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

ЗАДАНИЕ
по лабораторной работе №2
по дисциплине «Методы оптимизации»
Тема: «Методы одномерной оптимизации на основе поиска стационарной точки»

Преподаватель: **Каримов А.И.**

Санкт-Петербург
2019

Цель работы

Изучение среды MATLAB, создание программы для реализации одного из методов одномерного поиска на основе поиска стационарной точки:

- Метод Ньютона;
- Метод секущих;
- Метод Мюллера;
- Обратная параболическая интерполяция;
- Метод Брента-Деккера.

Основные теоретические положения

Критические и стационарные точки функции определяются следующим образом.

Критические точки функции $f(x)$ – точки, в которых производная $f'(x)$ не существует или обращается в нуль.

Стационарные точки функции $f(x)$ – точки, в которых производная $f'(x)$ обращается в нуль.

При этом стационарные точки подрязделяются на:

- экстремумы – точки минимума или максимума;
- седловые точки – точки, в которых производная нулевая, но минимум или максимум не достигается.

Лемма Ферма утверждает: производная $f'(x)$ дифференцируемой функции в точке экстремума равна нулю. В соответствии с этой леммой, возможно использования метода нахождения нуля производной в качестве метода оптимизации. Для этого осуществляются следующие шаги:

1) Поиск $x_i^* : f''(x^*) = 0$

2) Осуществляется проверка: x_i^* – экстремум, если

$$f'''(x^*) \neq 0 \tag{1}$$

или

$$f''(x^* - \epsilon)f''(x^* + \epsilon) < 0, \tag{2}$$

где $\epsilon > 0$ – малое число (взаимозаменяемые условия).

Метод Ньютона

Метод Ньютона минимизации функции является обобщением известного метода Ньютона отыскания корня уравнения

$$f'(x^*) = 0.$$

В качестве приближения x_{k+1} к минимуму x^* берется точка, в которой производная $f'(x) = 0$, т. е. $f'(x_k) + \frac{f''(x_k)}{\Delta x_k} = 0$. Таким образом,

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \tag{3}$$

Метод Ньютона формулируется следующим образом.

Подготовительный этап. Берется начальная точка x_1, устанавливается счетчик итераций $k = 1$.

Основной этап

1) Находится следующее значение x_{k+1} по формуле (3), проверяется критерий окончания поиска В качестве критерия окончания поиска используются следующие сотношения:

$$|x_{k+1} - x_k| < \epsilon$$
$$|f'(x_{k+1}) - f'(x_k)| < \delta$$

2) Если поиск не окончен, устанавливается новое значение счетчика $k := k + 1$, иначе – проверяются критерии (1) или (2), и если они удовлетвореня, найденное значение x_{k+1} является искомым экстремумом.

Метод секущих

Метод секущих предлагает заменить вторую производную $f''(x_k)$ в ньютоновской формуле её линейной аппроксимацией $(f'(x_k) - f'(x_{k-1}))/(x_k - x_{k-1})$. Тем самым,

$$x_{k+1} = x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k\smile 1})}.$$

Легко видеть, что x_{k+1} – точка пересечения с осью абсцисс секущей прямой, проходящей через точки x_k и x_{k-1}.

Псевдокод

Цикл

$x_{k+1} = b_k - f'(b_k)(b_k - a_k)/(f'(b_k) - f'(a_k))$;

Если

  $|f'(x_{k+1})| \le \epsilon$, //КОП

то

  остановиться

иначе // Уменьшить интервал поиска минимума

  Если

    $f'(x_{k+1}) > 0$,

  то

    $a_{k+1} = a_k$, $b_{k+1} = x_{k+1}$,

  иначе

    $a_{k+1} = x_{k+1}$, $b_{k+1} = b_k$;

$k = k + 1$;

Пока не выполнен КОП

Метод Мюллера

Метод Мюллера основан на методе секущих. Основная идея состоит в том, чтобы использовать не две, а три опорные точки, и строить секущие параболы, а не прямые. В качестве следующего приближения берется точка пересечения параболы и оси x.

