МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ЗАДАНИЕ
по лабораторной работе №1
по дисциплине «Методы оптимизации»
Тема: «Простые методы одномерного поиска минимума»

Преподаватель: **Каримов А.И.**

Санкт-Петербург
2019

Цель работы

Изучение среды MATLAB, создание программы для реализации одного из простейших методов одномерного поиска: метода дихотомии, метода трехточечного деления, метода золотого сечения, метода Фибоначчи.

Основные теоретические положения

Методы одномерного поиска минимума используются для решения одномерной экстремальной задачи

$$x^* = \underset{x \in [a;b]}{\arg\min} f(x), \tag{1}$$

где интервал $[a; b]$ задан, функция $f(x)$ унимодальна на данном интервале.

Метод дихотомии (метод Больцано)

Метод средней точки является вариантом метода деления интервала пополам. Последовательные сокращения интервала неопределенности произво-дятся на основе оценки производной минимизируемой функции в центре текущего интервала.

Начальный этап Для запуска метода необходимо:

(1) задать $[a_1, b_1]$ – начальный интервал локализации минимума, на границах которого знаки производных различны, т. е. $f'(a_1)f'(b_1) < 0$; ϵ, δ – малые положительные числа;

(2) установить счетчик числа итераций $k = 1$.

Основной этап

(1) Взять пробную точку x_k в центре текущего интервала

$$x_k = \frac{a_k + b_k}{2}$$

и проверить критерий окончания поиска (КОП): если

$$|f'(x_k)| \leq \epsilon \text{ и } |b_k - a_k| \leq \delta,$$

то остановиться ($x_k = x^*$ - аппроксимирующий минимум).

(2) Сократить текущий интервал: если $f'(x_k) > 0$, то положить $a_{k+1} = a_k$ и $b_{k+1} = x_k$, в противном случае – $a_{k+1} = x_k$, $b_{k+1} = b_k$; заменить k на $k + 1$ и вернуться на шаг (1).

Метод трехточечного деления

В реальной вычислительной практике нахождение производной на каждой итерации может быть сопряжено со значительными трудностями. Метод трехточечного деления свободен от этого недостатка, поскольку не требует вычисления производной на шаге.

Начальный этап

(1) Задать $[a_1, b_1]$ – начальный интервал поиска, где a_1, b_1 - границы интервала, удовлетворяющие условию $f'(a_1)f'(b_1) < 0$; ϵ – погрешность вычисления минимума x^*.

(2) Положить $x_m = (a_1 + b_1)/2$ и $k = 1$.

Основной этап

(1) Взять 2 пробные точки

$$x_1 = a_k + L_k/4 \text{ и } x_2 = b_k - L_k/4,$$

где $L_k = |b_k - a_k|$ – длина текущего интервала. Точки x_1, x_2 и x_m делят $[a_k, b_k]$ на 4 равные части.

(2) Сократить текущий интервал локализации минимума:

(2.1) если $f_1 < f_m$, то положить $a_k + 1 = a_k$, $b_{k+1} = x_m$, $x_m = x_1$, перейти к шагу (3);

(2.2) если $f_1 \geq f_m \leq f_2$, то положить $a_{k+1} = x_1$, $b_{k+1} = x_2$; иначе – $a_{k+1} = x_m$, $b_k + 1 = b_k$, $x_m = x_2$.

(3) Проверить критерий окончания поиска:

(3.1) заменить k на $k + 1$;

(3.2) если $L_k = |b_k - a_k| \leq \epsilon$, то остановиться. Если данное условие не выполняется, вернуться к шагу (1).

Метод золотого сечения

Метод золотого сечения является процедурой линейного поиска минимума унимодальной функции $f(x)$ на замкнутом интервале $[a, b]$, использующий не 3, а только 2 вычисления целевой функции на шаге, пользуясь свойсвом самоподобия отрезков, разделенных в отношении золотого сечения.

