Лабораторна робота № 1
Оптимізація функцій однієї змінної в MatLab

Мета роботи

Вивчення алгоритмів оптимізації одновимірних функцій

Завдання до роботи

1. Вивчити алгоритми пошуку екстремумів одновимірних функцій.
2. Розробити програмне забезпечення (ПО), яке виконує пошук мінімуму і максимуму функції з використанням методів Фібоначчі і золотого перетину.
3. Знайти чисельні значення мінімуму і максимуму функції методами Фібоначчі і золотого перетину на заданих інтервалах.
4. Порівняти ефективність методів Фібоначчі і золотого перетину.
5. Зробити висновки за результатами роботи.

Пояснення до роботи

Завдання одновимірної оптимізації являє собою просту математичну модель, в якій цільова функція залежить тільки від однієї змінної. Знаходження екстремуму функції $f(x)$ зводиться до пошуку деякого інтервалу [a, b] на дійсній осі x [1]. Вважається, що екстремум функції може відповідати будь-якій точці інтервалу з однаковою ймовірністю. В процесі пошуку відбувається поступове зменшення довжини інтервалу до деякого значення, яке визначається необхідною похибкою визначення положення екстремуму.

До математичного завданням одновимірної мінімізації призводять прикладні задачі оптимізації з однією керованою змінною. Необхідність мінімізації функцій однієї змінної також виникає при вирішенні більш складних завдань оптимізації [2].

Для вирішення завдання мінімізації функції $f(x)$ на відрізку [a, b] на практиці, як правило, застосовують наближені методи. Вони дозволяють знайти рішення цієї задачі з необхідною точністю в результаті визначення кінцевого числа значень функції $f(x)$ і її похідних в деяких точках інтервалу [a, b]. Методи, які використовують тільки значення функції $f(x)$ і не потребують обчислення її похідних, називаються **прямими методами оптимізації** (методами оптимізації нульового порядку) [2].

Важливою *перевагою* прямих методів є те, що від цільової функції не потрібно диференційованості і, більш того, вона може бути не задана в аналітичному вигляді. Для реалізації прямих методів необхідно, щоб значення функції $f(x)$ могли бути визначені в заданих точках.

Завдання оптимізації функції однієї змінної може бути сформульована таким чином. Будемо вважати, що унімодальна функція $f(x)$ має мінімум на інтервалі [a, b]. Потрібно приблизно визначити значення аргументу x^*, відповідно мінімуму (максимуму) функції, при цьому абсолютна похибка чисельного вирішення задачі не повинна перевищувати деякого значення ε (далі в поясненнях до роботи буде розглядатися тільки завдання пошуку мінімуму функції).

У більшості літературних джерел для позначення абсолютної похибки результату чисельного рішення задачі використовується вираз [2]

$$\Delta = |x^* - x_0| \tag{1.1}$$

де x^* - обчислене наближене значення величини; x_0 - точне значення величини. Абсолютна похибка виражається в одиницях вимірюваної величини і в загальному випадку може мати знак, тобто. формула (1.1) може використовуватися без модуля.

Відносна похибка результату чисельного рішення задачі визначається відповідно до виразу [3]

$$\Delta = \frac{|x^* - x_0|}{x^*} * 100\% \tag{1.2}$$

Формула (1.2) виражає значення відносної похибки у відсотках, при цьому значення δ також може бути виражено в частках [4, 5].

Обчислене наближене значення величини x * можна вважати рішенням завдання, якщо виконується умова

$$\Delta = \left| x^* - x_0 \right| \leq \varepsilon \tag{1.3}$$

де ε - задане позитивне число.

У більшості випадків для знаходження чисельного рішення задачі використовуються ітераційні методи, тобтоЗначення x* знаходиться шляхом багаторазового виконання деякої послідовності дій. Таким чином, умова (1.3) є умовою завершення ітераційного процесу. Завдання пошуку мінімуму одновимірної функції може бути зведена до задачі ітераційного зменшення довжини інтервалу [a, b], до значення, меншого або рівного ε .

Розглянемо методи оптимізації, що використовуються для мінімізації унімодальної цільової функції однієї зміною y = f (x).

Для знаходження мінімуму на інтервалі [a, b] може бути застосований **метод розподілу інтервалу навпіл [2, 3],** що використовує найбільш прості обчислення.

Для реалізації методу необхідно виконати наступні дії:

1. Встановити інтервал невизначеності [a, b], значення ε ;
2. Обчислити значення:

$$x_{cp} = \frac{a+b}{2}, \quad x_L = \frac{a+x_{cp}}{2}, \quad x_R = \frac{x_{cp}+b}{2};$$

3. Обчислити значення функції в точках x_L , x_{cp} , тобто f(x_L), f (x_{cp});
4. Якщо f (x_L) <f (x_{cp}) (рис. 1.1), поточний інтервал необхідно стиснути до інтервалу [a, x_{cp}], nj,nj: = x_{cp} , і перейти до дії 2, інакше , якщо f (x_L) ≥ f (x_{cp}), перейти до дії 5;
5. Обчислити значення функції f (x_R);
6. Якщо f (x_R) <f (x_{cp}) (рис. 1.2), поточний інтервал необхідно стиснути до інтервалу [x_{cp} , b], nj,nj A: = x_{cp} , і перейти до дії 2, інакше , якщо f (x_R) ≥ f (x_{cp}), перейти до кроку 7;
7. Якщо f (x_L) = f (x_R) (рис. 1.3), виконати операції a: = x_L , b: = x_R і перейти до кроку 2.