Найдем параболу через три точки:

$$y = f(x_k) + w(x - x_k) + f[x_k, x_{k-1}, x_{k-2}](x - x_k)^2, \tag{4}$$

где

$$w = f[x_k, x_{k-1}] + f[x_k, x_{k-2}] - f[x_{k-1}, x_{k-2}]. \tag{5}$$

Тогда очередная точка ищется по формуле

$$x_{k+1} = x_k - \frac{2f(x_k)}{w \pm \sqrt{w^2 - 4f(x_k)f[x_k, x_{k-1}, x_{k-2}]}}. \tag{6}$$

Все остальное в методе Мюллера аналогично методу секущих. **Разделенные разности** первого и второго порядка в уравнениях (4)–(6) находятся по следующим формулам. Разделенная разность первого порядка:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Разделенная разность второго порядка:

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Метод обратной параболической интерполяции

Ищет стационарную точку по формуле

$$x_{n+1} = \frac{f'_{n-1}f'_n}{(f'_{n_2} - f'_{n-1})(f'_{n-2} - f'_n)}x_{n-2} + \\ + \frac{f'_{n-2}f'_n}{(f'_{n-1} - f'_{n-2})(f'_{n-1} - f'n)}x_{n-1} + \frac{f'_{n-2}f'_{n-1}}{(f'_n - f'n - 2)(f'_n - f'_{n-1})}x_n$$

В остальном – метод аналогичен методу Ньютона

Метод Брента-Деккера

Метод Брента-Деккера объединяет метод средней точки, секущих и обратной квадратичной интерполяции для поиска нуля функции. Его реализация на псевдокоде представлена ниже.

Псевдокод

input a, b, and (a pointer to) a function for f
  calculate $f(a)$
  calculate $f(b)$
  if $f(a)f(b) \geq 0$ **then**
    exit function because the root is not bracketed.
  end if
  if $|f(a)| < |f(b)|$
  then
    swap (a, b)
  end if
  $c := a$
  set mflag
  repeat until $f(bors) = 0$ or $|b - a| < \epsilon$
    if $f(a) \neq f(c)$ and $f(b) \neq f(c)$ **then**
      (inverse quadratic interpolation)
      $s := \frac{af(b)f(c)}{(f(a)-f(b))(f(a)-f(c))} + \frac{bf(a)f(c)}{(f(b)-f(a))(f(b)-f(c))} + \frac{cf(a)f(b)}{(f(c)-f(a))(f(c)-f(b))}$
    else
      (secant method)

$s := b - f(b)\frac{b-a}{f(b)-f(a)}$
end if
if (condition 1) s is not between $(3a+b)/4$ and b or
(condition 2) (mflag is set and $|s-b| \geq |b-c|/2$) or
(condition 3) (mflag is cleared and $|s-b| \geq |c-d|/2$) or
(condition 4) (mflag is set and $|b-c| < \delta$) or
(condition 5) (mflag is cleared and $|c-d| < \delta$) **then**
(bisection method)
$s := \frac{a+b}{2}$
set mflag
else
clear mflag
end if
calculate $f(s)$
$d := c$
$c := b$
if $f(a)f(s) < 0$ **then**
$b := s$
else
$a := s$
end if
if $|f(a)| < |f(b)|$ **then**
swap (a, b)
end if
end repeat
output b or s (return the root)

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`. Для реализации методов понадобятся функции среды, аналогичные функциям, применяемым в лабораторной работе 1. Также потребуется функция экспорта растрового изображения `export_fig`, которая устанавливается одноименным add-on'ом в соответствующей вкладке среды (для этого необходимо войти в аккаунт на официальном сайте Mathworks).

Пример правильного вызова команды `export_fig` следующий:

```
export_fig(gcf,'1.jpg','-transparent','-r300')
```

Здесь `gcf` отвечает за выбор текущего окна графика, `'1.jpg'` – название файла, `'-transparent'` – выбор прозрачного фона (белого для форматов типа *.jpg, не поддерживающих прозрачных цветов), `'-r300'` – разрешение в 300 dpi, соответствующее заданному качеству печати.

