

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Методы оптимизации»

Студент: М. А. Бронников
Преподаватель: Т. И. Короткова
Группа: М8О-307Б
Дата:
Оценка:
Подпись:

Москва, 2020

1 Метод конфигураций

- **Класс задач:** безусловная оптимизация
- **Формулировка задачи:** Найти минимум функции 2-ух переменных $f(x_1, x_2)$ методом конфигурации.
- **Что вычисляется в процессе решения:** В процессе итерации находятся точки базиса, в которых происходит исследующий поиск и поиск по образцу.
- **Алгоритм:** Описание алгоритма:

1. *Инициализация значений:* Выбор начальной базисной точки $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$, начальных покоординатных приращений $dx_1^0, dx_2^0, \dots, dx_n^0$, а также минимальной длины шага ξ для останова и необязательный параметр останова - максимальное количество итераций N_{max} . Для корректной работы алгоритма $dx_i^0 > \xi$ для $\forall i \in \{1, \dots, n\}$. Начинаем первую итерацию при $j = 0$.
2. *Исследующий поиск:* Для базисной точки x^j , где j - будем называть номером итерации алгоритма, выполним:

- (a) Проверим, если выполняется условие $dx_i^j < \xi$, значит выполнен основной критерий останова и выходим из алгоритма с результатом: x^j .
- (b) Для $\forall i \in \{1, \dots, n\}$:

$$x_i^{j+1} = \begin{cases} x_i^j + dx_i^j, & f(x_1^j, \dots, x_i^j + dx_i^j, \dots, x_n^j) < f(x_1^j, \dots, x_i^j, \dots, x_n^j) \\ x_i^j - dx_i^j, & f(x_1^j, \dots, x_i^j - dx_i^j, \dots, x_n^j) < \min(f(x_1^j, \dots, x_i^j, \dots, x_n^j), f(x_1^j, \dots, x_i^j + dx_i^j, \dots, x_n^j)) \\ x_i^j, & \min(f(x_1^j, \dots, x_i^j - dx_i^j, \dots, x_n^j), f(x_1^j, \dots, x_i^j + dx_i^j, \dots, x_n^j)) \geq f(x_1^j, \dots, x_i^j, \dots, x_n^j) \end{cases}$$

- (c) Если $x^{j+1} = x^j$, значит уменьшаем значения приращений $dx_1^j, dx_2^j, \dots, dx_n^j$ (например, $dx_i^j = \frac{dx_i^j}{10}$ для $\forall i \in \{1, \dots, n\}$) и перейти к шагу (2).

- (d) Если $x^{j+1} \neq x^j$, значит получаем новую базисную точку x^{j+1} с сохранением приращений $dx_i^{j+1} = dx_i^j$ для $\forall i \in \{1, \dots, n\}$, увеличим $j = j + 1$ и перейдём к шагу (3).

3. *Поиск по образцу:* Для новой базисной точки x^j выполним:

- (a) Проверим, если выполняется условие достижения максимального количества итераций $j = N_{max}$, значит выходим из алгоритма с результатом: x^j .
- (b) Делаем шаг поиска и вычисляем $x^e = x^j + t_j(x^j - x^{j-1})$, где t_j - параметр движения (зависит от реализаций, обычно $t_j = 2$ для $\forall j \in N$).

(с) Вычислим точку x^y . Для $\forall i \in \{1, \dots, n\}$:

$$x_i^y = \begin{cases} x_i^e + dx_i^j, & f(x_1^e, \dots, x_i^e + dx_i^j, \dots, x_n^e) < f(x_1^e, \dots, x_i^e, \dots, x_n^e) \\ x_i^e - dx_i^j, & f(x_1^e, \dots, x_i^e - dx_i^j, \dots, x_n^e) < \min(f(x_1^e, \dots, x_i^e, \dots, x_n^e), f(x_1^e, \dots, x_i^e + dx_i^j, \dots, x_n^e)) \\ x_i^e, & \min(f(x_1^e, \dots, x_i^e - dx_i^j, \dots, x_n^e), f(x_1^e, \dots, x_i^e + dx_i^j, \dots, x_n^e)) \geq f(x_1^e, \dots, x_i^e, \dots, x_n^e) \end{cases}$$

(d) Если $x^e = x^y$, значит поиск по образцу не удался. Переходим к шагу (2).