Процедура пошуку завершується при виконанні умови b - a < ε .

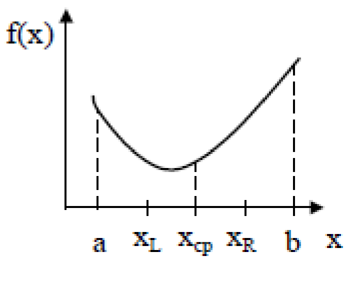

Рис. 1.1

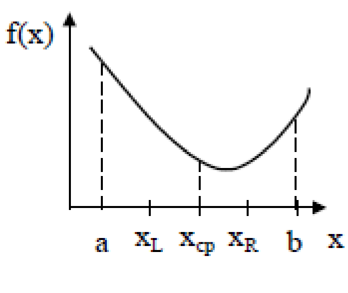

Рис. 1.2

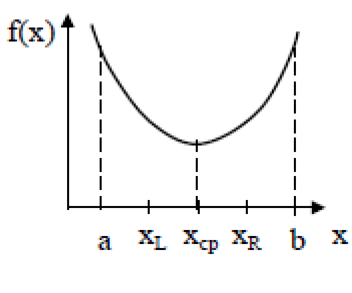

Рис. 1.3

В даний час існує декілька модифікацій методу розподілу відрізка навпіл. Зокрема, відомий метод дихотомії [3], що відрізняється зміненими правилами скорочення інтервалу [a, b] в ході пошуку. При використанні ділення інтервалу навпіл на кожній ітерації потрібно обчислення, по крайній мірі, двох значень цільової функції.

Найпростішими і широко використовуваними методами оптимізації функцій однієї змінної є *методи золотого перетину і Фібоначчі*. У порівнянні з методом розподілу відрізка навпіл дані методи вимагають меншої кількості обчислень значення цільової функції [2].

При використанні **методу золотого перетину** поточний інтервал невизначеності [a, b] розбивається на дві нерівні частини, співвідношення між якими залишається постійним і відповідає «золотому» перетину [2, 3]. Вважається, що точка робить «золотий» розтин інтервалу, якщо відношення довжини всього інтервалу до більшої частини дорівнює відношенню більшої частини до меншої [2]. Приклад «золотого» перетину інтервалу [a, b] показаний на рис. 1.4.

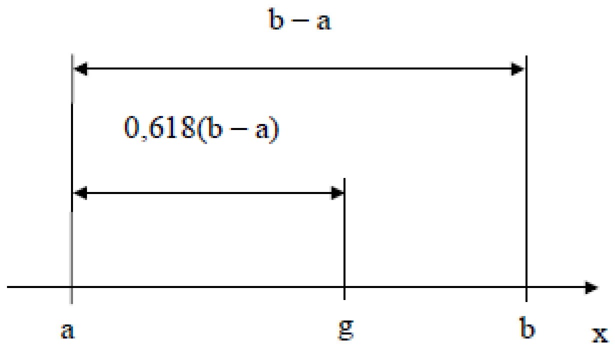

Рис. 1.4

На даному малюнку точка g робить розподіл інтервалу на дві частини при виконанні співвідношення

$$\frac{b-a}{g-a}=\frac{g-a}{b-g}=\frac{1}{z}, \text{ где } \ z=\frac{-1+\sqrt{5}}{2}\approx 0{,}618.$$

Вивід формули для знаходження параметра z в даній роботі не наводиться, тому що його можна знайти в багатьох джерелах, зокрема в роботах [2, 4].

На першій ітерації в межах поточного інтервалу невизначеності обчислюються значення (рис. 1.5)

$$x_L = z\cdot a + (1-z)\cdot b, \qquad (1.4)$$

$$x_R = (1-z)\cdot a + z\cdot b. \qquad (1.5)$$

Далі обчислюються значення функції в точках x_L, x_R, тобто $f(x_L)$, $f(x_R)$, і проводиться їх порівняння. Якщо виконується умова $f(x_L) \le f(x_R)$ (рис. 1.6), поточний інтервал необхідно стиснути до інтервалу [a, x_R], тобто B: = x_R, інакше, якщо $f(x_L) > f(x_R)$ (рис. 1.7), поточний інтервал необхідно стиснути до інтервалу [x_L, b], тобто a: = x_L. Після стиснення інтервалу необхідно перейти до наступної ітерації. Усі наступні ітерації починаються з перевірки виконання умови $b - a < \varepsilon$, яке є умовою завершення обчислень.

