НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Курсова робота

із дисципліни «Дослідження операцій»

на тему: Метод Марквардта

Студента групи КМ-83 Керівник:

Меркулова І. Д. викладач Норкін Б.

Кількість балів:

Оцінка:

Київ — 2021

**ЗМІСТ**

ВСТУП

ОСНОВНА ЧАСТИНА

Постановка задачі

Теоретична частина

Практична частина

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

ОСНОВНА ЧАСТИНА

Постановка задачі

Дослідити збіжність методу Марквардта при мінімізації функції Розенброка в залежності від:

1. Величини кроку h при обчисленні першої та другої похідних.
2. Схеми обчислення першої та другої похідних.
3. Вигляду критерію закінчення. .
4. Значення параметрів методу.
5. Модифікації методу для задач типу *F = ∑ fi2(x)*

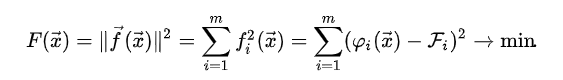
Використати метод штрафних функцій (метод внутрішньої точки) для умовної оптимізації в залежності від:

1. Розташування локального мінімума (всередині/поза допустимою областю).
2. Виду допустимої області (випукла/невипукла).
3. Розташування точки мінімума поза допустимої області.

Теоретична частина

Метод Марквардта

Метод Марквардта використовується для вирішення задач найменших квадратів. Метод є комбінацією методу Ньютона та методу градієнтного спуску. Рух в напрямку антіградіента з початкової точки пошуку, розташованої на значній відстані від точки мінімуму та напрямки ефективного пошуку в околі точки мінімуму, які визначаються за методом Ньютона.



*вигляд задачі найменших квадратів*

Суть метода полягає у тому, що ми на кожній ітерації шукаємо напрямок пошуку, який визначається формулою:

де а – це деякий параметр, який спочатку береться досить великим.

Е – це одинична матриця однакової розмірності з матрицею Гесса Hf(x)

Та якщо виконується умова f(x + s) < f(x), то перевіряється критерій закінчення.

Якщо критерій закінчення не виконується, то береться нова точка y та нове а, де а = a \* b, де b є (0,1). Та алгоритм працює знову, поки не виконається критерій закінчення, тобто не знайдеться мінімум функції.

**Переваги** методу полягають у високій швидкості збіжності в околі точки мінімуму, відносній простоті самого методу, цільова функція спадає на кожній ітерації.

**Недоліки** полягають у необхідності підрахунку матриці Гессе та вирішення системи лінійних рівнянь на кожній ітерації.

Алгоритм

1. Задати початкову точку x0, параметр a і b. Параметр b є (0,1). Також визначити параметр збіжності .
2. Підрахувати градієнт .
3. Підрахувати матрицю Гессе .
4. Далі вирішити систему рівнянь та знайти напрямок пошуку s.
5. Якщо f() f(), то покласти a = a / b та перейти до 4 пункту.
6. Покласти x = та a = a \* b.
7. Якщо не виконується критерій закінчення, то перейти до 2 пункту.
8. Нова точка x є шуканим мінімумом.

У даній роботі метод Марквардта буде використовуватись для пошуку мінімуму функції Розенброка. Буде проаналізовано як параметри, схема обчислення похідної, величина кроку та критерій закінчення впливає на швидкість збіжності методу та точність результатів.

Ресурси інформації

<https://studfile.net/preview/4243517/page:8/>

<https://habr.com/ru/post/470181/>

<https://ru.wikipedia.org/wiki/Алгоритм_Левенберга_—_Марквардта>

<https://ru.wikipedia.org/wiki/Функция_Розенброка>

<https://matica.org.ua/metodichki-i-knigi-po-matematike/metody-mnogomernoi-bezuslovnoi-minimizatcii-v-p-severin/16-metod-markvardta>

<https://math.semestr.ru/optim/marquardt.php>

Метод штрафних функцій (метод внутрішньої точки)

Метод внутрішньої точки - це метод дозволяє вирішувати завдання опуклою оптимізації з умовами, заданими у вигляді нерівності, зводячи вихідну задачу до задачі опуклою оптимізації. Тобто шукати мінімум функції з обмеженнями.

Початкова функція перетворюється на нову виду:

– дана функція називається штрафом

R – набір штрафних параметрів.

Види штрафів:

Для нерівності:

Для рівності:

Для цієї, нової, функції знаходиться мінімум за допомогою методу безумовної оптимізації.

Потім перевіряється умова:

Якщо умова виконується, то точку мінімуму знайдено, але якщо – ні, то все працює заново, але з новим r, який дорівнює:

Тобто з кожною, новою, ітерацією r набуває менших значень.

З явних **недоліків** даного методу можна виділити: необхідність пошуку допустимої точки, щоб використати її у якості початкової, при малих значеннях r ускладнюється рішення задач безумовної оптимізації.

Алгоритм

1. Визначення початкової точки.
2. Побудова нової функції
3. Знаходження мінімуму нової функції за допомогою методу безумовної оптимізації.
4. Якщо умова не виконується, то береться нове та робота алгоритму починається знову.
5. Отримано шукану точку мінімуму функції з обмеженнями.

Ресурси інформації

<https://uk.wikipedia.org/wiki/Метод_внутрішньої_точки#:~:text=Методи%20внутрішніх%20точок%20(їх%20також,лінійної%20та%20нелінійної%20опуклої%20оптимізації>.

