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 №2

МЕТОДЫ ТОЧЕЧНОГО ОЦЕНИВАНИЯ 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 методы точечного оценивания 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 методов

Метод квадратичной аппроксимации

Начальный этап. Выбрать x1 - начальную точку, Dx - величину шага по оси x. Задать точность поиска по x и f(x) e1=0,01 и e2=0,1. Перейти к основному этапу.

Основной этап. Шаг 1. Вычислить x2 = x1 + Dx.

Шаг 2. Вычислить f(x1), f(x2).

Шаг 3. Если f(x1) > f(x2), положить x3 = x1 + 2Dx. Если f(x1) < f(x2), положить x3 = x1 - Dx.

Шаг 4. Вычислить f(x3) и найти Fмин = min{f1, f2, f3}.

Xмин = (×) xi, которая соответствует Fмин.

Шаг 5. По трем точкам x1, x2, x3 вычислить используя формулу для оценивания с помощью квадратичной аппроксимации.

Шаг 6. Проверка на окончание поиска:

- является ли разность Xмин - достаточно малой(/Xмин - x /<e1)?

- является ли разность Fмин - f(x) достаточно малой (/Fмин - f(x)/<e2)?

Если оба условия выполняются, закончить поиск. В противном случае перейти к шагу 7.

Шаг 7. Выбрать “наилучшую” точку (Xмин или x) и две точки по обе стороны от нее. Обозначить эти точки в естественном порядке и перейти к шагу 4.

Необходимо отметить, что за счет последовательных приближений, совмещенных с квадратичной аппроксимацией, метод имеет высокую эффективность.

Метод Ньютона – Рафсона

Предполагается, что функция $F(x)$ дважды дифференцируема, причем $F''(x) > 0$. Тогда для поиска корня уравнения $F'(x) = 0$ используется метод касательных. Сущность метода заключается в том, что в очередной точке x_k строится линейная аппроксимация функции $F(x)$ (касательная к графику $F(x)$), а точка, в которой линейная аппроксимирующая функция обращается в нуль, используется в качестве следующего приближения x_{k+1}.

Координата точки x_{k+1} находится по формуле

$$x_{k+1} = x_k - \frac{F'(x_k)}{F''(x_k)}, \quad k = 0,1,...,$$

где x_0 - начальная точка выбирается пользователем. Вычисления по приведенной формуле продолжаются до тех пор, пока не выполнится условие
$|F'(x_k)| \le \varepsilon$, после чего полагают $x^* = x_k$, $F^* = F(x^*)$.

Алгоритм поиска точки минимума методом Ньютона

Алгоритм поиска минимума функции методом Ньютона сводится к выполнению следующих этапов.

1 этап. Задается начальный интервал неопределенности $L_0 = [a_0, b_0]$ и $\varepsilon > 0$ - требуемая точность.

2 этап. Задать $k = 0$ и начальную точку $x_k \in [a_k, b_k]$.

3 этап. Вычислить $F'(x_k)$. Проверить условие окончания:

- если $|F'(x_k)| \le \varepsilon$, то процесс поиска завершается и $x^* = x_k$, $F^* = F(x^*)$;

- если $|F'(x_k)| > \varepsilon$, то вычислить $F''(x_k)$ и если $F''(x_k) > 0$ перейти к этапу 4. В противном случае закончить вычисление связи с нарушением обязательного условия $F''(x_k) > 0$.

4 этап. Вычислить $x_{k+1} = x_k - \dfrac{F'(x_k)}{F''(x_k)}$.

5 этап. Принять $k = k + 1$ и перейти к этапу 3.

Примечание. В связи с выбором начального приближения x_0, удаленного достаточно далеко от искомого решения x^*, возможно, что последовательность $\{x_k\}$ будет расходиться. В этом случае рекомендуется найти лучшее начальное приближение x_0 другим методом (метод золотого сечения и т. д.).

Метод средней точки

Метод средней точки направлен на повышение эффективности метода деления отрезка пополам при использовании технологии исключения отрезков за счет замены вычислений функции в трех точках на операцию вычисления производной в средней точке $\tilde{x} = \frac{a+b}{2}$.

Если $F'(\tilde{x}) > 0$, то точка $\tilde{x}$ лежит на участке монотонного возрастания $F(x)$, поэтому $x^* < \tilde{x}$ и точку минимума следует искать на отрезке $[a, \tilde{x}]$.