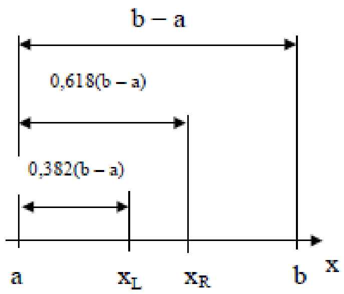

Рис. 1.5

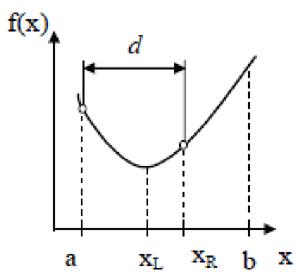

Рис. 1.6

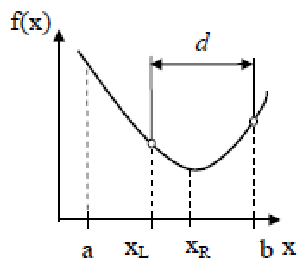

Рис. 1.7

Відзначимо, що після стиснення вихідного інтервалу, наприклад, до інтервалу [a, x_R] (рис. 1.6), буде виконуватися співвідношення

$$\frac{x_R - a}{x_L - a} = \frac{1}{z}.$$

Таким чином, значення x_L може бути використано в якості значення x_R на наступній ітерації. Тому після стиснення інтервалу відповідно до рис. 1.6 обчислення x_R за формулою (1.4) не потрібно, т. К. Воно може бути знайдено шляхом привласнення $x_R := x_L$. Аналогічно для випадку, показаного на рис. 1.7, буде виконуватися співвідношення

$$\frac{b - x_L}{b - x_R} = \frac{1}{z},$$

і значення x_L на наступній ітерації може бути знайдено шляхом привласнення $x_L := x_R$.

Схема алгоритму оптимізації функції за методом золотого перетину представлена на рис. 1.8.

Метод Фібоначчі реалізує алгоритм, що забезпечує максимальне скорочення інтервалу невизначеності при заданій кількості виконаних ітерацій. Основою алгоритму є числа Фібоначчі, які обчислюються за такою формулою:

$$F_0 = F_1 = 1,\ F_n = F_{n-1} + F_{n-2},$$

де n - індекс числа Фібоначчі, що представляє собою ціле число, $n \geq 2$.

Використання даної рекуррентної формули дає послідовність чисел $F_0 = 1$, $F_1 = 1$, $F_2 = 2$, $F_3 = 3$, $F_4 = 5$, $F_5 = 8$, $F_6 = 13$, $F_7 = 21$, $F_8 = 34$, $F_9 = 55$, $F_{10} = 89$ і т. д. Числа Фібоначчі пов'язані з співвідношенням, відповідним золотого перетину. Так, наприклад, при $n \geq 3$ буде виконуватися умова

$$F_{n-1} / F_n \approx z.$$

Чим більше значення n, тим ближче відношення F_{n-1} / F_n до числа z, що визначається золотим перетином.

