МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»
КАФЕДРА №51

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

ИНТЕРВАЛЬНЫЕ МЕТОДЫ ОДНОПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ

по курсу: МЕТОДЫ МОДЕЛИРОВАНИЯ И ОПТИМИЗАЦИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	5711М		Пятаков В.С.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2017

Цель работы

Сравнить интервальные методы однопараметрической оптимизации по эффективности.

Постановка задачи однопараметрической оптимизации

Однопараметрическая оптимизация – поиск экстремумов функций одной переменной без наличия ограничений. Такие задачи могут быть как самостоятельными, так и частью более сложных задач поиска экстремума функции многих переменных.

$$f(x) \to \min, x \in R^1,$$
$$K = J = 0, D \equiv R^1.$$

где $x -$ x - одномерный вектор аргументов; $R^1 -$ множество допустимых значений вектора x; вид функции $f(x)$ произвольный.

Сравнение методов поиска экстремума функции выполним по двум характеристикам:

1. скорости сходимости;
2. числу шагов k получения экстремума с точностью ε .

Пусть показателем скорости сходимости будет величина:

$$\alpha(N) = \frac{L_N}{L_1}$$

где N – число итераций;

$L_1 -$ длина интервала [a, b] на первой итерации, $x \in [a,b]$;

$L_N -$ длина интервала $[a^N, b^N]$ на последней *N*-й итерации, $x^* \in [a^N, b^N]$.

Для нахождения числа шагов *N* воспользуемся тем же соотношением

$$L_N = L_1 \alpha(N).$$

Порядок выполнения работы

1. Реализовать алгоритмы интервальных методов однопараметрической оптимизации функции *f* согласно варианту .
2. Найти оценку числа итераций *N* для каждого метода.
3. Найти оценку показателя скорости α для каждого метода.
4. Количественные характеристики интервальных методов, полученные в результате решения однопараметрической оптимизации функции f записать в таблицу 1.

Описание алгоритмов

Метод дихотомии

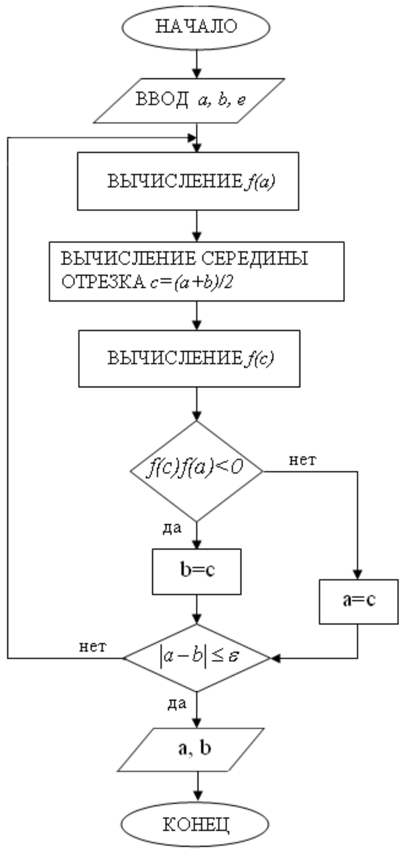
Считаем, что отделение корней произведено и на интервале [a,b] расположен один корень, который необходимо уточнить с погрешностью ε.

Итак, имеем $f(a)f(b)<0$. Метод дихотомии заключается в следующем. Определяем половину отрезка $c=\frac{a+b}{2}$ и вычисляем *f(c)*. Проверяем следующие условия :

1. Если $|f(c)| < ε$, то *c* – корень. Здесь ε - заданная точность.
2. Если $f(c)f(a)<0$, то корень лежит в интервале [a,c].
3. Если $f(c)f(b)<0$, то корень лежит на отрезке[c,b].

Продолжая процесс половинного деления в выбранных подинтервалов, можно дойти до сколь угодно малого отрезка, содержащего корень ε.

Алгоритм



Вис.1 - Структурная схема алгоритма метода дихотомии

Метод золотого сечения

Считаем, что отделение корней произведено и на интервале [a,b] расположен один корень, который необходимо уточнить с погрешностью ε.