Если $F'(\tilde{x}) < 0$, то точка $\tilde{x}$ лежит на участке монотонного убывания $F(x)$, поэтому $x^* > \tilde{x}$ и точку минимума следует искать на отрезке $[\tilde{x}, b]$.

Равенство $F'(\tilde{x}) = 0$ означает, что точка минимума найдена точно и $x^* = \tilde{x}$.

Такое исключение отрезков требует на каждой итерации только одного вычисления $F'(\tilde{x})$ и уменьшает отрезок поиска точки минимума ровно в два раза.

Поиск заканчивается, если абсолютная величина производной меньше заданной погрешности.

Алгоритм поиска точки минимума методом средней точки

Алгоритм поиска минимума функции сводится к выполнению следующих этапов.

1 этап. Задается начальный интервал неопределенности $L_0 = [a_0, b_0]$ и $\varepsilon > 0$ - требуемая точность.

2 этап. Задать $k = 0$.

3 этап. Вычислить среднюю точку $\tilde{x} = \frac{a_k + b_k}{2}$, $F'(\tilde{x})$.

4 этап. Проверить условие окончания:

- если $|F'(\tilde{x})| \le \varepsilon$, то процесс поиска завершается и $x^* = \tilde{x}, F^* = F(x^*)$;

- если $|F'(\tilde{x})| > \varepsilon$, то сравнить $F'(\tilde{x})$ с нулем.

Если $F'(\tilde{x}) > 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k = k+1$, $a_k = a_{k-1}, b_k = \tilde{x}_{k-1}$.

Если $F'(\tilde{x}) \le 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k = k+1$, $a_k = \tilde{x}_{k-1}, b_k = b_{k-1}$.

Перейти к этапу 3.

Метод секущих

Метод секущих опирается на равенство $F'(x)=0$, которое является необходимым и достаточным условием глобального минимума выпуклой дифференцируемой функции $F(x)$.

Если на концах отрезка $L=[a,b]$ производная имеет разные знаки, то на интервале (a,b) найдется точка, в которой $F'(x)=0$ и поиск точки минимума $F(x)$ на отрезке $[a,b]$ эквивалентен решению уравнения

$$F'(x)=0, \quad x\in[a,b].$$

Таким образом, любой приближенный метод решения уравнения $F'(x)=0$, $x\in[a,b]$ можно рассматривать как метод минимизации выпуклой дифференцируемой функции $F(x)$ на отрезке $[a,b]$. Одним из таких методов является метод секущих. Он основан на исключении отрезка путем определения точки

$$\tilde{x}=a-\frac{F'(a)}{F'(a)-F'(b)}(a-b)$$

пересечения с осью Ox секущих графика функции $F'(x)$ на очередном отрезке.

.

Отрезок дальнейшего поиска определяется по следующему правилу.

Новыми точками отрезка $[a,b]$ для осуществления следующей итерации являются концы того из отрезков $[a,\tilde{x}]$ и $[\tilde{x},b]$, который содержит точку x^*. Его определяют по знаку производной $F'(\tilde{x})$.

Если $F'(\tilde{x})>0$, то точка $\tilde{x}$ лежит на участке монотонного возрастания $F(x)$, поэтому $x^*<\tilde{x}$ и точку минимума следует искать на отрезке $[a,\tilde{x}]$, то есть $b=\tilde{x}$.

Если $F'(\tilde{x})<0$, то точка $\tilde{x}$ лежит на участке монотонного убывания $F(x)$, поэтому $x^*>\tilde{x}$ и точку минимума следует искать на отрезке $[\tilde{x},b]$, то есть $a=\tilde{x}$.

Равенство $F'(\tilde{x})=0$ означает, что точка минимума найдена точно и $x^*=\tilde{x}$.

На каждой итерации, кроме первой, следует вычислять одно новое значение $F'(x)$.

Поиск заканчивается, если абсолютная величина производной меньше заданной погрешности.

Алгоритм поиска точки минимума методом секущих

Алгоритм поиска минимума функции методом секущих сводится к выполнению следующих этапов.

1 этап. Задается начальный интервал неопределенности $L_0=[a_0,b_0]$ и $\varepsilon>0$ - требуемая точность, $\varepsilon>0$ - малое положительное число.

2 этап. Задать $k=0$. Вычислить $F'(a_k), F'(b_k)$.

Если $F'(a_k)*F'(b_k)<0$, то перейти к этапу 3, иначе к этапу 5.

3 этап. Вычислить $x_k=a_k-\dfrac{F'(a_k)}{F'(a_k)-F'(b_k)}(a_k-b_k)$, $F'(x_k)$.