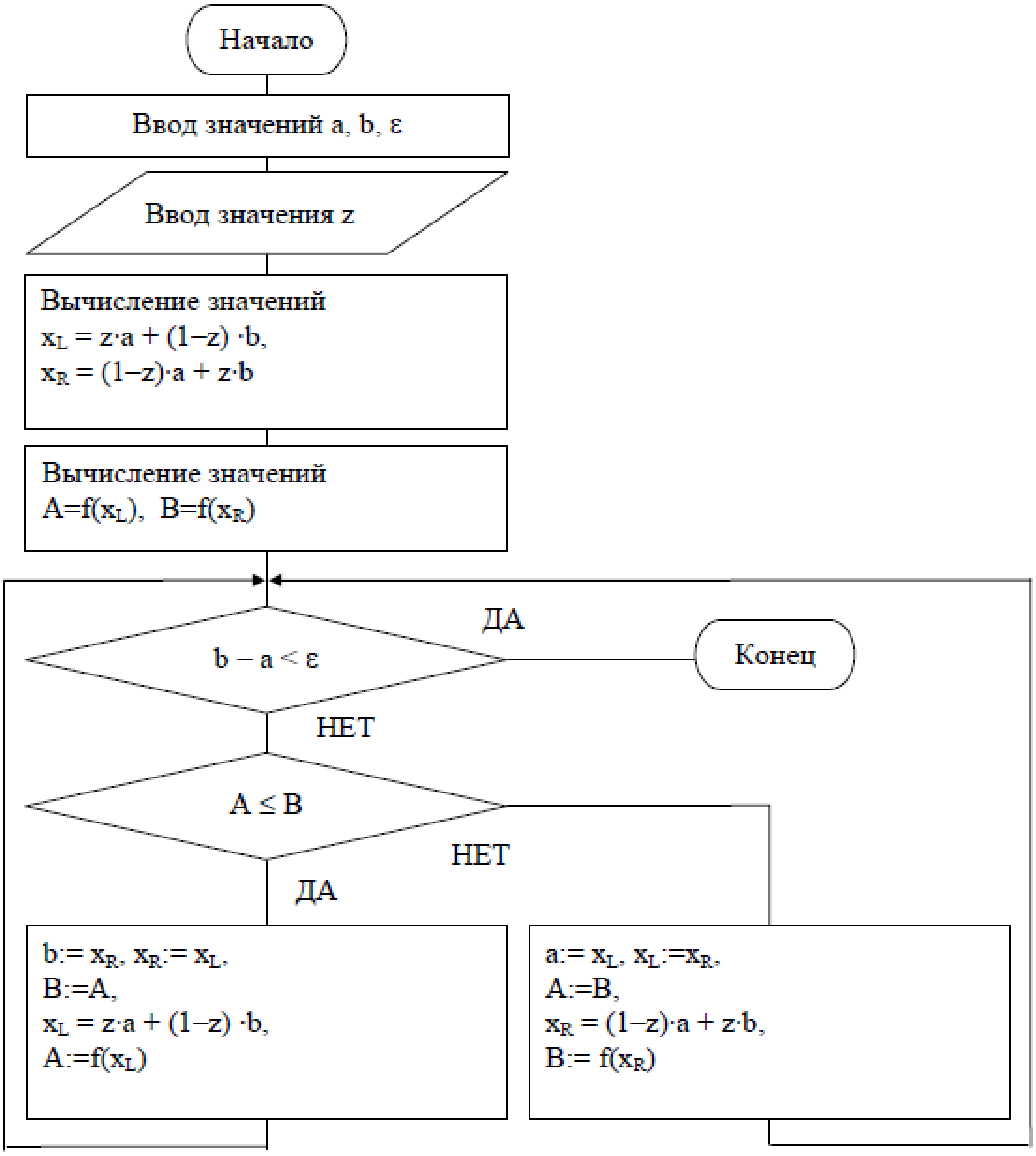

Рис. 1.8

Оптимізація функції за методом Фібоначчі виконується за чітко визначена кількість ітерацій. При цьому число необхідних ітерацій визначається необхідним значенням похибки результату чисельного рішення задачі.

На першій ітерації (k = 1) з чисел Фібоначчі вибирається мінімальне число, яке задовольняє умові

$$F_N \ge \frac{b-a}{\varepsilon},$$

де N - індекс обраного числа Фібоначчі. Число необхідних ітерацій методу визначається значенням N -1. У межах поточного інтервалу невизначеності обчислюються значення

$$x_L = a + \frac{F_{N-2}}{F_N}(b-a), \quad x_R = a + \frac{F_{N-1}}{F_N}(b-a).$$

Далі знаходяться значення функцій f (x_L), f (x_R) і порівнюються між собою. Стиснення інтервалу проводиться за правилами, викладеним при описі методу золотого перетину. Далі слід виконати наступні ітерації, обчислюючи значення x_L, x_R за формулами

$$x_L = a + \frac{F_{N-k-1}}{F_{N-k+1}}(b-a),\ x_R = a + \frac{F_{N-k}}{F_{N-k+1}}(b-a).$$

Обчислення закінчуються після виконання ітерації з номером k = *N*- 1.

Схема алгоритму оптимізації функції за методом золотого перетину представлена на рис. 1.9.

Істотним *недоліком* методу Фібоначчі є необхідність попередньої оцінки необхідного числа ітерацій. Ця особливість методу обумовлює його невисоку ефективність при вирішенні одновимірних підзадач багатовимірної оптимізації. Наприклад, на практиці нерідко потрібно знайти мінімум функції f (x) з деякою відносною похибкою δ або в результаті виконання алгоритму значення функції f (x) має бути зменшено в деяке число раз. У розглянутих випадках виконання попередньо встановленої кількості ітерацій може не привести до досягнення необхідного значення похибки.