(е) Если $x^e \neq x^y$, значит получаем новую базисную точку $x^{j+1} = x^y$ с сохранением приращений $dx_i^{j+1} = dx_i^j$ для $\forall i \in \{1, \dots, n\}$, увеличим $j = j + 1$ и перейдём к шагу (3).

• **1-ая итерация:**

$$f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + x_2 + 9$$

$$\xi = 0.01$$

$$N_{max} = 8$$

$$j = 0$$

$$x_1^0 = 11, x_2^0 = 7$$

$$dx_1^0 = 1, dx_2^0 = 1$$

Делаем исследующий поиск:

$$dx_1^0 = 1 > 0.01 = \xi$$

$$dx_2^0 = 1 > 0.01 = \xi$$

$j < N_{max}$, значит критерий останова не выполнен.

$$f(x_1^j, x_2^j) = 5 \cdot 121 + 3 \cdot 77 + 6 \cdot 49 + 22 + 7 + 9 = 1168$$

1. Для x_1 :

$$f(x_1^j + 1, x_2^j) = 5 \cdot 144 + 3 \cdot 84 + 6 \cdot 49 + 24 + 7 + 9 = 1306 > f(x_1^j, x_2^j)$$

$$f(x_1^j - 1, x_2^j) = 5 \cdot 100 + 3 \cdot 70 + 6 \cdot 49 + 20 + 7 + 9 = 1040 < f(x_1^j, x_2^j)$$

$$\text{Значит } x_1^{j+1} = x_1^j - 1 = 11 - 1 = 10$$

2. Для x_2 :

$$f(x_1^j, x_2^j + 1) = 5 \cdot 121 + 3 \cdot 88 + 6 \cdot 64 + 22 + 8 + 9 = 1292 > f(x_1^j, x_2^j)$$

$$f(x_1^j, x_2^j - 1) = 5 \cdot 121 + 3 \cdot 66 + 6 \cdot 36 + 22 + 6 + 9 = 1056 < f(x_1^j, x_2^j)$$

$$\text{Значит, } x_2^{j+1} = x_2^j - 1 = 7 - 1 = 6$$

$$dx_1^{j+1} = dx_1^j = 1, dx_2^{j+1} = dx_2^j = 1$$

$$j = j + 1 = 1$$

$$x^1 = (10, 6)$$

Конец первой итерации.

- Результат компьютерных вычислений:

Расчет окончен

Протокол расчета

Выполнил: Бронников, группа 80-307, 10.03.2020

Квадратичная функция: $f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + 1x_2 + 9$

Метод конфигураций

Точность метода: 0.01, $N_{\max} = 8$, Количество итераций: 9

$N_{\text{ит}}$	x_1	x_2	$f(x_1, x_2)$	dx_1	dx_2	коэф-т k
0	11	7	1168	1	1	
1	10	6	931	1	1	0
2	9	5	722	1	1	0
3	8	4	541	1	1	0
4	7	3	388	1	1	0
5	6	2	263	1	1	0
6	5	1	166	1	1	0
7	4	0	97	1	1	0
8	3	-1	56	1	1	0
9	2	-1	32			

Критерий окончания не выполнен

$$\|x - x^*\| = 2.39202$$

$$|f(x) - f(x^*)| = 23.20721$$

2 Метод Марквардта

- **Класс задач:** безусловная оптимизация
- **Формулировка задачи:** Найти минимум функции 2-ух переменных $f(x_1, x_2)$ методом Марквардта.
- **Что вычисляется в процессе решения:** В процессе итерации для поиска точки x^{k+1} с меньшим значением функции $f(x)$ вычисляются шаг-приращение dx^k для точки x^k при помощи градиента и матрицы Гессе в этой точке.
- **Алгоритм:** Перед началом описания алгоритма следует ввести следующие обозначения:
Матрица Гессе:

$$H_f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$$

Градиент:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

Функции $f(x)$, где $x = (x_1, x_2, \dots, x_n)$.