Начальный этап

Задать начальный интервал $[a_1, b_1]$, длину конечного интервала L_n и определить константу $\Phi = \frac{1+\sqrt{5}}{2}$. Определить малое положительное число ϵ – константу различимости двух значений x. Положить $k = 1$. Определить $L_1 = b_1 - a_1$.

Основной этап

(1) Найти

$$\lambda_k = b_1 - \frac{L_k}{\Phi} \text{ и } \mu_k = a_1 + \frac{L_k}{\Phi}.$$

(2) Сократить текущий интервал локализации: если $f(\lambda_k) > f(\mu_k)$, то положить $a_{k+1} = \lambda_k$, иначе $b_{k+1} = \mu_k$. Положить $k = k + 1$.

(3) Проверить критерий окончания поиска: если $L_k = |b_k - a_k| \leq \epsilon$, то остановиться. Если данное условие не выполняется, вернуться к шагу (1).

Метод Фибоначчи

Метод Фибоначчи является процедурой линейного поиска минимума унимодальной функции $f(x)$ на замкнутом интервале $[a, b]$, отличающейся от процедуры золотого сечения тем, что очередная пробная точка делит интервал локализации в отношении двух последовательных чисел Фибоначчи. Последовательность чисел Фибоначчи задается условиями $F_0 = F_1 = 1$, $F_{k+1} = F_k + F_{k-1}$, $k = 1, 2, \ldots$. Начальными членами последовательности будут $1, 1, 2, 3, 5, 8, 13, \ldots$. Стратегия поиска Фибоначчи требует заранее указать n – число вычислений минимизируемой функции и ϵ – константу различимости двух значений $f(x)$. Рассмотрим один из возможных вариантов метода.

Начальный этап

(1) Задать начальный интервал $[a_1, b_1]$, длину конечного интервала L_n и определить число n так, чтобы выполнялось условие $F_n > (b_1 - a_1)/L_n$. Вычислить $\epsilon \leq L_1/F_n + 1$.

(2) Взять 2 пробные точки

$$\lambda_1 = a_1 + \frac{F_{n-2}}{Fn}(b_1 - a_1) \text{ и } \mu_1 = a_1 + \frac{F_{n-1}}{F_n}(b_1 - a_1).$$

Положить $k = 1$.

Основной этап

(1) Сократить текущий интервал локализации:

(1.1) если $f(\lambda_k) < f(\mu_k)$, то положить $a_k + 1 = a_k$, $b_{k+1} = \mu_k$, $\mu_{k+1} = \lambda_k$ и вычислить новую точку

$$\lambda_{k+1} = ak + 1 + \frac{F_{n-k-2}}{F_{n-k}} L_{k+1}, \text{ где } L_{k+1} = b_{k+1} - a_{k+1};$$

перейти к шагу 2; (1.2) если $f(\lambda_k) \geq f(\mu_k)$, то положить $a_{k+1} = \lambda_k$, $b_{k+1} = bk$, $\lambda_{k+1} = \mu_k$ и вычислить

$$\mu_{k+1} = a_{k+1} + \frac{F_{n-k-1}}{F_{n-k}} L_{k+1}.$$

(2) Проверить критерий окончания поиска:
(2.1) заменить k на $k+1$;
(2.2) если $k = n-1$, перейти к шагу 3, иначе – к шагу 1.
(3) Найти аппроксимирующий минимум x^*:
(3.1) положить $\mu_k = \lambda_k + \epsilon$;
(3.2) если $f(\lambda_k) > f(\mu_k)$, то $x^* = (\lambda_k + b_k)/2$. В противном случае – $x^* = (a_k + \mu_k)/2$.

Реализация в MATLAB

При реализации методов следует придерживаться принципа: один метод, одна функция – один `*.m`-файл. Следует тщательно комментировать код и писать раздел описания файла в начале, чтобы можно было воспользоваться командой `help`.