<https://mipt.ru/aspirantura/upload/8bb/c_6d8i66.pdf>

<https://ru.wikipedia.org/wiki/Метод_внутренней_точки#:~:text=Метод%20внутренней%20точки%20—%20это%20метод,сопромату%2C%20математическому%20моделированию%20и%20эконометрике>.

<http://sopromat.org.ua/sopromat_files/Оптимізація%20конструкцій/Lec_OPT_9.pdf>

<http://www.machinelearning.ru/wiki/images/8/81/MOMO12_ipm.pdf>

<https://people.inf.ethz.ch/fukudak/lect/opt2011/aopt11note4.pdf>

<https://www.maths.ed.ac.uk/hall/NATCOR_2014/IPMforLP.pdf>

<https://en.wikipedia.org/wiki/Interior-point_method>

<https://er.nau.edu.ua/bitstream/NAU/35938/8/Лекція%208.pdf>

Практична частина

Метод Марквардта

Спочатку перевіримо правильність роботи методу на функції Розенброка, бо мінімум цієї функції відомий и знаходиться в точці (1,1). Візьмемо такі початкові дані:

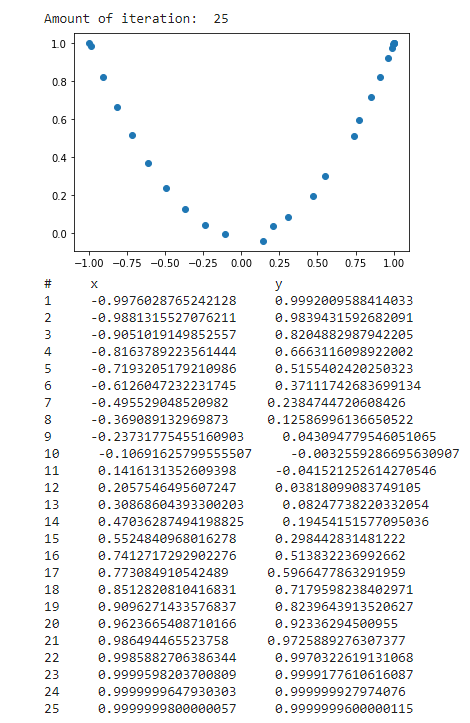
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.1

= 0.0000001

h = 0.00001



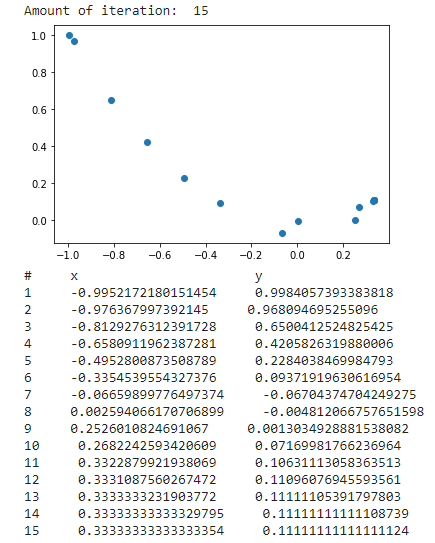
Отже, як видно з графіку, наша точка мінімуму збігається до точки (1,1).

Також точність збігається з обраним .

Програма працює правильно. На основі цього можна робити аналіз з різними вхідними даними.

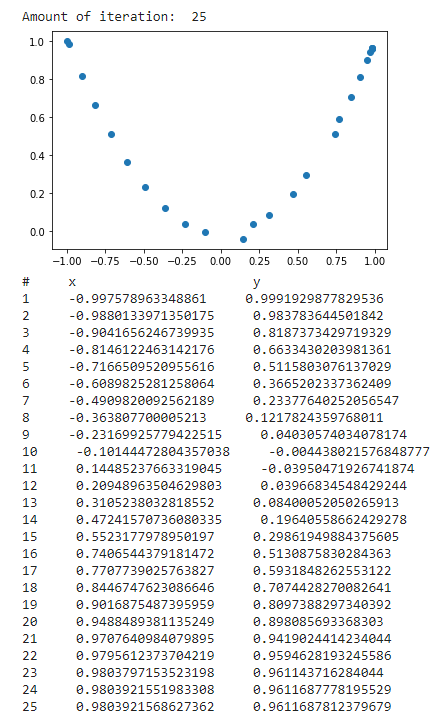
Проаналізуємо результати при різних значеннях h при підрахунку першої та другої похідних.

Візьмемо h рівну 0.1. Інші параметри залишаються незміненими.

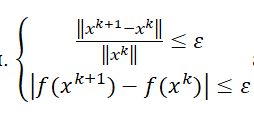


Як можна побачити значення h було занадто великим, тому результат вийшов зовсім не точний.

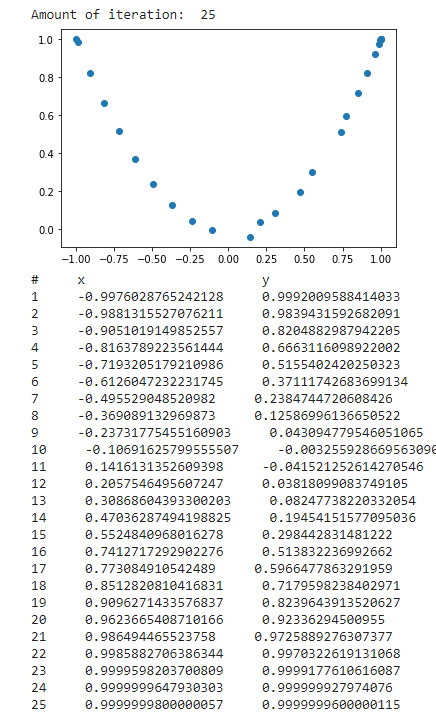
Тому в цей раз зменшимо h до 0.01. Інші параметри залишаються незміненими.