1. Вычисляется значение функции $f(x_1)$, где $x_1=a+0{,}382(b-a)$.
2. Вычисляется значение функции $f(x_2)$, где $x_1=b+0{,}382(b-a)$.
3. Определяется новый интервал (a,x2) или (x1,b), в котором локализован минимум.
4. Внутри полученного интервала находится новая точка (x_1 в случае 1) или (x_2 в случае 2), отстоящая от его конца на расстоянии, составляющем *0,382* от его длины. В этой точке рассчитывается значение $f(x)$. Затем вычисления повторяются, начиная с пункта 3, до тех пор, пока величина интервала неопределенности станет меньше или равна ε, где ε - заданное сколь угодно малое положительное число.

Алгоритм

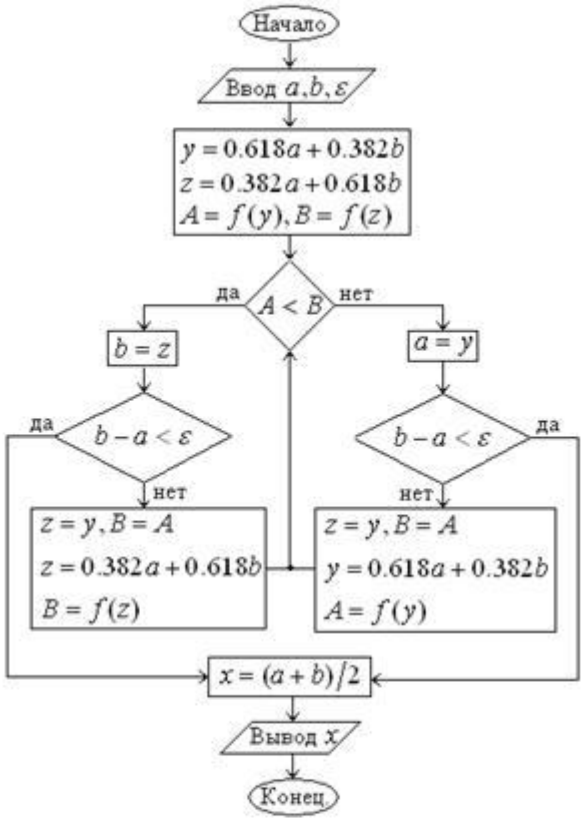


рис.2 - Структурная схема алгоритма метода золотого сечения

Метод равномерного поиска

Этот метод основан на том, что переменной x присваиваются значения $x_i = x_0 + n$ с шагом n=const (шагом поиска), где $i=0,1,2,\dots$ и вычисляются значения $f(x)$ в соседних точках. Если $f(x_{i+1}) \le f(x_i)$, то переменной x дается новое приращение. Как только становится $f(x_{i+1}) > f(x_i)$, поиск останавливается и предпоследняя точка считается ответом.

Выбор x_0 (начального значения переменной x_0) определяется пользователем. Шаг поиска - фактическая погрешность определения результата. При поиске решения на отрезке, обычно в качестве начального приближения берут один из его концов, а при изменении переменной x предусматривается проверка на выход ее за границу отрезка.