Пример кода

Пример грамотно оформленного кода приведен в Листинге 1.

Листинг 1: Метод золотого сечения

```
function [xmin, fmin, neval] = goldensectionsearch(f,interval,tol)
% GOLDENSECTIONSEARCH searches for minimum using golden section
%        [xmin, fmin, neval] = GOLDENSECTIONSEARCH(f,interval,tol)
%    INPUT ARGUMENTS:
%        f is a function
%        interval = [a, b] - search interval
%        tol - set for bot range and function value
%    OUTPUT ARGUMENTS:
%        xmin is a function minimizer
%        fmin = f(xmin)
%        neval - number of function evaluations

        %unparse the search interval
        a = interval(1);
        b = interval(2);
        L = realmax; %set the largest real number
        neval = 0;
        Phi = (1 + sqrt(5))/2; %constant Phi

        while L > tol
                L = b - a;
                %find two points in a golden ratio
                x1 = b - L/Phi;
                x2 = a + L/Phi;
                %evaluate th objective function
                y1 = feval(f,x1);
                y2 = feval(f,x2);
                %set new bounds
                if y1 > y2
```

```
                        a = x1;
                        xmin = x2;
                        fmin = y2;
                else
                        b = x2;
                        xmin = x1;
                        fmin = y1;
                end
                %estimate the number of evaluations
                neval = neval + 2;
        end
end
```

Проверьте, как работает команда `help`, введя в командную строку
`>>help goldensectionsearch`
Убедитесь, что заглавные буквы в справочной части комментариев соответствуют полужирному шрифту при выводе справки на экран.

Визуализация

Для вывода графиков существует несколько команд MATLAB. Пусть имеются два вектора данных `X` и `Y` одинакового размера. Для начала, вызовите команду

```
figure(1)
```

чтобы перевести контекст на окно 1, где в дальнейшем будут выводиться графики. Затем, постройте в окне график с линейным масштабом осей

```
plot(X,Y)
```

Чтобы при перерисовке нового графика имеющиеся графики сохранялись, необходимо использовать команду `hold on`. Для отключения перерисовки используется команда `hold off`.

Для отрисовки графиков в логарифмическом масштабе по осям x, y и обеим осям соответственно, используйте команды

```
semilogx(X,Y);
semilogy(X,Y);
loglog(X,Y);
```

Для рисования линии с заданными координатами начала и конца используйте функцию

```
line(X,Y)
```

Чтобы вывести графики в одном окне, воспользуйтесь функцией

```
subplot(m,n,i);
```

Аргументы функции `subplot` – число графиков по вертикальной оси, число графиков по горизонтальной оси, и номер графика, начиная с левого верхнего, соответственно.

Графики по умолчанию выводятся разными цветами, начиная с синего, и непрерывной линией. Чтобы узнать о форматировании графиков, используйте команду `help` для вызова текстовой справки или `doc` для вызова справки в браузере.

Визуализируем статистику количества вычислений целевой функции и достигнутой погрешности в зависимости от заданной точности. Пусть заданная точность будет отложена по оси x, а число вычислений – по оси y. Аналогично, построим график заданной точности и график фактически достигнутой погрешности. В Листинге 2 приведен пример кода, реализующий эту визуализацию.

Листинг 2: Визуализация статистики

```
% TEST GOLDEN SECTION SEARCH
interval = [0, 10];
%set the number of points on a plot
N = 30;
tolspan = logspace(-14,0,N); %evenly spaced logarithmic sequence
Xmin  = zeros(1,N);
Neval = zeros(1,N);
i = 1;%iteration counter
for tol = tolspan %select the required tolerance
        [xmin, ~, neval] = goldensectionsearch(@f,interval,tol); % search for the xmin
        Xmin(i) = xmin; %save stats
        Neval(i) = neval;
        i = i + 1;
end
figure(1);
%plot the number of evaluations
subplot(2,1,1);
semilogx(tolspan,Neval,'.-b');
xlabel('tol');
ylabel('Neval');
%plot the error
subplot(2,1,2);
loglog(tolspan,abs(Xmin - 4),'s-b');
xlabel('tol');
ylabel('err');
```

Результат работы кода представлен на рисунке 1.