Также понадобится команда

```
num2str(x)
```

Эта команда преобразует переменную `x` в текст – массив символов типа `char`.

Помимо реализации самих методов, в данной работе потребуется реализовать также и визуализацию процесса работы оптимизации. Пример соответствующего когда показан в листинге 1.

Листинг 1: Метод золотого сечения с визуализацией

```
function [xmin, fmin, neval] = goldensectionsearch2slides(f,
    interval,tol)
% goldensectionsearchvisual searches minimum using golden section
% [xmin, fmin, neval] = goldensectionsearch(f,interval,tol)
% f - an objective function handle
% interval = [a, b] - search interval
% tol - set for bot range and function value
    a = interval(1);
    b = interval(2);
    deltainterval = realmax; %value greater then tol
    L = b - a;
    neval = 0;
    Fi = (1 + sqrt(5))/2;

    % VISUALIZATION
    ctr = 1;
    deltaX = (b-a)/100;
    figure(3); hold on
    [miny, maxy] = drawplot(f,a,b,a,b);
    deltaY = abs(maxy - miny)/100;
    placelabel(a,0,deltaX,deltaY,ctr);
    placelabel(b,0,deltaX,deltaY,ctr);
    export_fig(gcf,'1.jpg','-transparent','-r300');

    % MAIN LOOP
    while L > tol
        L = b - a;

        x1 = b - L/Fi;
        x2 = a + L/Fi;

        y1 = feval(f,x1);
        y2 = feval(f,x2);
        ctr = ctr + 1;

        if ctr < 10
            drawplot(f,a,b,x1,x2);
        end

        if y1 > y2
            a = x1;
            placelabel(x1,0,deltaX,deltaY,ctr);
            xmin = x2;
            fmin = y2;
        else
            b = x2;
            placelabel(x2,0,deltaX,deltaY,ctr);
            xmin = x1;
            fmin = y1;
        end
```

```

        neval = neval + 2;

        if ctr <= 10
           export_fig(gcf,[num2str(ctr),'.jpg'],'-transparent','-r300');
        end
    end
end
```

В Листинге 2 показана реализация используемой в коде функции `drawplot`.

Листинг 2: Функция рисования графика

```
function [miny maxy] = drawplot(f,a,b,x1,x2)
    figure(3);
    h = (b-a)/100;
    x = a:h:b;
    y = feval(f,x);

    miny = min(y);
    maxy = max(y);

    colp = hsv2rgb([rand(), 1, 0.5+0.5*rand()]);
    plot(x,y,'LineWidth',1,'Color',colp);
    scatter([a b],[feval(f,a), feval(f,b)],'Marker','o','MarkerFaceColor',colp,'MarkerEdgeColor',colp);
    xlabel('\itx');
    ylabel('\ity');
    line([a b],[0 0],'Color','k','LineWidth',1); %axis x
    col = hsv2rgb([rand(), 1, 0.5+0.5*rand()]);
    y1 = feval(f,x1);
    line([x1 x1],[0 y1],'Marker','s','Color',col,'LineWidth',1,'MarkerSize',4);
    y2 = feval(f,x2);
    line([x2 x2],[0 y2],'Marker','s','Color',col,'LineWidth',1,'MarkerSize',4);
end
```

В Листинге 3 показана реализация дополнительной функции `placelabel`.

Листинг 3: Функция установки меток с нумерацией итераций

```
function placelabel(x,y,deltaX,deltaY,iternumber)
if iternumber <=10
    text(x - deltaX/2,y + 4*deltaY,num2str(iternumber));
end
end
```

Проверьте, как работает команда `help`, введя в командную строку

`>>help goldensectionsearch2slides`

Программа нарисует серию из 10 кадров, озаглавленных от “1.jpg” до “10.jpg”. Пример кадра приведен на рисунке 1.