4 этап. Проверить условие окончания:

- если $\left|F'(x_k)\right| \le \varepsilon$, то процесс поиска завершается и $x^* = x_k$, $F^* = F(x^*)$;

- если $\left|F'(x_k)\right| > \varepsilon$, то сравнить $F'(x_k)$ с нулем.

Если $F'(x_k) > 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k = k+1$, $a_k = a_{k-1}, b_k = x_{k-1}, F'(b_k) = F'(x_{k-1})$.

Если $F'(x_k) \le 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k = k+1$, $a_k = x_{k-1}, b_k = b_{k-1}, F'(a_k) = F'(x_{k-1})$.

Перейти на этап 3.

5 этап. Если $F'(a_k) > 0, F'(b_k) > 0$, то $F(x)$ возрастает на отрезке $L_k = [a_k, b_k]$ и, следовательно, $x^* = a_k$.

Если $F'(a_k) < 0, F'(b_k) < 0$, то $F(x)$ убывает на отрезке $L_k = [a_k, b_k]$ и, следовательно, $x^* = b_k$.

Если $F'(a_k) * F'(b_k) = 0$, то $x^* = a_k$ или $x^* = b_k$, в зависимости от того, на каком из концов отрезка $L_k = [a_k, b_k]$ производная $F'(x) = 0$.

Полученные результаты

Вариант 14

$f(x)= x^2 +12/ x^2 -2.$

Экстремум = 1.9 или -1.9, так как функция симметрична.

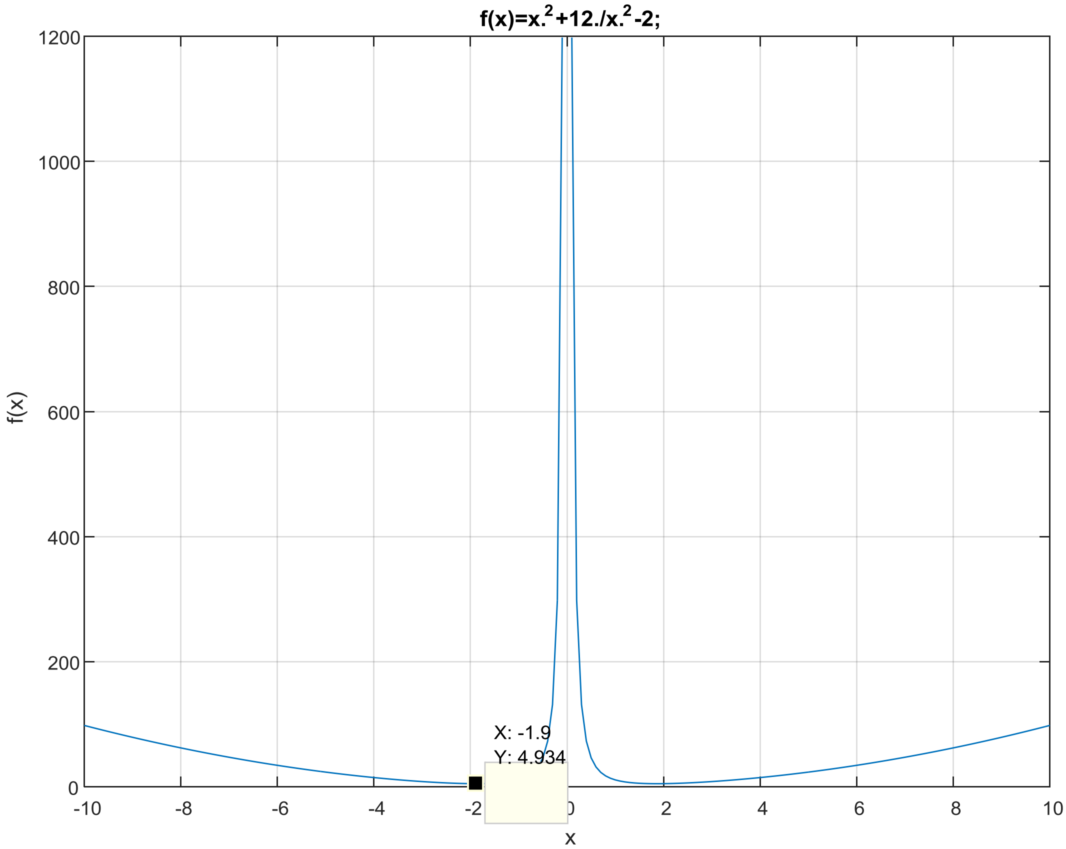

рис.1 - График *f(x)*

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

Метод	*N*			α		
	$\varepsilon=10^{-1}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-3}$	$\varepsilon=10^{-1}$	$\varepsilon=10^{-2}$	$\varepsilon=10^{-3}$
Квадратичной аппроксимаци	6	8	10	0.032334226	0.0220320702	0.01550624124
Ньютона–Рафсона	5	6	7	0.02622180	0.0012227126	0.00020445
Средней точки	8	12	15	0.0078125	0.0004882812	0.000061035156
Секущих	204	646	2227	0.002531825	0.0002561759	0.000827075989

