної функції потрапляє в інтервал $[-10, 0]$, що видно за графіком функції.

Глобальний мінімум індивідуальної функції потрапляє в інтервал $[-1.5, 1.5]$.

Побудуємо графіки функцій на вказаних
інтервалах:



3. Складемо функцію пошуку мінімуму методом золотого перетину:

```
gr = (np.sqrt(5) + 1) / 2
```

```
'''змінна, що необхідна для алгоритму пошуку мінімуму  
методом золотого перетину'''
```

```
def gss(f, a, b, tol=1e-3):  
    '''
```

```
golden section search
```

```
знаходження мінімуму функції f на проміжку [a,b]
```

```
f повинна бути унімодальною на проміжку [a,b]
```

```
'''
```

```
c = b - (b - a) / gr
```

```
d = a + (b - a) / gr
```

```
iter = 0
```

```
while abs(a - b) > tol:
```

```
    print
```

```
    '{:04d}\t{:020.18f}\t{:020.18f}\t{:020.18f}'.format(i  
ter, a, b, abs(a-b))
```

```
    iter += 1
```

```
    if f(c) < f(d):
```

```
        b = d
```

```
    else:
```

```
        a = c
```

```
# пересчет c и d производится, чтобы не потерять  
точность, что в итоге приводит к бесконечному циклу
```

```
c = b - (b - a) / gr
```

```
d = a + (b - a) / gr
```

```
print
```

```
    '{:04d}\t{:020.18f}\t{:020.18f}\t{:020.18f}'.format(i  
ter, a, b, abs(a-b))
```

```
    return (b + a) / 2
```

Результати роботи програми наведено в таблиці:

| Iteration | Квадратична функція | | | Індивідуальна функція | | |
|-----------|---------------------|--------------|------------|-----------------------|------------|------------|
| | a | b | len | a | b | len |
| 0 | -10 | 0 | 10 | -1.5 | 1.5 | 3 |
| 1 | -6.1803399 | 0 | 6.18033989 | -0.354102 | 1.5 | 1.85410197 |
| 2 | -6.1803399 | -2.360679775 | 3.81966011 | -0.354102 | 0.79179607 | 1.14589803 |
| 3 | -6.1803399 | -3.819660113 | 2.36067977 | -0.354102 | 0.35410197 | 0.70820393 |
| 4 | -5.2786405 | -3.819660113 | 1.45898034 | -0.0835921 | 0.35410197 | 0.4376941 |
| 5 | -4.7213595 | -3.819660113 | 0.90169944 | -0.0835921 | 0.1869177 | 0.27050983 |
| 6 | -4.7213595 | -4.16407865 | 0.5572809 | -0.0835921 | 0.08359214 | 0.16718427 |
| 7 | -4.7213595 | -4.376941013 | 0.34441854 | -0.0835921 | 0.01973343 | 0.10332556 |
| 8 | -4.5898034 | -4.376941013 | 0.21286236 | -0.0441253 | 0.01973343 | 0.06385871 |
| 9 | -4.5898034 | -4.4582472 | 0.13155617 | -0.0197334 | 0.01973343 | 0.03946685 |
| 10 | -4.5395534 | -4.4582472 | 0.08130619 | -0.0197334 | 0.00465843 | 0.02439186 |
| 11 | -4.5395534 | -4.4893034 | 0.05024999 | -0.0104166 | 0.00465843 | 0.015075 |
| 12 | -4.5203596 | -4.4893034 | 0.0310562 | -0.0046584 | 0.00465843 | 0.00931686 |
| 13 | -4.5084972 | -4.4893034 | 0.01919379 | -0.0046584 | 0.00109971 | 0.00575814 |
| 14 | -4.5084972 | -4.496634775 | 0.01186241 | -0.002459 | 0.00109971 | 0.00355872 |
| 15 | -4.5039661 | -4.496634775 | 0.00733137 | -0.0010997 | 0.00109971 | 0.00219941 |
| 16 | -4.5011658 | -4.496634775 | 0.00453104 | -0.0010997 | 0.00025961 | 0.00135931 |
| 17 | -4.5011658 | -4.498365477 | 0.00280034 | -0.0005805 | 0.00025961 | 0.0008401 |
| 18 | -4.5011658 | -4.49943511 | 0.0017307 | | | |
| 19 | -4.5005047 | -4.49943511 | 0.00106963 | | | |
| 20 | -4.5005047 | -4.499843674 | 0.00066107 | | | |
| Res | x_min | y_min | | x_min | y_min | |
| | 4.50 | -12.25 | | 0.00 | 0.01 | |