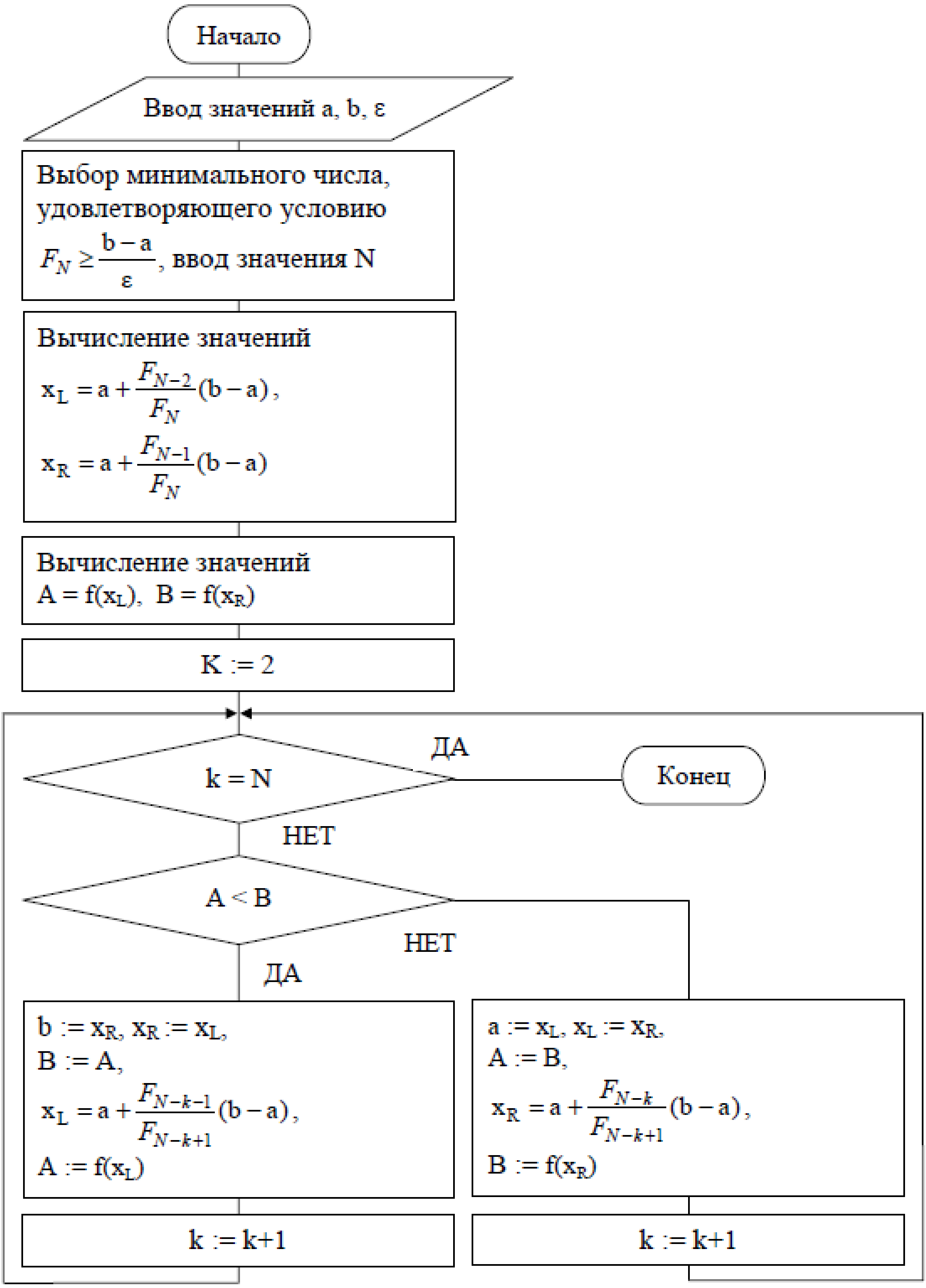

Рис. 1.9

Значним *недоліком* усіх розглянутих методів оптимізації одновимірних функцій є вимога наявності єдиного екстремуму в межах інтервалу [a, b].

У багатьох випадках завдання чисельного знаходження мінімуму функції є обумовленою [2]. Під обумовленістю обчислювальної задачі розуміють чутливість її рішення до погрішностей вхідних даних [7]. Завдання називається *добре обумовленої* , якщо малим похибок вихідних даних відповідають малі похибки рішення. В іншому випадку завдання називається *погано обумовленої* [7]. Для оцінки обумовленості завдання використовується так зване число обумовленості [7]. Обумовленість обчислювальної задачі пов'язана з метрологічними характеристиками обчислювального пристрою, на базі якого ця задача вирішується [5].

Похибки вхідних даних можуть мати різні складові. Однією з найбільш істотних складових є інструментальна похибка, викликана кінцевої розрядністю аналого-цифрового перетворювача або обчислювального пристрою. На рис. 1.10 показаний графік деякої функції, побудований в результаті виконання високоточних обчислень. Для даної функції результат чисельного рішення задачі пошуку мінімуму також може бути знайдений з високою точністю, тобтоІнтервал невизначеності може бути зменшений до дуже малих розмірів.