У цей раз результати краще, але все одно не досить точні, тому зменшимо h ще до 0.00001. Інші параметри залишаються незміненими.

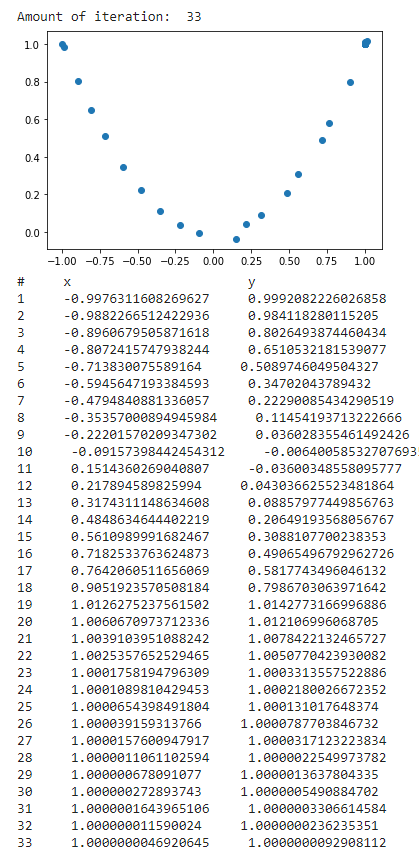


Якщо не уточняється вибір критерію закінчення, то це значить, що обраний критерій написаний вище.



Отже чим менше ми беремо h тим точніше виходить результат. Що не дивно враховуючи спосіб підрахунку похідної в точці через ліміти. Але не все так просто. Тому візьмемо дуже мале h.

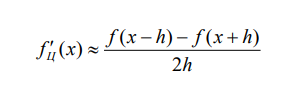
Тепер візьмемо дуже мале h рівне 0.00000000000001. Інші параметри залишаються незміненими.



Як можна побачити при дуже малих значеннях точність стає меншою, тому треба знаходити баланс при виборі h. Бо дуже велике значення або дуже мале призведе до неточності у підрахунку похідних і як наслідок у всій задачі.

Також можна побачити, що функція збігається швидше, коли h взято збалансованим.

Для підрахунку похідних було використано спосіб через ліміти, тобто:



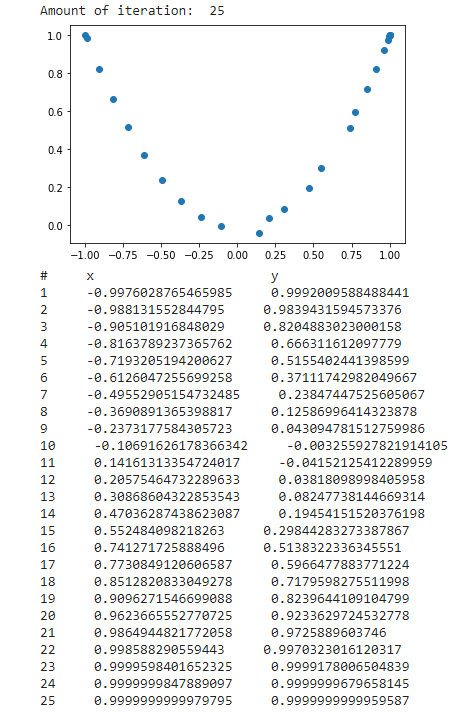
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.1

= 0.0000001

h = 0.0000001



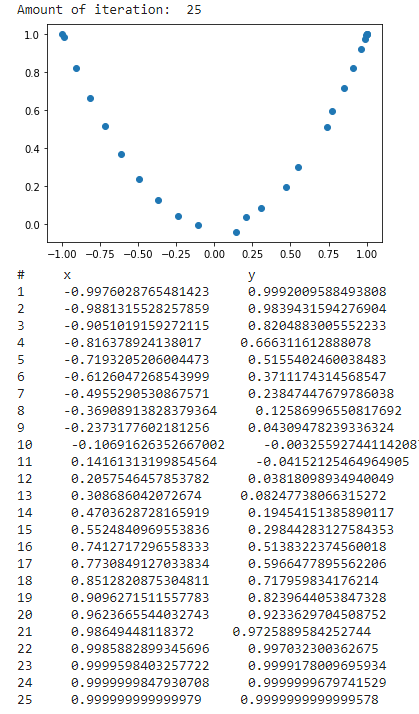
Як можна побачити, розрахунки досить точні. А тепер порахуємо похідну через інструменти пітону, тобто візьмемо похідну та потім підставимо нашу точку у отриману функцію.

X = (-1,1) (точка початку пошуку)

a = 1000

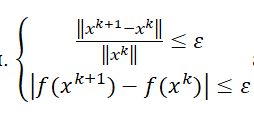
b = 0.1

= 0.0000001



У цьому випадку результати ще точніше, але вибір різних схем ніяк не впливає на кіл-ть ітерацій, тобто на те як швидко збігається метод.