Описание алгоритма:

1. *Начальная инициализация:* Задать точку для начала движения $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$, точность приближения ξ , необязательный параметр - максимальное количество итераций N_{max} , а также выбрать значение параметра λ_0 , которое, впрочем, можно задать как $\lambda_0 = 50 \cdot \max(h_{ij}^0)_{0 \leq i, j \leq n}$, где h_{ij}^k - элемент матрицы $H_f(x^k)$ на i -ой строке и j -ом столбце. Начиннаем первую итерацию при $k = 0$.
2. *Проверим критерий останова:* Если выполнено основное условие $\|\nabla f(x^k)\| < \xi$ или дополнительное $k = N_{max}$, где k назовем номером итерации, выходим из алгоритма с результатом x^k .

3. *Вычисление новой точки.* Вычислим приращение $dx^k = -[H_f(x^k) + \lambda_k E]^{-1} \nabla f(x^k)$ и новую точку $x^{k+1} = x^k + dx^k$.

- Если $f(x^{k+1}) < f(x^k)$, значит мы успешно нашли новую точку и можем уменьшить значение параметра λ_{k+1} , например $\lambda_{k+1} = \frac{\lambda_k}{2}$. Положим $k = k + 1$ и перейдем к шагу (2).
- Если $f(x^{k+1}) \geq f(x^k)$, значит поиск неудачен и следует увеличить значение параметра λ_k , например $\lambda_k = \lambda_k \cdot 2$. Параметр k не меняется и повторим шаг (3) для поиска x^{k+1} с новым параметром λ_k .

• **1-ая итерация:**

$$f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + x_2 + 9$$

$$\xi = 0.01$$

$$N_{max} = 5 \quad k = 0$$

$$x_1^0 = 11, \quad x_2^0 = 7$$

$$\nabla f(x) = \begin{pmatrix} 10x_1 + 3x_2 + 2 \\ 12x_2 + 3x_1 + 1 \end{pmatrix}$$

$$H_f(x) = \begin{pmatrix} 10 & 3 \\ 3 & 12 \end{pmatrix}$$

Матрица Гессе положительно определена, поэтому для ускорения сходимости возьмем небольшой параметр $\lambda_0 = 1$, хотя в общем случае рекомендуется брать значение на порядок больше чем элементы в матрице Гессе.

Проверим критерий останова:

$$\|\nabla f(x^k)\| = \left\| \begin{pmatrix} 133 \\ 118 \end{pmatrix} \right\| = \sqrt{133^2 + 118^2} = 177.8004 > 0.01 = \xi$$

$k = 0 < 5 = N_{max}$, значит критерий останова не выполнен.

Вычислим следующую точку:

$$dx^k = -[H_f(x^k) + \lambda_k E]^{-1} \nabla f(x^k) = \left[\begin{pmatrix} 10 & 3 \\ 3 & 12 \end{pmatrix} + \begin{pmatrix} \lambda_k & 0 \\ 0 & \lambda_k \end{pmatrix} \right]^{-1} \begin{pmatrix} 133 \\ 118 \end{pmatrix} = (-10.261, -6.709)$$

$$x^{k+1} = x^k + dx^k = (11, 7) + (-10.261, -6.709) = (0.739, 0.291)$$

$$f(x^{k+1}) = 12.2598 < 1168 = f(x^k), \text{ значит новая точка найдена успешно.}$$

$$\lambda_{k+1} = \frac{\lambda_k}{2} = 0.5$$

$$k = k + 1$$

Конец первой итерации.