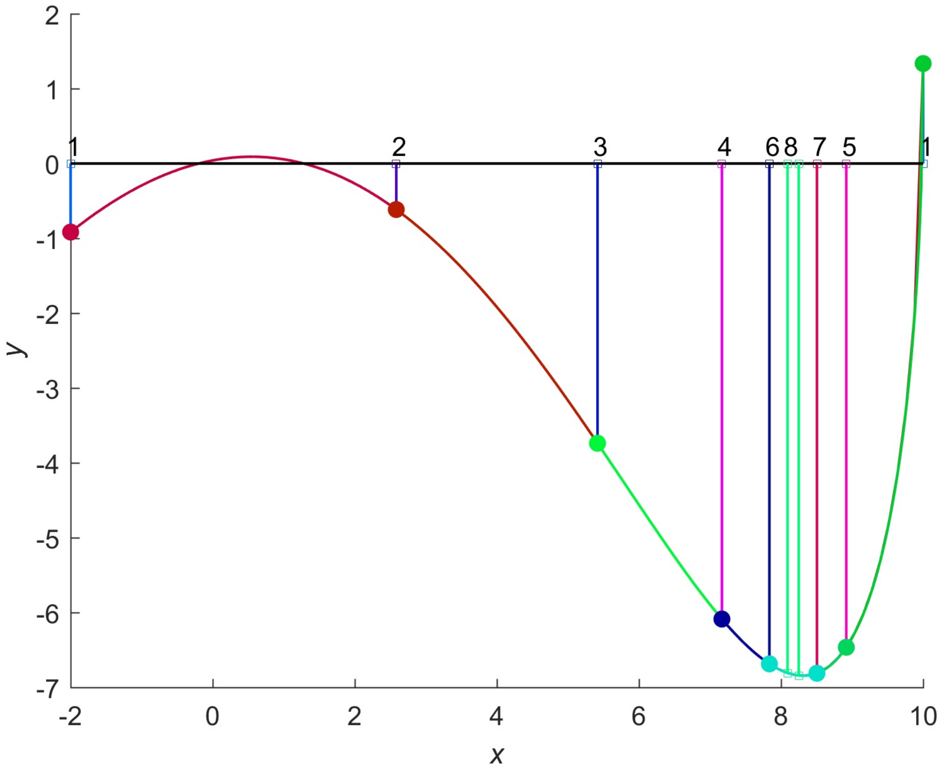

Рис. 1: Кадр номер 8, полученный кодом `goldensectionsearch2slides`

Содержание работы и отчета

Лабораторная работа должна включать в себя *.m-файлы кодов на MATLAB для решения задачи и от 5 до 10 файлов изображений со стадиями выполнения каждого из заданных в варианте методов в формате *.jpg (количество файлов зависит от того, насколько читаемым является последнее изображение). Тип и формат визуализации может выбираться по усмотрению студента в зависимости от метода оптимизации.

Отчет должен содержать титульный лист и разделы:

- Цель работы
- Основные теоретические положения.
- Коды для решения задачи оптимизации.
- Реализация отрисовки изображений для двух заданных методов и заданной целевой функции (в тексте приводится к качестве примера по одному изображению для каждого метода).
- Выводы.

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

```
f %objective function (2)
goldensectionsearch2slides %golden section search with
   visualization implementation
test_goldensectionsearch2slides %test file for
   goldensectionsearch2slides
```

В Таблице 1 приведены тестовые функции. Обратите внимание, что функция 1 неунимодальна!

Номер	Функция	Точка x^*
1	$y = x^2 - 10\cos(0.5\pi x) - 110$	0
2	$y = (x-3)^2 - 50$	3

Таблица 1: Тестовые функции

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

Таблица 2: Варианты

Таблица 2 содержит варианты заданий по вариантам. Номерами методов обозначены: 1 – Больцано, 2 – трехточечного деления, 3 – золотого сечения, 4 – Фибоначчи, 5 – Ньютона, 6 – секущих, 7 – Мюллера, 8 – обратная параболическая интерполяция, 9 – Брента-Деккера.