Тепер проаналізуємо швидкість збігання при різних критеріях закінчення.



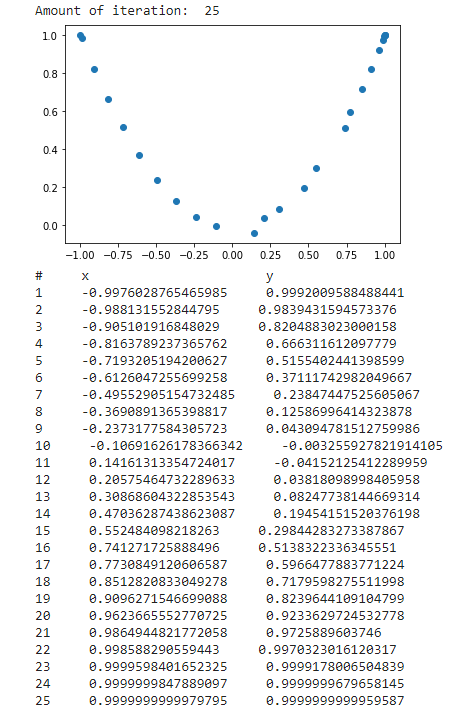
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.1

= 0.0000001

h = 0.0000001





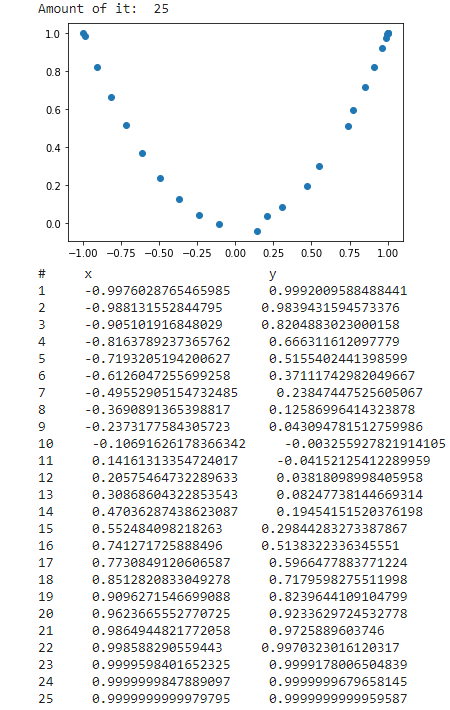
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.1

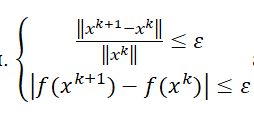
= 0.0000001

h = 0.0000001



Отже, вибір різних критеріїв закінчення не вплинув ні на точність підрахунків, ні на швидкість збіжності.

Також проведено випробування з іншими параметрами.



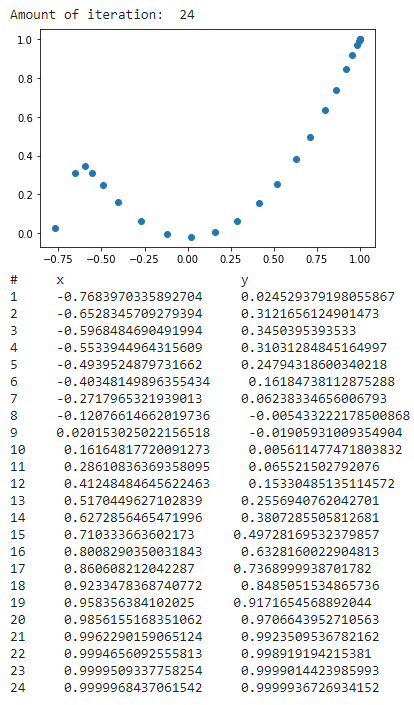
X = (-1,-1) (точка початку пошуку)

a = 100

b = 0.7

= 0.0001

h = 0.0000001





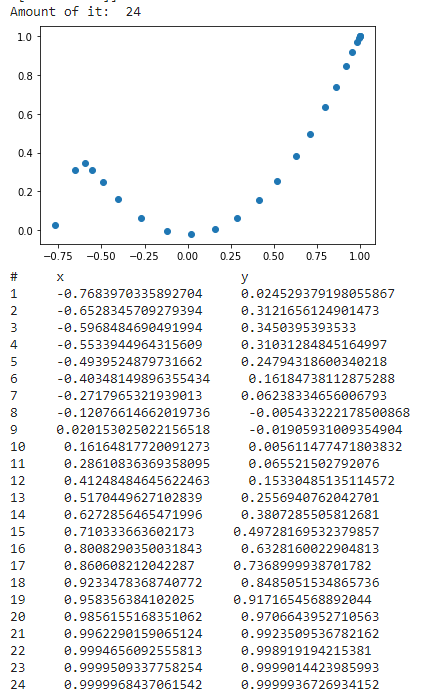
X = (-1,-1) (точка початку пошуку)

a = 100

b = 0.7

= 0.0001

h = 0.0000001



Але, як видно, зміна критерію закінчення ні на що не впливає.

Далі буде аналіз в залежності від параметрів a,b,,(x,y).