- Результат компьютерных вычислений:

Расчет окончен

Сохранить протокол

Выбрать другой метод

Выход

Протокол расчета

Выполнил: Бронников, группа 80-307, 10.03.2020

Квадратичная функция: $f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + 1x_2 + 9$

Метод Марквардта

Точность метода: 0.01, $N_{\max} = 5$, Количество итераций: 4

$N_{\text{ит}}$	шаг μ	x_1	x_2	$f(x_1, x_2)$	f'_{x_1}	f'_{x_2}	$\ \nabla f(x_1, x_2)\ $
0	0	11	7	1168	133	118	177.80045
1	0	0.73881	0.29104	14.65115	10.26119	6.70896	12.25978
2	0	-0.10648	-0.02996	8.82872	0.84529	0.32101	0.90419
3	0	-0.1813	-0.03739	8.79308	0.07482	0.00743	0.07519
4	0	-0.18839	-0.03632	8.7928	0.00709	-0.00107	0.00717

Критерий окончания выполнен

$$\|x - x^*\| = 0.00085$$

$$|f(x) - f(x^*)| = 0$$

3 Метод Нельдера-Мида

- **Класс задач:** безусловная оптимизация
- **Формулировка задачи:** Найти минимум функции 2-ух переменных $f(x_1, x_2)$ методом Нельдера-Мида.
- **Что вычисляется в процессе решения:** В процессе итераций вычисляется центр тяжести набора точек, с целью заменить «худшую» точку из набора на более близкую к минимуму функции.
- **Алгоритм:** Минимизируем функцию $f(x)$, где $x = (x_1, x_2, \dots, x_n)$
Описание алгоритма:

1. *Начальная инициализация:* Задать начальную систему из $n + 1$ точек: $\{x^{0(1)}, x^{0(2)}, \dots, x^{0(n+1)}\}$, точность ξ , максимальное количество итераций N_{max} , а также параметр отражения $\alpha > 0$ (рекомендуется $\alpha = 1$), параметр сжатия $\beta > 0$ (рекомендуется $\beta \in [0.4, 0.6]$) и параметр растяжения $\gamma > 0$ (рекомендуется $\gamma \in [2, 3]$). Начиннаем первую итерацию при $k = 0$.
2. *Выбор трёх точек:* Выберем три точки x_h, x_l, x_g из набора, таких что $f(x_h) \geq f(x_g) \geq f(x_i)$, для $\forall x_i \in \{x^{k(1)}, x^{k(2)}, \dots, x^{k(n+1)}\} \setminus \{x_h, x_g\}$ и $f(x_l) \leq f(x_i)$ для $\forall x_i \in \{x^{k(1)}, x^{k(2)}, \dots, x^{k(n+1)}\}$.

3. *Проверим критерий останова:* Обозначим $\bar{f} = \frac{\sum_{j=1}^{n+1} f(x^{k(j)})}{n+1}$.

Если $\sqrt{\frac{\sum_{i=1}^{n+1} |f(x^{k(i)}) - \bar{f}|^2}{n+1}} < \xi$ или $k = N_{max}$, то выходим из алгоритма с значением x_l , иначе $k = k + 1$.

4. *Найдем центр тяжести:* Найдем центр тяжести $x_c = \frac{(\sum_{i=1}^{n+1} x^{k(i)} - x_h)}{n}$
5. *Отражение точки:* Получим точку отражения $x_r = (1 + \alpha)x_c - \alpha x_h$.
 - Если $f(x_r) < f(x_l)$, значит переходим к шагу (6).
 - Если $f(x_l) < f(x_r) < f(x_g)$, значит $x_h = x_r$, переходим на шаг (2).
 - Если $f(x_g) < f(x_r) < f(x_h)$, значит x_h меняют значения местами x_r , переходим на шаг (7).
 - Если $f(x_h) < f(x_r)$, перейдем на шаг (7).
6. *Растяжение точки:* Получим точку растяжения $x_e = (1 - \gamma)x_c + \gamma x_r$.