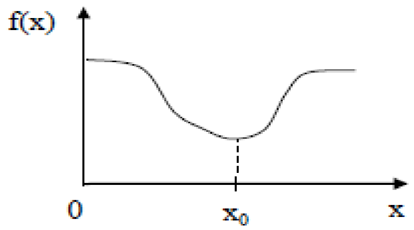

Рис. 1.10

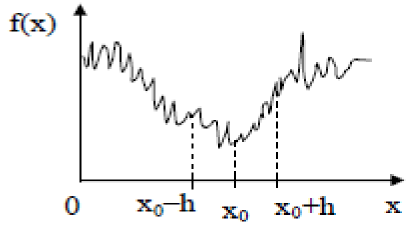

Рис. 1.11

На рис. 1.11 представлений графік функції, що обчислюється з похибкою. Відповідно до малюнків на графіку функції може бути виділено такий малий окіл точки мінімуму (x_0 - h, x_0 + h), для якого положення точки мінімуму не може бути визначено достовірно (мінімум функції не може бути достовірно визначений на основі порівняння обчислюваних значень функції). В даному випадку інтервал невизначеності положення локального мінімуму функції буде заданий значеннями [x_0 - h, x_0 + h]. Таким чином, похибка знаходження мінімуму функції в значній мірі залежить від похибки обчислення значень функції.

Одним з основних джерел обчислювальних похибок є наближене уявлення чисел в комп'ютері, обумовлене його кінцевою розрядністю. Сучасні обчислювальні пристрої можуть використовувати різні мікропроцесори, що підтримують уявлення чисел в формах з фіксованою і плаваючою комою (крапкою).

В даний час ПК підтримують стандарт двійковій арифметики з плаваючою комою IEEE 754-2008 (IEEE Standard for Floating-Point Arithmetic) [10]. Стандарт передбачає наступні основні типи чисел з плаваючою точкою: числа одинарної, подвійної і четверної точності. У стандарті також передбачені розширені формати даних [7]. Числа одинарної точності представляються 32 бітами (4 байтами) у вигляді: 1 біт під знак, 8 біт під порядок, 23 біта під мантиссу. Числа подвійної точності представляються 64 бітами (8 байтами) у вигляді: 1 біт під знак, 11 біт під порядок, 52 біта під мантиссу.

У програмному середовищі MATLAB за замовчуванням використовується уявлення чисел з подвійною точністю (тип даних - double). Для визначення будь-яких інших типів даних вказівку типу при оголошенні змінної є обов'язковим.

Відповідно до стандарту арифметика з плаваючою комою підтримує ряд спеціальних символів (нескінченність, нечислове значення та ін.). У програмному середовищі MATLAB символ $\pm\infty$ має позначення «inf», а нечислове значення позначається словом «NaN». Будь-яка операція, результат якої не може бути визначений коректно, генерує символ NaN (Not a Number - нечислове значення) [7].

Відносна похибка подання чисел в ПК визначається розрядністю мантиси. Значення відносної похибки подання числа не перевищує значення так званого «машинного Епсілон» [7, 8].

В системі MATLAB значення «машинного Епсілон» позначено змінною eps = 2^{-52} (для типу даних з подвійною точністю під мантиссу відводиться 52 біта), що приблизно дорівнює $2{,}2204 \cdot 10^{-16}$. Таким чином, відносна похибка подання числа в середовищі MATLAB (виражена у відсотках) не перевищує значення $\delta = 2^{-52} \cdot 100\%$. Для чисел з плаваючою комою відносна похибка має фіксоване значення, а абсолютна похибка може змінюватися.

Відносна похибка *будь-* арифметичної операції над числами, представленими у формі з плаваючою крапкою, також не перевищує значення «машинного Епсілон» [7].

Хід роботи

1. Запустити програмне середовище Matlab. Ознайомитись з документацією https://docs.exponenta.ru/optim/getting-started-with-optimization-toolbox.html?s_tid=CRUX_lftnav

2. У текстовому редакторі середовища Matlab створити новий файл і зберегти його під ім'ям f11.m;

3. У файлі f11.m помістити заголовок файлу-функції:

```
function y = f11 (x )
```

4. У файлі f11.m після наявного коду помістити запис виразу аналізованої функції (відповідно до варіанта завдань, вказаною викладачем):

```
y = x. * x-sin (x);
```

5. Збережемо зміни файлу f11.m;

6. У текстовому редакторі середовища Matlab створити новий файл і зберегти його під ім'ям gs_minimum.m;

7. У файлі gs_minimum.m помістити заголовок файлу-функції:

```
function gs_minimum (a, b, eps)

% --------------------------------------------
% Gs_minimum (a, b, eps) - функція для знаходження
% Мінімуму функції f11 (x). %%%
% Пошук проводиться методом золотого перетину
% На інтервалі [a, b] з похибкою eps.
% Функція f11 (x) задається у вигляді М-файлу,%
який повинен знаходитися в цій папці.
% (!!!) для коректного функціонування необхідно,
% Щоб виконувалася умова a <b і
% Шукане значення було єдиним% на
інтервалі [a, b].
```

8. У файлі gs_minimum.m помістити наступний програмний код (після наявного коду):

```
% -------------------------------------- ------
% Вивід повідомлення про помилку
```

```
if a> b
    error ( '"a" має бути менше "b"'); end

x = a: 0.001: b;
y = f11 (x);
plot (x, y, 'k', a, f11 (a), 'o', b, f11 (b), 'o');
text (a, f11 (a) + 5, 'f11 (a)', 'FontSize', 14);
text (b, f11 (b) + 5, 'f11 (b)', 'FontSize', 14);
xlabel ( 'x'); ylabel ( 'f11 (x)');
grid on; hold on;
```