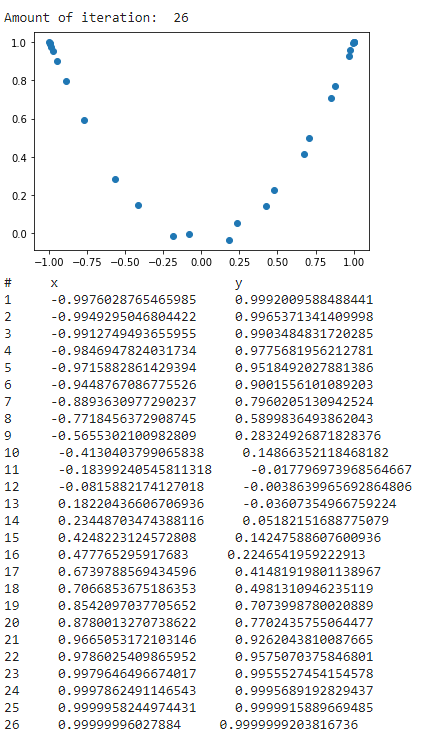
X = (-1,1) (точка початку пошуку)

a = 1000

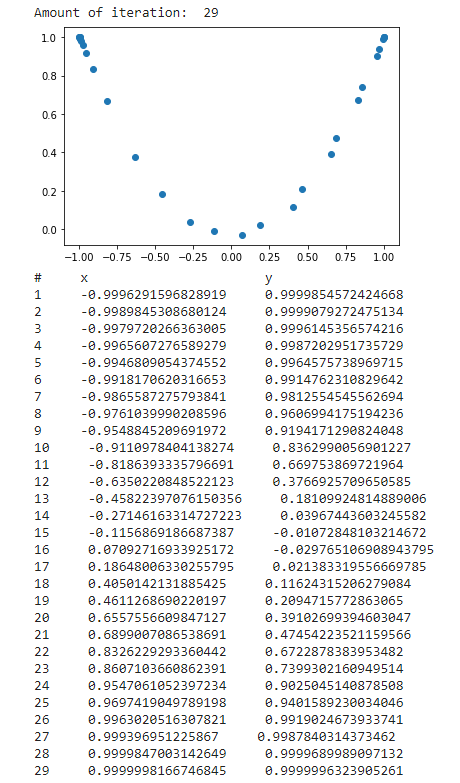
b = 0.5

= 0.0001

h = 0.0000001

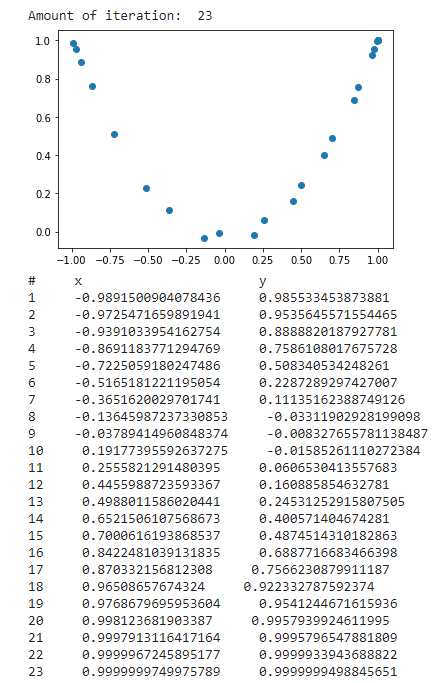


Збільшимо a при тих самих параметрах.



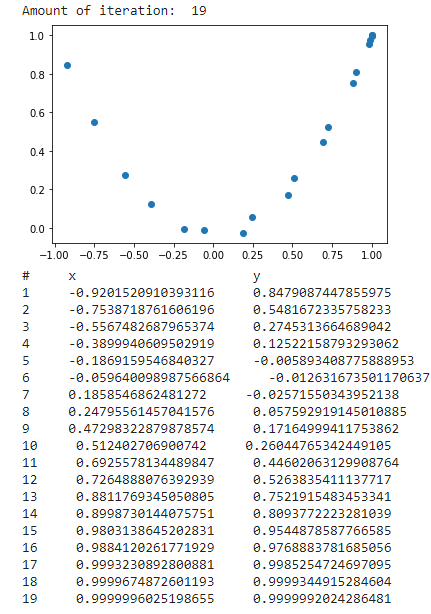
Збільшення a знизило швидкість збіжності та точність результатів.

Тепер зменшимо a до 100



При зменшені а до 100, швидкість збіжності стала більшою(всього 23 ітерації)

Зменшимо ще, до 10.



Як можна побачити, метод став збігатись ще швидше, та точність результатів залишилась задовільною.(тобто все в рамках похибки).

Подальше зменшення параметру a нічого не змінить, кіл-ть ітерацій залишиться такою самою. Просто при зовсім малих параметрах а, алгоритм буде його збільшувати тому що нове значення функції буде більше старого.

Якщо,

Тепер проаналізуємо при зміні параметру b із сталим та динамічним параметром a.