Интерпретируйте полученные результаты.

Второй вариант визуализации – визуализация хода решения задачи. Пусть оптимизируется функция

$$f(x) = (x-4)^2 \tag{2}$$

на интервале $[-2; 10]$. Процесс последовательного вычисления точек, приближающихся к минимуму, представлен на рисунке 2.

Содержание работы и отчета

Лабораторная работа должна включать в себя *.m-файлы кодов на MATLAB для решения задачи. Отчет должен содержать титульный лист и разделы:

- Цель работы
- Основные теоретические положения.
- Коды для решения задачи оптимизации.

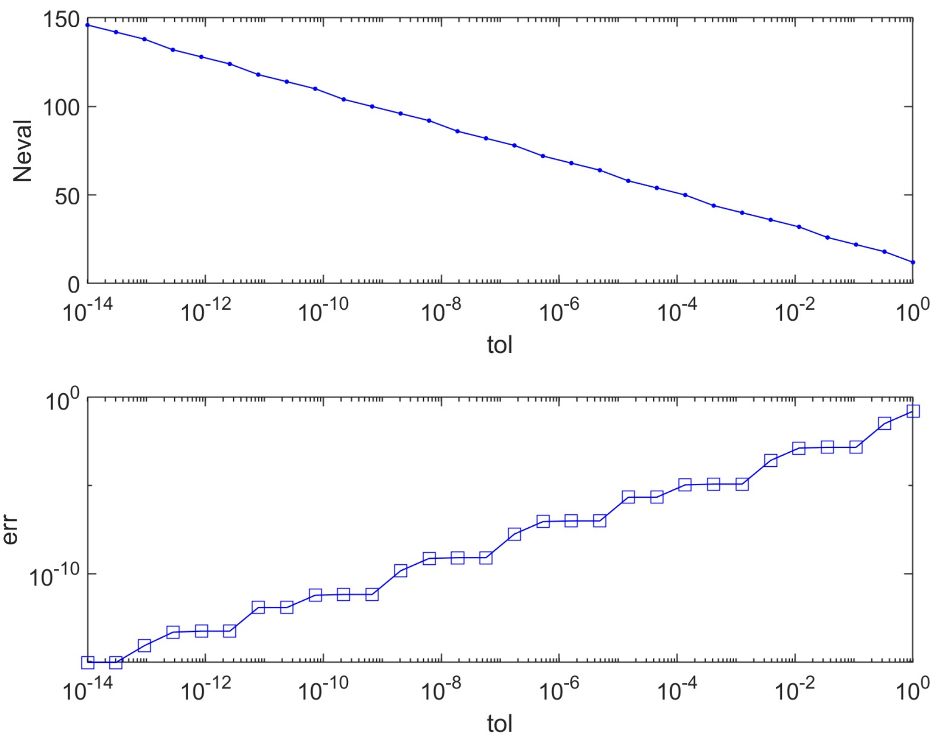

Рис. 1: Статистика количества вычислений функции и достигнутой точности

Номер	Функция	Минимум x^*
1	$(x-3)^2 - 3x + x^2$	2,25
2	$(x-4)^4 + 8\sin(x) + 5$	≈ 4.602703
3	$\exp(x^2)^0.1$	0
4	$x^2 - 4\cos(2\pi x)$	0
5	$\cos(x)\exp(-(x-\pi)^2)$	π
6	$-2\sin\left(\sqrt{\lvert x/2+10\rvert}\right) - x\sin\left(\sqrt{\lvert x-10\rvert}\right)$	≈ 8.310296061
7	$(x-3)^2$	3
8	$\cos(2\pi(x+2)/10)$	3

Таблица 1: Тестовые функции

- Сравнение двух методов (по вариантам). Для этого на одном графике выведите число вычислений целевой функции в зависимости от требуемой точности для каждой из заданных тестовых функций. Аналогично, визуализируйте фактическую погрешность для обоих методов на другом графике.