– Если $f(x_e) < f(x_r)$, тогда $x_h = x_c$ и переходим на шаг (2).

– Если $f(x_e) > f(x_r)$, тогда $x_h = x_r$ и переходим на шаг (2).

7. *Сжатие точки*: Получим точку сжатия $x_s = \beta x_h + (1 - \beta)x_c$.

– Если $f(x_s) < f(x_h)$, тогда $x_h = x_s$ и переходим к шагу (2).

– Если $f(x_s) > f(x_h)$, тогда выполняем редукцию: $x_i = x_l + \frac{(x_i - x_l)}{2}$ и переходим к шагу (2)

• **1-ая итерация:**

$$f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + x_2 + 9$$

$$\xi = 0.01$$

$$N_{max} = 5 \quad k = 0$$

$$x_1^0 = 11, \quad x_2^0 = 7$$

$$\alpha = 1, \quad \gamma = 2.8, \quad \beta = 0.5$$

$$x^{0(1)} = (-3, 3), \quad x^{0(2)} = (-3, -2), \quad x^{0(3)} = (11, 7)$$

Выберем 3 точки:

$$x_l = (-3, 3), \quad x_h = (11, 7), \quad x_g = (-3, -2)$$

Найдем центр тяжести:

$$x_c = \frac{x_l + x_g}{2} = \left(\frac{-3 - 3}{2}, \frac{3 - 2}{2} \right) = (-3, 0.5)$$

Отражение точки:

$$x_r = (1 + \alpha)x_c - \alpha x_h = (-6, 1) - (11, 7) = (-17, -6)$$

$$f(x_r) = 1936$$

$$f(x_r) > 1168 = f(x_h), \text{ значит делаем сжатие.}$$

Сжатие точки:

$$x_s = \beta x_h + (1 - \beta)x_c = (5.5, 3.5) + (-1.5, 0.25) = (4, 3.75)$$

$$f(x_s) = 230.125$$

$$f(x_s) < 1168 = f(x_h), \text{ значит заменяем точку } x^{k(3)} = x_h \text{ в наборе на точку } x_s = (4, 3.75).$$

Проверим критерий окончания:

$$f(x^{k(1)}) = 78, \quad f(x^{k(2)}) = 88, \quad f(x^{k(3)}) = 230.125$$

$$\bar{f} = \frac{78 + 88 + 230.125}{3} = 132.041667$$

$$\sqrt{\frac{\sum_{i=1}^3 |f(x^{k(i)}) - \bar{f}|^2}{n+1}} = 69.4754403 > 0.01 = \xi$$

$k = 0 < 5 = N_{max}$
 $k = k + 1$

Конец первой итерации.

• **Результат компьютерных вычислений:**

Расчет окончен

Протокол расчета

Выполнил: Бронников, группа 80-307, 10.03.2020

Квадратичная функция: $f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + 1x_2 + 9$

Метод Нелдера-Мида

Точность метода: 0.01, $N_{max} = 5$, Количество итераций: 6

$N_{ит}$	α	операция	коэффициент	x_1	x_2	$f(x_1, x_2)$
0	1	редукция		-3	3	78
				-3	-2	88
				11	7	1168
1	1	редукция		-3	0.5	45.5
				-3	3	78
				4	5	312
2	1	редукция		-3	0.5	45.5
				-3	1.75	52.375
				0.5	2.75	63.5
3	1	растяжение	2.8	-1.25	1.625	25.6875
				-3	0.5	45.5
				-3	1.125	46.59375
4	1	редукция		0.325	0.8875	16.65688
				-1.25	1.625	25.6875
				-3	0.5	45.5
5	1	растяжение	2.8	-1.3375	0.69375	16.06734
				0.325	0.8875	16.65688
				-0.4625	1.25625	18.12672
6				-0.55	0.325	9.835
				-1.3375	0.69375	16.06734
				0.325	0.8875	16.65688