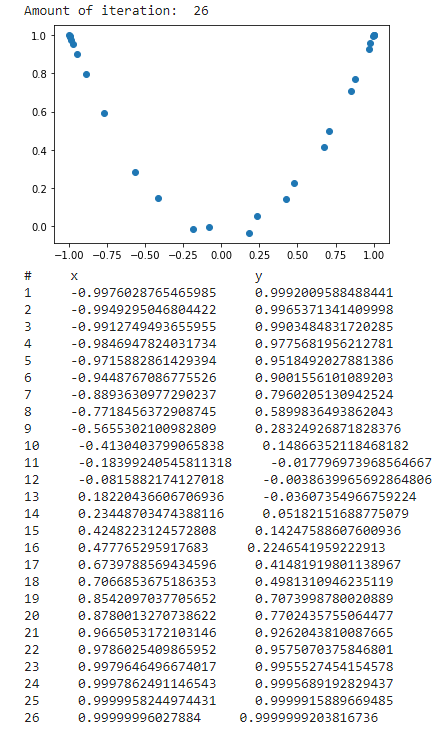
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.5

= 0.0001

h = 0.0000001



Зменшимо b.

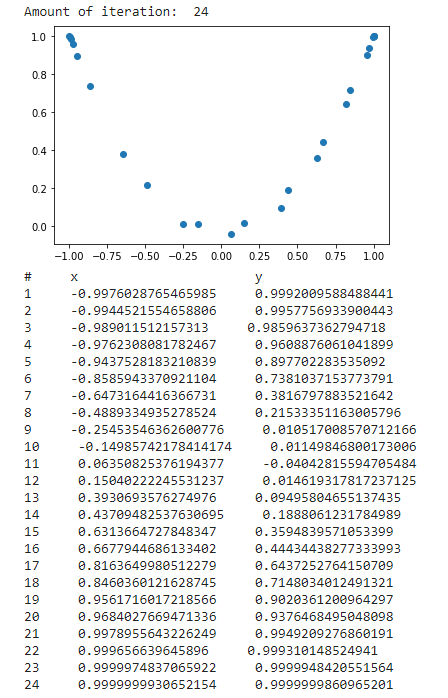
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.4

= 0.0001

h = 0.0000001



Кіл-ть ітерацій зменшилась.

Ще зменшим b.

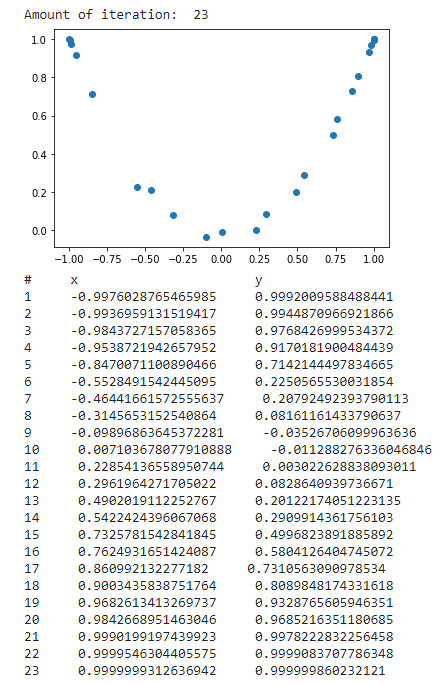
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.3

= 0.0001

h = 0.0000001



Ще зменшимо b.

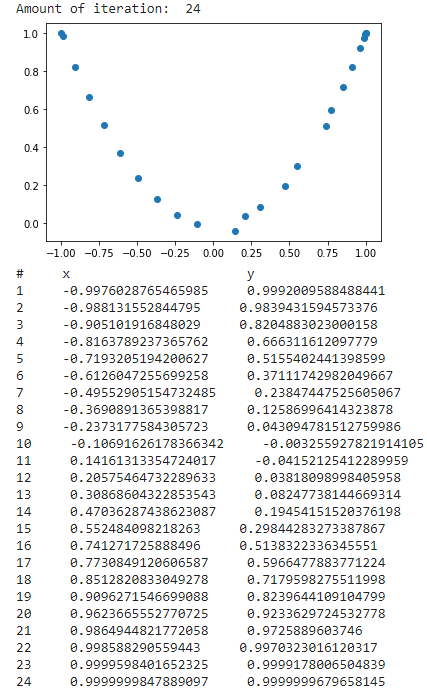
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.1

= 0.0001

h = 0.0000001



Тобто до якогось моменту швидкість збіжності збільшувалась, але потім стала зменшуватись. Тепер будемо збільшувати b до 1.

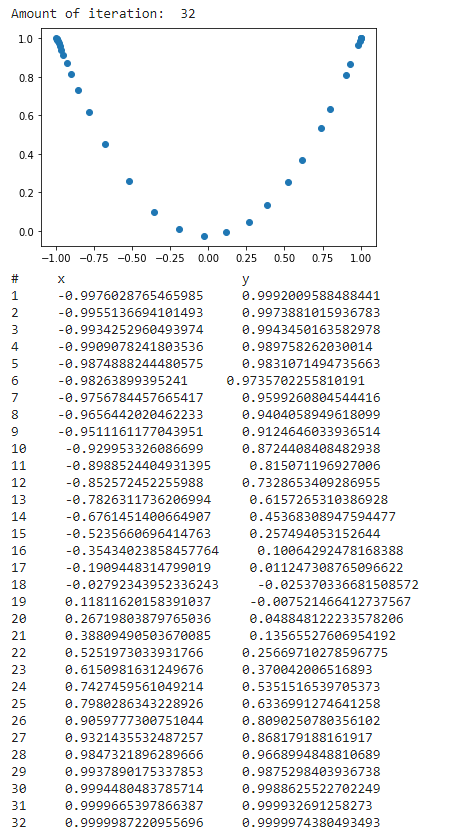
X = (-1,1) (точка початку пошуку)

a = 1000

b = 0.7

= 0.0001

h = 0.0000001



Отже при збільшені b кіл-ть ітерацій тільки збільшується.