- Выводы.

В приложении к данному документу находятся *.m-файлы, реализующие функции:

```
f %objective function (2)
goldensectionsearch %golden section search implementation
goldensectionsearchvisual %golden section search implementation
    with visualization
test_goldensection %golden section search test
test_goldensectionsearchvisual %search test with visualization
```

В Таблице 1 приведены тестовые функции. Необходимо найти их минимум на интервале $[-2; 10]$. Учтите, что некоторые из них не унимодальны!

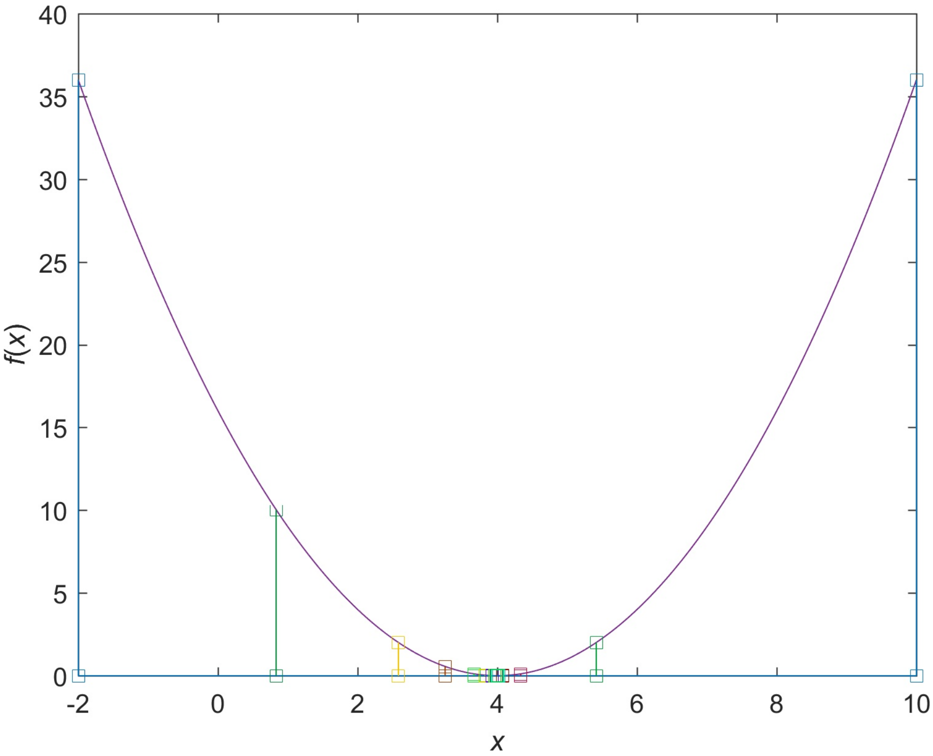

Рис. 2: Иллюстрация процесса оптимизации

Таблица 2 содержит варианты заданий по вариантам. Номерами методов обозначены: 1 - Больцано, 2 - трехточечного деления, 3 - золотого сечения, 4 - Фибоначчи.

Вариант	Номера функций	Номера методов
1	1,5	2,1
2	2,7	1,2
3	3,4	2,3
4	2,3	2,4
5	7,5	1,3
6	5,8	3,4
7	8,2	1,2
8	6,8	3,2
9	7,1	3,1
10	7,4	3,2
11	3,4	4,1
12	4,5	3,2
13	1,8	3,1
14	3,7	1,3
15	2,6	2,1
16	3,1	3,2

Таблица 2: Варианты