Критерий окончания не выполнен

$$\|x - x^*\| = 0.51042$$

$$|f(x) - f(x^*)| = 1.04221$$

4 Метод сопряженных градиентов

- **Класс задач:** безусловная оптимизация
- **Формулировка задачи:** Найти минимум функции 2-ух переменных $f(x_1, x_2)$ методом сопряженных градиентов.
- **Что вычисляется в процессе решения:** В процессе итерации для поиска точки x^{k+1} с меньшим значением функции $f(x)$ вычисляются длина шага t_k и его направление dx^k для точки x^k при помощи градиента, направления dx^{k-1} с предыдущей итерации и метода дихотомии.
- **Алгоритм:** Перед началом описания алгоритма следует ввести следующие обозначения:

Градиент:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

Функции $f(x)$, где $x = (x_1, x_2, \dots, x_n)$.

Описание алгоритма:

1. *Начальная инициализация:* Задать точку для начала движения $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$, точность приближения ξ , необязательный параметр - максимальное количество итераций N_{max} , а также выбрать значение параметра точности поиска ϵ и отрезка $[a, b]$ для расчета длин шагов. Начинаем первую итерацию при $k = 0$ и $dx^0 = -\nabla f(x^0)$.
2. *Вычислим длину шага:* Найдем длину шага $t_k = \underset{t \in [a, b]}{\operatorname{argmin}} |f(x^k + t \cdot dx^k)|$, поиск значения которой производится методом дихотомии на заданном отрезке $[a, b]$ с точностью ϵ .
3. *Вычисление следующей точки:* Найдем новую точку $x^{k+1} = x^k + t_k \cdot dx^k$ и инкрементируем счетчик $k = k + 1$.
4. *Проверим критерий останова:* Если выполнено основное условие $\|\nabla f(x^k)\| < \xi$ или дополнительное $k = N_{max}$, где k - номер итерации, выходим из алгоритма с результатом x^k .

5. *Вычислим направление поиска:* Найдем новое направление движения

$dx^k = -\nabla f(x^k) + \beta_{k-1} dx^{k-1}$, где коэффициент $\beta_{k-1} = \frac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2}$ и перейдем к шагу (2).

• **1-ая итерация:**

$$f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + x_2 + 9$$

$$\xi = 0.01$$

$$\epsilon = 0.001$$

$$[a, b] = [0.05, 0.5]$$

$$N_{max} = 5$$

$$k = 0$$

$$x_1^0 = 11, x_2^0 = 7$$

Градиент функции рассчитывается по формуле:

$$\nabla f(x) = \begin{pmatrix} 10x_1 + 3x_2 + 2 \\ 12x_2 + 3x_1 + 1 \end{pmatrix}$$

Найдем направление движения:

Определим начальное направление движения $dx^0 = -\nabla f(x^k) = (-133, -118)$.

Найдем длину шага:

Методом дихотомии с точностью ϵ на отрезке $[a, b]$ определили оптимальное значение как $t_k = t_0 = 0.0722$.

Вычислим следующую точку:

Найдем новое значение точки:

$$x^{k+1} = x^k + t_k \cdot dx^k = (11, 7) + 0.0722 \cdot (-133, -118) = (1.4021, -1.5155)$$

И увеличим значение $k = k + 1$

Проверим критерий останова:

$$\|\nabla f(x^k)\| = \left\| \begin{pmatrix} 11.4743 \\ -12.9793 \end{pmatrix} \right\| = \sqrt{(11.4743)^2 + (-12.9793)^2} = 17.3240 > 0.01 = \xi$$

$k = 1 < 5 = N_{max}$, значит критерий останова не выполнен.

Конец первой итерации.