В цілому нічого дивного в цьому нема, бо параметр b відповідає за зміну а. Тобто ця зміна не повинна бути сильно велика та сильно маленька, щоб a змінювалась збалансовано.

Також при зміні параметру a на інший, параметр b працює так саме. Тобто швидше метод збігається, коли b збалансовано.

Найшвидше вийшло при таких параметрах.

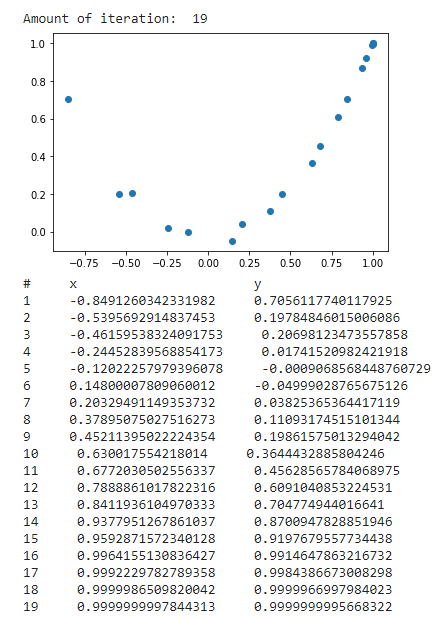
X = (-1,1) (точка початку пошуку)

a = 1

b = 0.2

= 0.0001

h = 0.0000001



Якщо змінювати точку на -10 10, тобто вона буде знаходитись дальше. То при збільшені a буде зменшуватись кіл-ть ітерацій, наприклад:

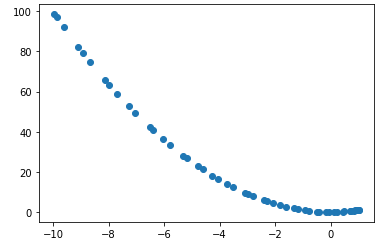
X = (-10,10) (точка початку пошуку)

a = 1

b = 0.2

= 0.0001

h = 0.0000001



Вийшло більше 50 ітерацій і програма навіть не встигла дорахувати.

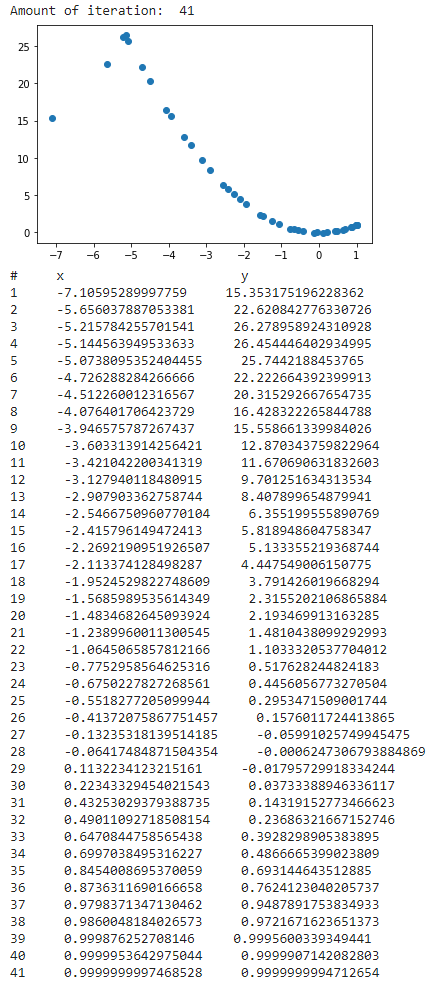
X = (-10,10) (точка початку пошуку)

a = 1000

b = 0.2

= 0.0001

h = 0.0000001



Тобто при більших відстанях краще брати більшу a. Але поведінка параметру b не змінилась, тобто найкращих результатів можна досягти приблизно,

коли b =0.2.

В кінці подивимось на швидкість збіжності при різних значеннях похибки.

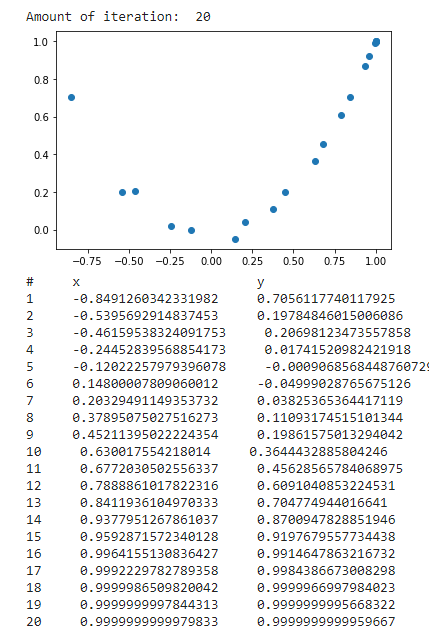
X = (-1,1) (точка початку пошуку)