Алгоритм

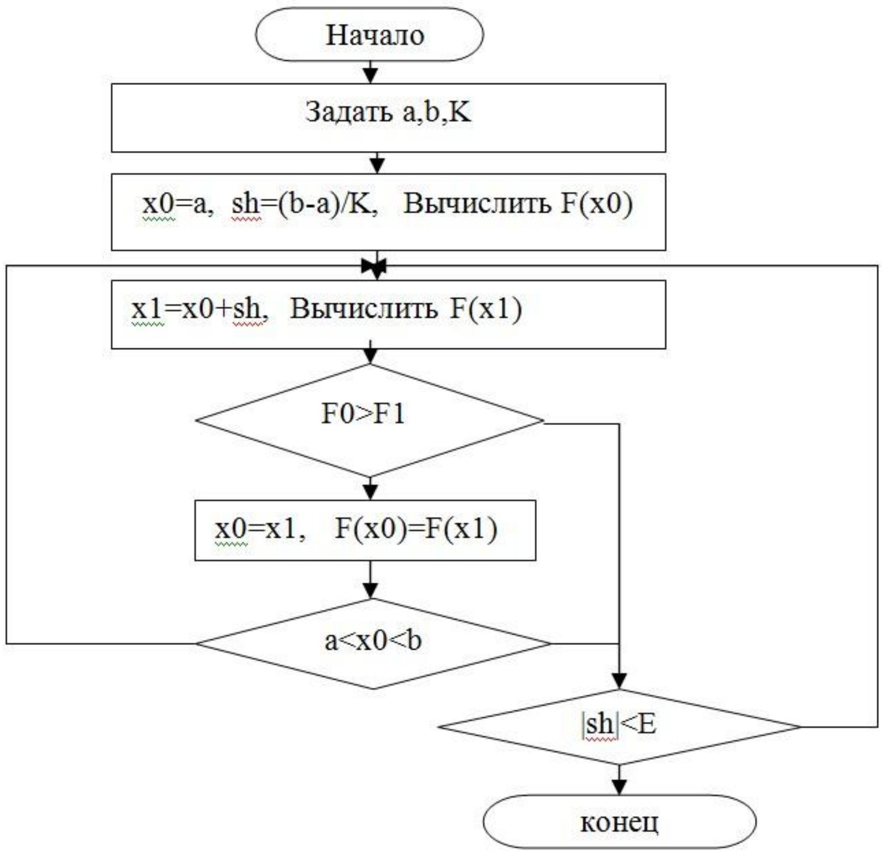


рис.3 - Структурная схема алгоритма метода равномерного поиска

Полученные результаты

Вариант 14

f(x)= -5x 2 +4,4x+0.6

Экстремум = 0.4

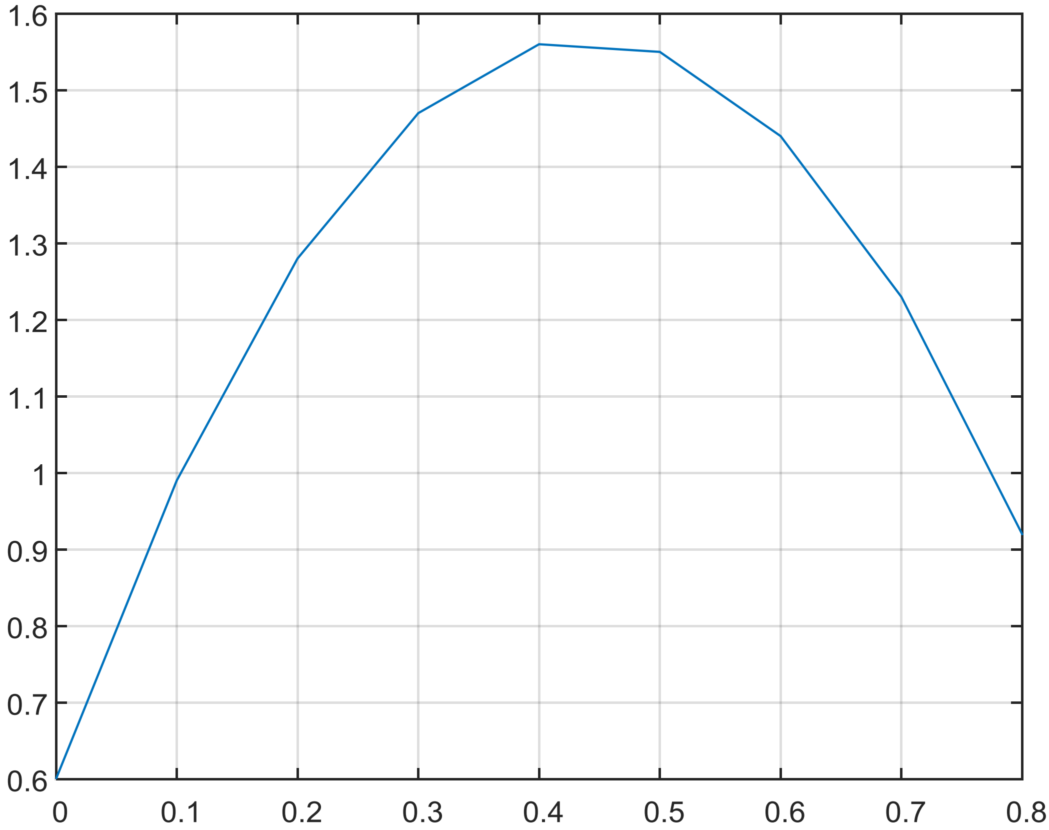


рис.4 - График *f(x)*

Таблица 1 - Сравнительные характеристик интервальных методов на интервале [-10,10]