- Результат компьютерных вычислений:

Расчет окончен

Протокол расчета

Выполнил: Бронников, группа 80-307, 10.03.2020

Квадратичная функция: $f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + 1x_2 + 9$

Метод сопряженных градиентов

Точность метода: 0.01, $N_{\max} = 2$, Количество итераций: 3

$N_{\text{ит}}$	шаг t	x_1	x_2	$f(x_1, x_2)$	f'_{x_1}	f'_{x_2}	$\ \nabla f(x_1, x_2)\ $
0	0.07216	11	7	1168	133	118	177.80045
1	0.12484	1.40207	-1.51546	27.52301	11.47428	-12.97932	17.32403
2	0.0709	-0.18807	-0.03492	8.79281	0.01456	0.01674	0.02218
3	0	-0.1891	-0.03611	8.79279	0.00066	-0.00059	0.00089

Критерий окончания не выполнен

$$\|x - x^*\| = 0.00011$$

$$|f(x) - f(x^*)| = 0$$

5 Метод Ньютона-Рафсона

- **Класс задач:** безусловная оптимизация
- **Формулировка задачи:** Найти минимум функции 2-ух переменных $f(x_1, x_2)$ методом Ньютона-Рафсона.
- **Что вычисляется в процессе решения:** В процессе итерации для поиска точки x^{k+1} с меньшим значением функции $f(x)$ вычисляются длина t_k и направление dx^k шага для точки x^k при помощи градиента и матрицы Гессе в этой точке.
- **Алгоритм:** Перед началом описания алгоритма следует ввести следующие обозначения:
Матрица Гессе:

$$H_f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$$

Градиент:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

Функции $f(x)$, где $x = (x_1, x_2, \dots, x_n)$.

Описание алгоритма:

1. *Начальная инициализация:* Задать точку для начала движения $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$, точность приближения ξ , необязательный параметр - максимальное количество итераций N_{max} , а также выбрать значение параметра точности поиска ϵ и отрезка $[a, b]$ для расчета длин шагов. Начинаем первую итерацию при $k = 0$.
2. *Проверим критерий останова:* Если выполнено основное условие $\|\nabla f(x^k)\| < \xi$ или дополнительное $k = N_{max}$, где k назовем номером итерации, выходим из алгоритма с результатом x^k .

3. *Вычислим направление поиска:* Найдем новое направление движения $dx^k = -H_f^{-1}(x^k)\nabla f(x^k)$.
4. *Вычислим длину шага:* Найдем длину шага $t_k = \underset{t \in [a,b]}{\operatorname{argmin}} |f(x^k + t \cdot dx^k)|$, поиск значения которой производится методом дихотомии на заданном отрезке $[a, b]$ с точностью ϵ .
5. *Вычисление следующей точки:* Найдем новую точку $x^{k+1} = x^k + t_k \cdot dx^k$, инкрементируем счетчик $k = k + 1$ и перейдем к шагу (2).

• **1-ая итерация:**

$$f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + x_2 + 9$$

$$\xi = 0.01$$

$$\epsilon = 0.001$$

$$[a, b] = [0.05, 0.5]$$

$$N_{max} = 5$$

$$k = 0$$

$$x_1^0 = 11, x_2^0 = 7$$

Градиент:

$$\nabla f(x) = \begin{pmatrix} 10x_1 + 3x_2 + 2 \\ 12x_2 + 3x_1 + 1 \end{pmatrix}$$

Матрица Гессе:

$$H_f(x) = \begin{pmatrix} 10 & 3 \\ 3 & 12 \end{pmatrix}$$

Проверим критерий останова:

$$\|\nabla f(x^k)\| = \left\| \begin{pmatrix} 133 \\ 118 \end{pmatrix} \right\| = \sqrt{133^2 + 118^2} = 177.8004 > 0.01 = \xi$$

$k = 0 < 5 = N_{max}$, значит критерий останова не выполнен.

Найдем направление движения:

$$dx^k = -H_f^{-1}(x^k)\nabla f(x^k) = \begin{pmatrix} 10 & 3 \\ 3 & 12 \end{pmatrix}^{-1} \begin{pmatrix} 133 \\ 118 \end{pmatrix} = (-11.1913, -7.0408)$$

Найдем длину шага:

Методом дихотомии определили оптимальное значение как $t_k = t_0 = 0.5$.