9. У файлі gs_minimum.m помістити наступний програмний код (після наявного коду):

```
% ---------------------------------------------
% Виконання ітерацій
n = 0; % Кількість ітерацій
z = (sqrt (5) -1) / 2;
xL = z * a + (1-z) * b; xR = (1-z) * a + z * b;
A = f11 (xL); B = f11 (xR);
while 1
if b-a <eps
    break;
else
if A <= B %%% b = xR;
    xR = xL; B = A; xL = z * a + (1-z) * b; A = f11 (xL);
else
a = xL;
    xL = xR; A = B; xR = (1-z) * a + z * b; B = f11(xR);
end
end
x = (a + b) / 2;
fprintf ( '% s% .15f\n', 'x_min =', x);
fprintf ( '% s% .15f\n', 'f11 =', f11 (x));
plot (x, f11 (x), 'o');
n = n + 1;
end
% ---------------------------------------------
% Вивід результатів
text (x, f11 (x) -10, 'X_ {min}', 'FontSize', 14);
fprintf ( '% s\n', '');
fprintf ( '% s\n', ''Межі інтервалу невизн.:');
fprintf ( '% s% .15f\n', 'a =', a);
fprintf ( '% s% .15f\n', 'b =', b);
fprintf ( '% s\n', '');
fprintf ( '% s% d\n', 'n =', n);
hold off;
```

10. Збережемо зміни файлу gs_minimum.m;

11. Виконати пошук мінімуму функції **методом золотого перетину**. Значення меж інтервалу пошуку вибрати відповідно до варіанту завдань, вказаною викладачем. Наприклад, для запуску процесу пошуку мінімуму функції в інтервалі [-5, 10] в головному вікні середовища Matlab слід виконати наступну команду:

```
>> gs_minimum (-5, 10, 0.01)
```

12. У текстовому редакторі середовища Matlab створити новий файл і зберегти його під ім'ям gs_maximum.m;

13. У файлі gs_maximum.m помістити код файлу-функції, що виконує пошук максимуму методом золотого перетину;

14. Збережемо зміни файлу gs_maximum.m;

15. Виконати пошук максимуму функції методом золотого перетину. Значення меж інтервалу пошуку вибрати відповідно до варіанту завдань, вказаною викладачем. Наприклад, для запуску процесу пошуку максимуму функції в інтервалі [-5, 10] в головному вікні середовища Matlab слід виконати наступну команду:

```
>> gs_maximum (-5, 10, 0.01)
```

16. У текстовому редакторі середовища Matlab створити новий файл і зберегти його під ім'ям fib_minimum.m на жорсткому диску ПК в папці X: \ ... \ Komplex_MO \ Lab1;

17. У файлі fib_minimum.m помістити заголовок файлу-функції:

```
function fib_minimum (a, b, eps)

if a> b
    error ( ' "a" не має бути менше "b"'); end

x = a: 0.001: b;
y = f11 (x);
plot (x, y, 'k', a, f11 (a), 'o', b, f11 (b), 'o');
text (a, f11 (a) + 5, 'f11 (a)', 'FontSize', 14);
text (b, f11 (b) + 5, 'f11 (b)', 'FontSize', 14);
xlabel ( 'x'); ylabel ( 'f11 (x)');
grid on; hold on;

ratio = (b-a) / eps; n = 3;
f = [1 1];
disp(f(1))
disp(f(2))

while 1
f(n) = f(n-1) + f(n-2);
asd = f(n);
disp(f(n))
if asd>ratio
    break;
end
n = n+1;
end
N=n;

xL = a + (f(N-2)/f(N)) * (b-a);
xR = a + (f(N-1)/f(N)) * (b-a);
A = f11(xL); B = f11(xR);
```

```
for k = 2: N-1
if A<B
    b = xR;
xR = xL; B=A;xl = a+(f(N-k-1)/f(N-k+1)) * (b-a);
A = f11(xL);
else a = xL;
xL = xR;
A=B; xR=a+(f(N-k)/f(N-k+1)) * (b-a);B=f11(xR);
end
x = (a+b)/2;
fprintf ( '% s% d', 'k =', k);
fprintf ( '% s% .15f', 'x_min =', x);
fprintf ( '% s% .15f', 'f11 =', f11(x));
plot(x, f11(x), 'o');
end

text (x, f11 (x) -10, 'X_ {min}', 'FontSize', 14);
fprintf ( '% s\n', '');
fprintf ( '% s\n', 'Результати виконання ітерацій.:');
fprintf ( '% s% .15f\n', 'a =', a);
fprintf ( '% s% .15f\n', 'b =', b);
hold off;
```

22. Збережемо зміни файлу fib_minimum.m;