a = 1

b = 0.2

= 0.000001

h = 0.0000001



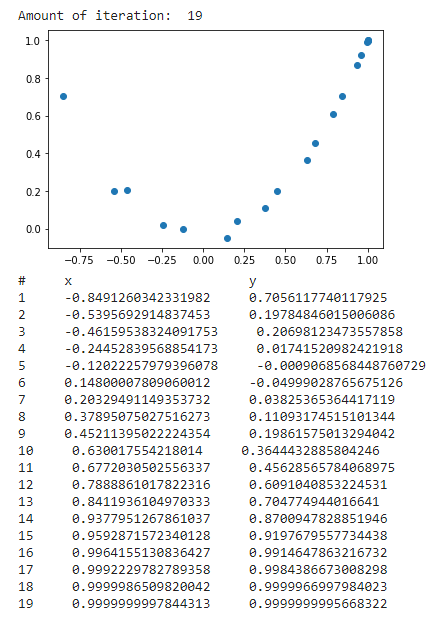
X = (-1,1) (точка початку пошуку)

a = 1

b = 0.2

= 0.001

h = 0.0000001



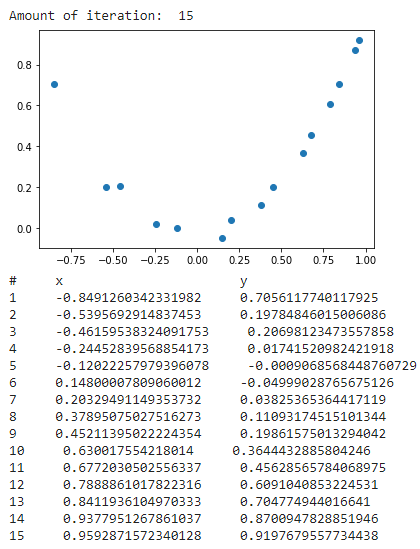
X = (-1,1) (точка початку пошуку)

a = 1

b = 0.2

= 0.1

h = 0.0000001

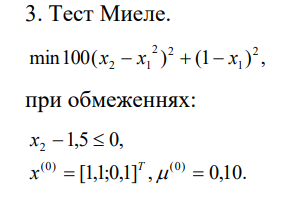


Як можна побачити, нічого дивного, тобто при збільшені похибки – збільшується швидкість збіжності, але зменшується точність.

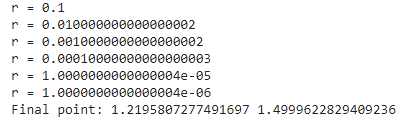
Отже, з даного аналізу видно деякі закономірності. Наприклад, при великих відстанях між точками, краще брати велике a і тд.

Метод внутрішньої точки

Умова



Результат



ВИСНОВКИ

Було проведено роботу по розробці програмного забезпечення для роботи методу Марквардта та методу внутрішньої точки. Та аналіз роботи при різних вхідних даних.

Аналіз методу Марквардта.

Різна величина кроку h впливає на точність результатів. Але треба брати не дуже великий і не дуже маленький h. Тобто шукати баланс, бо тільки таким чином можна досягти великої точності результатів.

В залежності від вибору схеми обчислення першої та другої похідної змінюється тільки точність результатів. Але якщо підібрати h то можна досягти однакової точності результатів.

Вигляд критерію закінчення взагалі не впливає на точність чи швидкість збіжності методу.

При малій відстані від початкової да точки мінімуму краще брати малий a для кращої швидкості збіжності. А при великій відстані – велику a. Параметр b в будь-якій ситуації давав найкращий результат при значенні близькому до 0.2.

Вибір різної похибки впливає на швидкість збіжності методу та на точність обчислень. Тобто чим вище похибка тим швидше збігається метод, але точність результатів нижче.

Метод внутрішньої точки працює правильно згідно с отриманими результатами.

ДОДАТКИ

Програмна реалізація - <https://colab.research.google.com/drive/1A3gXE2EpmuQ56rtvl6RJlg7tE_vFPrHE?usp=sharing>

СПИСОК ЛІТЕРАТУРИ

<https://uk.wikipedia.org/wiki/Метод_внутрішньої_точки#:~:text=Методи%20внутрішніх%20точок%20(їх%20також,лінійної%20та%20нелінійної%20опуклої%20оптимізації>.

<https://mipt.ru/aspirantura/upload/8bb/c_6d8i66.pdf>

<https://ru.wikipedia.org/wiki/Метод_внутренней_точки#:~:text=Метод%20внутренней%20точки%20—%20это%20метод,сопромату%2C%20математическому%20моделированию%20и%20эконометрике>.

<http://sopromat.org.ua/sopromat_files/Оптимізація%20конструкцій/Lec_OPT_9.pdf>

<http://www.machinelearning.ru/wiki/images/8/81/MOMO12_ipm.pdf>

<https://people.inf.ethz.ch/fukudak/lect/opt2011/aopt11note4.pdf>

<https://www.maths.ed.ac.uk/hall/NATCOR_2014/IPMforLP.pdf>

<https://en.wikipedia.org/wiki/Interior-point_method>

<https://er.nau.edu.ua/bitstream/NAU/35938/8/Лекція%208.pdf>

<https://studfile.net/preview/4243517/page:8/>

<https://habr.com/ru/post/470181/>

<https://ru.wikipedia.org/wiki/Алгоритм_Левенберга_—_Марквардта>

<https://ru.wikipedia.org/wiki/Функция_Розенброка>

<https://matica.org.ua/metodichki-i-knigi-po-matematike/metody-mnogomernoi-bezuslovnoi-minimizatcii-v-p-severin/16-metod-markvardta>

<https://math.semestr.ru/optim/marquardt.php>