Вычислим следующую точку:

$$x^{k+1} = x^k + t_k \cdot dx^k = (11, 7) + 0.5 \cdot (-11.1913, -7.0408) = (5.4054, 3.4820)$$

$f(x^{k+1}) = 298.5946 < 1168 = f(x^k)$, значит новая точка найдена успешно.

$$k = k + 1$$

Конец первой итерации.

• Результат компьютерных вычислений:

Расчет окончен

Протокол расчета

Выполнил: Бронников, группа 80-307, 10.03.2020

Квадратичная функция: $f(x_1, x_2) = 5x_1^2 + 3x_1x_2 + 6x_2^2 + 2x_1 + 1x_2 + 9$

Метод Ньютона-Рафсона

Точность метода: 0.01, $N_{\max} = 5$, Количество итераций: 6

N _{ит}	шаг t	x ₁	x ₂	f(x ₁ , x ₂)	f' _{x₁}	f' _{x₂}	∇f(x ₁ , x ₂)
0	0.5	11	7	1168	133	118	177.80045
1	0.5	5.40541	3.48198	298.59459	66.5	59	88.90022
2	0.5	2.60811	1.72297	81.24324	33.25	29.5	44.45011
3	0.5	1.20946	0.84347	26.90541	16.625	14.75	22.22506
4	0.5	0.51014	0.40372	13.32095	8.3125	7.375	11.11253
5	0.5	0.16047	0.18384	9.92483	4.15625	3.6875	5.55626
6	0	-0.01436	0.0739	9.0758	2.07813	1.84375	2.77813

Критерий окончания не выполнен

$$\|x - x^*\| = 0.20652$$

$$|f(x) - f(x^*)| = 0.28301$$

6 Выводы

Выполнив лабораторную работу по курсу «Методы оптимизации», я узнал пять новых для себя алгоритмов безусловной оптимизации, позволяющих находить минимумы функций в процессе итерационных приближений, и познакомился с преимуществами и недостатками каждого из них.

Самым полезным алгоритмом из рассмотренных мною при выполнении лабораторной работы, на мой взгляд, является алгоритм Нельдера-Мида, поскольку он не достаточно оригинален и не требует вычисления градиента и матрицы Гессе, что дает преимущество в производительности и в работе с негладкими функциями, а также не накладывает ограничений на положительную определенность матриц.

В процессе изучения я столкнулся с такими трудностями, как нехватка хорошего и подробного материала по методу конфигураций и методу Нельдера-Мида. Немало хлопот мне доставило разобраться с неточностью в методичке, касающуюся метода конфигураций, а также сформулировать этот метод в виде последовательного, понятного алгоритма.

Данная лабораторная обогатила меня новыми знаниями о методах, которые я непременно буду использовать в будущем, решая задачи машинного обучения.

Список литературы

- [1] *Методы дихотомии*
URL: http://www.machinelearning.ru/wiki/index.php?title=Методы_дихотомии
(дата обращения: 06.03.2020).
- [2] *Метод Нелдера-Мида - Википедия*
URL: https://ru.wikipedia.org/wiki/Метод_Нелдера_-_Мида
(дата обращения: 06.03.2020).
- [3] *Лекция 10: Многометрическая оптимизация - НОУ ИНТУИТ*
URL: <https://www.intuit.ru/studies/courses/1020/188/lecture/4931>
(дата обращения: 07.03.2020).
- [4] *Алгоритм Левенберга-Марквардта - Википедия*
URL: https://ru.wikipedia.org/wiki/Алгоритм_Левенберга_-_Марквардта
(дата обращения: 09.03.2020).
- [5] *Метод Ньютона - Википедия*
URL: https://ru.wikipedia.org/wiki/Метод_Ньютона
(дата обращения: 09.03.2020).
- [6] *Метод сопряженных градиентов - Википедия*
URL: https://ru.wikipedia.org/wiki/Метод_сопряжённых_градиентов
(дата обращения: 09.03.2020).