Міністерство освіти та науки України
Національний Технічний Університет України
«Київський Політехнічний Інститут»

Фізико – технічний інститут

Лабораторна робота №1

«Метод спряжених напрямків»

Виконали:
Студенти 4 курсу
Групи ФІ-51
Макаренко Сергій
Скірдін Євгеній
Сітко Дарина
Сімакова Катерина

Перевірив:
Данилов В.Я

Київ, 2018 р.

Теоретичні відомості

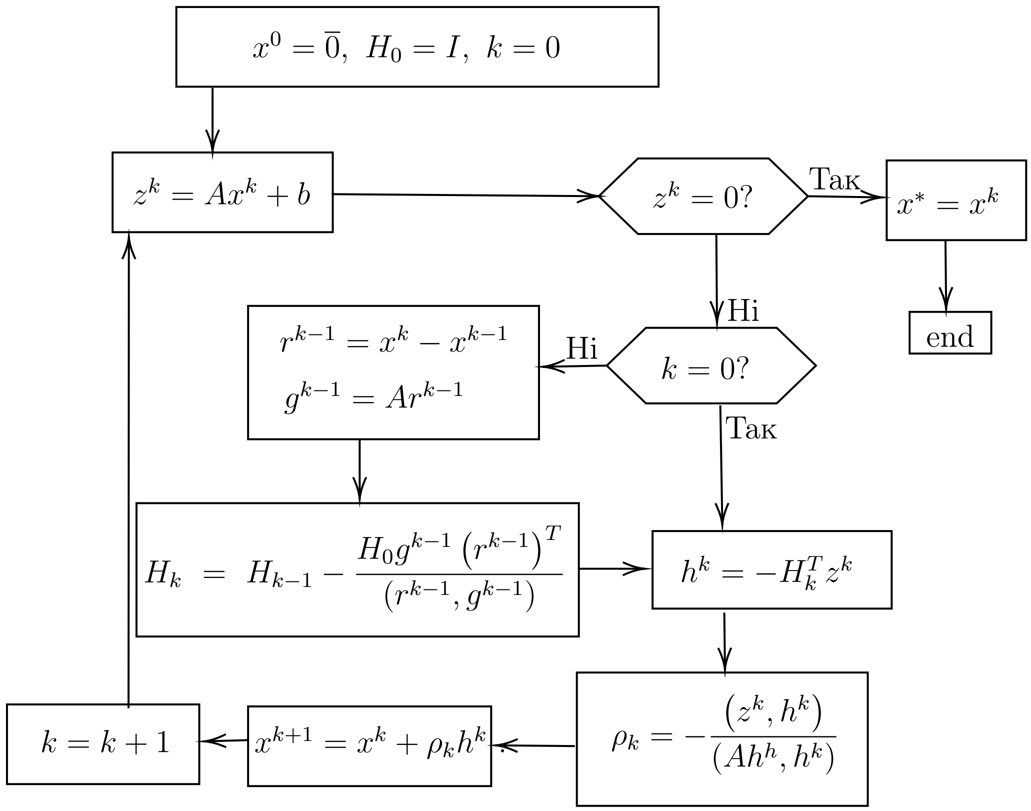
Нехай f(x) – випукла диференційована в усьому просторі функція і треба знайти її точку мінімуму.

Тобто знайти $\underset{x \in R^n}{\operatorname{argmin}} \; f_0(x)$ для заданої неперервно диференційованої функції $f_0 : R^n \to R^1$

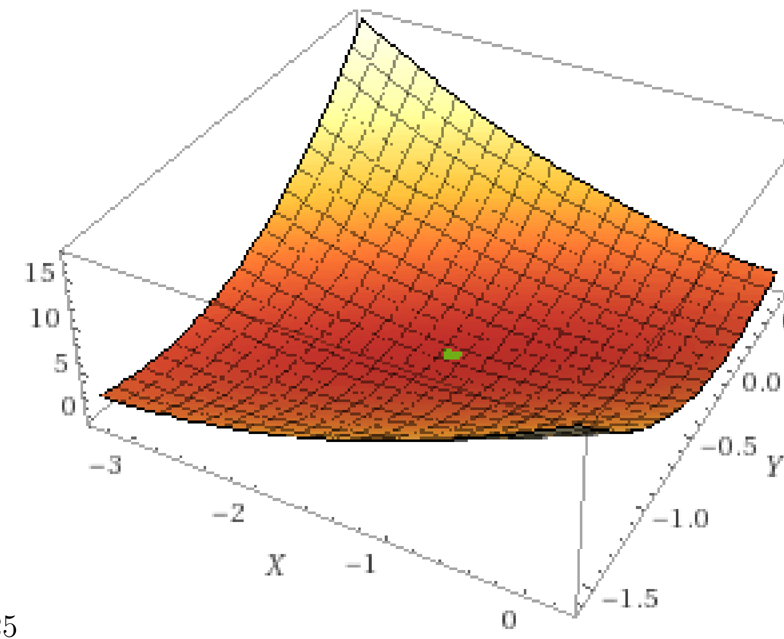
Алгоритм:

1. Вибрати довільне початкове наближення $x^0 \in R^n$, довільну симетричну строго додатно визначену матрицю H_0 , покласти $k = 0$.
2. Обчислити $Ax^k + b$ і покласти $z^k = Ax^k + b$. Якщо $z^k = 0$, то покласти $x^* = x^k$ і завершити обчислення, інакше перейти на крок 3.
3. Якщо $k = 0$, то перейти на крок 6, інакше перейти на крок 4.
4. Обчислити вектори: $r^{k-1} = x^k - x^{k-1}$, $g^{k-1} = Ar^{k-1}$.
5. Обчислити матрицю $H_k = H_{k-1} - \dfrac{H_0 g^{k-1} \left(r^{k-1}\right)^T}{\left(r^{k-1}, g^{k-1}\right)}$
6. Обчислити вектор руху h^k до наближення x^{k+1} $h^k = -H_k^T z^k$
7. Обчислити кроковий множник $\rho_k = -\dfrac{\left(z^k, h^k\right)}{\left(Ah^h, h^k\right)}$.
8. Обчислити наближення $x^{k+1} = x^k + \rho_k h^k$.
9. Покласти $k = k + 1$ і перейти на крок 2.

Блок - схема алгоритму:



```
import numpy as np
x = np.array([[0],[0]], float)
H = np.array([[1,0],[0,1]], float)
A = np.array([[2,-2],[-2,12]], float)
b = np.array([[1],[-1]], float)

z= np.dot(A, x)+b

h=-np.dot(np.transpose(H),z)

rho = -np.dot(np.transpose(z),h)/np.dot(np.transpose(np.dot(A,h)),h)

x_old=x
x=x+rho*h

z= np.dot(A, x)+b


H0=H

while (np.linalg.norm(z)>0.0001):
        r=x-x_old
        g=np.dot(A,r)

        H=H-np.dot(np.dot(H0,g),np.transpose(r))/np.dot(np.transpose

        h=-np.dot(np.transpose(H),z)

        rho = -np.dot(np.transpose(z),h)/np.dot(np.transpose(np.dot(

        x_old=x
        x=x+rho*h

        z= np.dot(A, x)+b

print("rezult", x)
```



Результати: $x_1 = -0.5;\ x_2 = 0;\ f\left(\overline{x}\right) = -0.25$