Метод	*N*			α		
	$\varepsilon=10^{-1}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-3}$	$\varepsilon=10^{-1}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-3}$
Дихотомии	8	11	15	0.0039062	0.0004882	0.0000305
Золотого сечения	12	16	21	0.0031056	0.0004531	0.0000408
Равномерного поиска	99	999	9999	0.0202020	0.0020020	0.0002000

Выводы

По данным полученным в ходе исследования и внесенных в таблицу 1 можно сделать следующие выводы:

1. Сравнивая три метода оптимизации по параметру количества итераций необходимых для нахождения экстремума, можно сказать, что метод дихотомии дает наилучшие результаты по сравнению с двумя другими методами, так как при одном и том же значении точности епсилон, задача решается с использованием меньшего числа итераций.
2. По скорости сходимости метод равномерного поиска является наилучшим, но сильно проигрывает методам дихотомии и золотого сечения по количеству итераций, необходим для решения задачи, что можно было предположить, так как по виду своему метод напоминает метод полного перебора, а с точки зрения скорости, этот метод всегда являлся наихудшим.
3. Метод золотого сечения дает средние значения по количеству итераций, по сравнению с другими двумя методами, но при низкой точности выигрывает у метода дихотомии по скорости сходимости.

Список использованной литературы

1. Татарникова Т.М. Методы моделирования и оптимизации: Методические указания к выполнению лабораторных работ. СПб.: ГУАП, 2017
2. https://textbook.news/optimizatsiya/metod-ravnomernogo-poiska-61582.html
3. Методи оптимізації. Практикум: Навчальний посібник. – Лу- ганськ: Вид-во НЦППРК “Ноулідж”, 2006.
4. https://studfiles.net/preview/3802031/
5. http://dit.isuct.ru/IVT/sitanov/Literatura/M171/Pages/Glava1_3.htm

ПРИЛОЖЕНИЕ

Листинг программ реализующих методы оптимизации реализованных в среде matlab

Метод дихотомии

```
close all;
clear all;
clc;
%% метод дихотомии
%%вычисение производной исходной функции
syms x;
y=-5*x^2 +4.4*x+0.6;
f=diff(y);
%% выбор интервала [a,b]
a=-10;
b=10;
interval_begin=b-a;%длина интервала в начале
n=0;%счетчик итераци, не знаю зачем если есть для этого формула
пусть будет
eps=0.001;%точность(епсилент)1

d=b-a;

%% основной цикл
while b-a>eps

    c=(a+b)/2;
    x=a;
    f_a=eval(f);%подстановка начала интервала
    x=c;
    f_c=eval(f);%подстановка вычисленного деленного пополам

    r=f_a*f_c;

    % выбор интервала либо [a,x], либо [x,b]
    if r>0
        a=c;
    else
        b=c;
    end
    n=n+1;
end

interval_end=b-a;%длина интервала в конце
%%
speed_sxod=interval_end/interval_begin;% скорость схдимости
N_step=speed_sxod*interval_begin;%количество шагов
```

Метод золотого сечения

```
close all;
clear all;
clc;

%y=-5*x^2 +4.4*x+0.6;
a=-10;
b=10;
eps=0.01;
Q=(-1+sqrt(5))/2;

interval_begin=b-a;%длина интервала в начале
nn=0;
while b-a>eps
    x1=b-((b-a)*Q);
    x2=a+((b-a)*Q);

    y1=f(x1);
    y2=f(x2);

    if y1>y2
        b=x2;
    else
        a=x1;
    end
    nn=nn+1;
end

x=(a+b)/2;

interval_end=b-a;%длина интервала в конце
%%
speed_sxod=interval_end/interval_begin;% скорость схдимости
N_step=speed_sxod*interval_begin;%количество шагов
```

Метод равномерного поиска

```
close all;
clear all;
clc;

%% метод равномерного поиска


%% выбор интервала [a,b]
a=0;
b=10;
interval_begin=b-a;%длина интервала в начале
eps=0.01;%точность(епсилент)1
dx =  (b-a)/eps-1     ;
```

```
yMin = f(a);
xMin = a;
nn=0;
for i=1:1:dx
    x=a+i*eps;
    y=f(x);
            if y>yMin
        xMin = x;
        yMin = y;

            end
            nn=nn+1;
end



interval_end=2/dx;%длина интервала в конце
%%
speed_sxod=2/dx;% скорость схдимости
N_step=speed_sxod*interval_begin;%количество шагов
```