Выводы

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

1. Сравнивая четыре метода оптимизации по параметру количества итераций необходимых для нахождения экстремума, можно сказать, что метод Ньютона–Рафсона дает наилучшие результаты по сравнению с тремя другими методами, так как при одном и том же значении точности епсилон, задача решается с использованием меньшего числа итераций.
2. По скорости сходимости метод квадратичной аппроксимации является наилучшим, также по количеству итераций, необходим для решения задачи практически не уступает первому методу.
3. Метод средней точки, исход из своего название, показал себя средне, как по количеству итераций, так по скорости сходимости .
4. Метод секущих показал себя хуже всех по параметру количества итераций необходимых для решения задачи, однако по скорости сходимости алгоритма при увеличении скорости не сильно увеличивает время на решение задачи.

Список использованной литературы

1. Татарникова Т.М. Методы моделирования и оптимизации: Методические указания к выполнению лабораторных работ. СПб.: ГУАП, 2017
2. http://life-prog.ru/2_60884_algoritm-metoda-pauella.html
3. https://math.semestr.ru/optim/method-powell.php
4. http://www.kti.ru/data/83/m_aprox.html

ПРИЛОЖЕНИЕ

Листинг программ реализующих методы оптимизации реализованных в среде matlab

Метод квадратичной аппроксимации

```
clear all
clc
 %% метод пауэло
eps=0.001;
x1=-1;
deltax=0.1;
x2=x1+deltax;
interval_begin=10+1;
n=1;
while n<1000
    if n==1
    f1 = funcc( x1 );
    f2=  funcc( x2 );

    if f1>f2
        x3=x1+2*deltax;
    else
        x3=x1-deltax;
    end
    f3=funcc( x3 );
    else
        f1 = funcc( x1 );
        f2=  funcc( x2 );
        f3=funcc( x3 );
    end
    fmin=min([f1,f2,f3]);
    [x1,x2,x3,f1,f2,f3 ] = sortich( x1,x2,x3,f1,f2,f3 );
    x_opt  = optimum( x1,x2,x3,f1,f2,f3 );
     f_opt=funcc( x_opt );
    if (fmin-f_opt)/f_opt <=eps
           break;
    else
        x1=x1-deltax;
        x2=x2-deltax;
        x3=x3-deltax;
    end
    n=n+1;
end

interval_end=x3-x_opt;
speed_sxod=interval_end/interval_begin;% скорость сходимсоти
```

Метод Ньютона – Рафсона

```
clc;
clear;

%%метод ньютона
eps=0.001;
x=-1;
n=1;
x1=1;
int_beg=x-x1;

while n<500000
    fx=f(x1);
    fd=fp(x1);
    fd2=fp2(x1);
    if fx<=eps || abs(x-x1)<=eps
        break;
    else
        x=x1;
        x1=x-fd/fd2;
    end



    n=n+1;
end
int_end=x-x1;

speed_sxod=int_end/int_beg;% скорость сходимсоти
```

Метод средней точки

```
clear;
clc;

%%средняя точка
a0=-1;
b0=10;
x=(a0+b0)/2;
a=a0;
b=b0;
eps=0.001;
int_beg=b0-a0;
n=1;
while (b-a)>eps

    fd=fp(x);
    if fd<0
        a=x;
    else
        b=x;
    end
    x=(a+b)/2;

   n=n+1;
end
int_end=b-a;
sped=int_end/int_beg
```

Метод секущих

```
clc;
clear;

%%метод хорд


a=-1;
b=10;
eps=0.001;
n=1;

int_beg=b-a;
while abs(b-a)>eps
    fa=f(a);
    fb=f(b);
    a=b-(b-a)*fb/(fb-fa);
     fa=f(a);
     fb=f(b);
    b=a+(a-b)*fa/(fa-fb);
    n=n+1;
end

int_end=b-a;

sped=abs(int_end/int_beg);
```