23. Виконати пошук мінімуму функції методом Фібоначчі. Значення меж інтервалу пошуку вибрати відповідно до варіанту завдань, вказаною викладачем. Наприклад, для запуску процесу пошуку мінімуму функції в інтервалі [-5, 10] в головному вікні середовища Matlab слід виконати наступну команду:

```
>> fib_minimum (-5,10,0.01)
```

24. **Порівняти значення довжини інтервалу невизначеності, отримані в результаті пошуку мінімуму функції методами Фібоначчі і золотого перетину.**

25. **За результатами виконання п. 24 зробити висновок про переваги і недоліки методів Фібоначчі і золотого перетину.**

	Мінімум	Максимум	Кількість ітерацій
Золотого перетину			
Метод Фібоначчі			

26. У текстовому редакторі середовища Matlab створити новий файл і зберегти його під ім'ям fib_maximum.m на;

27. У файлі fib_maximum.m помістити код файлу-функції, що виконує пошук максимуму методом Фібоначчі;

28. Збережемо зміни файлу fib_maximum.m;

29. Виконати пошук максимуму функції методом Фібоначчі. Значення меж інтервалу пошуку вибрати відповідно до варіанту завдань, вказаною викладачем. Наприклад, для запуску

процесу пошуку максимуму функції в інтервалі [-5, 10] в головному вікні середовища Matlab слід виконати наступну команду:

```
>> fib_maximum (-5,10,0.01)
```

Варіанти завдань наведені в таблиці.

Таблиця1. Варіанти завдань

1. $f(x)=\sin(x)$, $x\in[-\pi,\pi/2]$	2. $f(x)=\cos(x)$, $x\in[0,\pi]$.
3. $f(x)=(x-2)^2$, $x\in[0,3]$	4. $f(x)=(x-15)^2+5$, $x\in[12,20]$
5. $f(x)=(x+5)^4$, $x\in[-6,2]$	6. $f(x)=xe^x$, $x\in[-2,0]$,
7. $f(x)=x^2+2x-4$, $x\in[-2,1]$	8. $f(x)=x^3-x$, $x\in[0,1]$
9. $f(x)=x^5-x^2$, $x\in[0,1]$	10. $f(x)=-x/e^x$, $x\in[0,3]$
11. $f(x)=x^4-x$, $x\in[0,1]$	12. $f(x)=x^4/\ln x$, $x\in[1.1,1.5]$
13. $f(x)=xe^{-x}$, $x\in[-2,6]$	14. $f(x)=xe^{-2x}$, $x\in[-2,6]$

Контрольні питання

1. Наведіть послідовність команд на мові Matlab, необхідну для реалізації методу золотого перетину. Поясніть призначення кожної наведеної команди.
2. Наведіть послідовність команд на мові Matlab, необхідну для реалізації методу Фібоначчі. Поясніть призначення кожної наведеної команди.
3. Наведіть послідовність команд на мові Matlab, необхідну для реалізації методу половинного ділення. Поясніть призначення кожної наведеної команди.
4. В чому полягає сутність методів Фібоначчі і золотого перетину ? Як пов'язані між собою методи Фібоначчі і золотого перетину?
5. Що означають терміни «критерій оптимальності», «цільова функція», «оптимальне значення критерію оптимальності», «поточний інтервал невизначеності», «модель об'єкта проектування», «вектор керованих параметрів», «область допустимих значень вектора керованих параметрів»?
6. За якими ознаками можуть бути класифіковані критерії оптимальності? Що таке детермінований, стохастичний, унімодальне, багатоекстремального, опуклий критерії оптимальності?
7. Що означає термін «задача локальної оптимізації»? Що таке локальний і глобальний мінімуми?
8. Що означає термін «задача умовної оптимізації»? Що таке обмежує функція змінних параметрів об'єкта? Як класифікуються обмеження варійованих параметрів об'єкта в завданні умовної оптимізації?
9. Що означає термін «критерій якості алгоритму оптимізації»?
10. Які умови закінчення пошуку найбільш часто використовуються в алгоритмах оптимізації? Наведіть формули найбільш часто використовуваних умов.

11. Назвіть найбільш відомі методи оптимізації одновимірних функцій, поясніть сутність кожного методу, перерахуйте їх переваги і недоліки.

12. Як визначаються абсолютна і відносна похибки результату чисельного рішення задачі?

13. Що означає термін «метод оптимізації нульового порядку»? Назвіть найбільш відомі методи оптимізації нульового порядку, поясніть сутність кожного методу, перерахуйте їх переваги і недоліки.

14. Що таке швидкість збіжності методу оптимізації?

15. Які вимоги пред'являються до чисельних алгоритмах оптимізації?

16. Що таке коректність і обумовленість обчислювальної задачі?

17. Яке значення має похибка арифметичних операцій в системі Matlab? Яке значення має похибка подання числа в форматі double в системі Matlab?

18. Які спеціальні символи підтримує арифметика з пла-вающей коми відповідно до стандарту IEEE 754-2008?