**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Методы оптимизации»**

Тема: **Методы безусловной минимизации функций**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0304 |  | Люлин Д.В. |
| Преподаватель |  | Мальцева Н.В. |
|  |  |  |

Санкт-Петербург

2023

**Методы безусловной минимизации функций**

***Цели работы:***

1. Решение задачи безусловной минимизации функций с помощью стандартной программы.
2. Исследование и объяснение полученных результатов.

***Постановка задачи.***

Минимизировать функцию F(x1,x2,a) = (x2 - x12)2  + a(x1 - 1)2

с точностью до 10-5  ( abs ( F(x1k,x2k,a) - F(x1\*,x2\*,a) ) < 10-5 ) предложенными в задании методами. Оценить скорость и порядок сходимости методов. Провести сравнительный анализ эффективности методов в зависимости от предложенных параметров (начальной точки, величины шага, параметра а>0).

Вариант 21. Метод с дроблением шага и метод наискорейшего спуска.

***Краткие общие сведения***

 - порядок сходимости метода, где k=||xk-x\*||

(xk)-(x\*)<constqk – геометрическая скорость сходимости, где q<1

(xk)-(x\*)<constq2k – квадратичная скорость сходимости, где q<1

Для проведения лабораторной работы составлена программа, обеспечивающая решение задачи безусловной минимизации при задании с терминала исходных значений.

**Градиентные методы.** При использовании градиентных методов в задаче минимизации функции выбирается начальное приближение и строится релаксационная последовательность такая, что . Точка минимума функции обозначается за . Последовательность строится по следующему принципу:

Параметр – величина шага. Если , то – стационарная точка, и процесс прекращается. Выбор величины шага зависит от используемого градиентного метода.

**Метод наискорейшего спуска.** В данном методе рассматривается луч, направленный в сторону антиградиента (то есть, в направлении -):



На луче вводится функция . За величину шага принимается точка минимума этой функции, то есть:



Таким образом, на каждом шаге нужно составлять функцию и находить её точку минимума с помощью производной .

Метод наискорейшего спуска имеет линейную скорость сходимости и порядок сходимости, равный 1. На каждом шаге направление спуска меняется на ортогональное, а также обеспечивается минимальное значение в определённом для данного шага направлении.

**Метод с дроблением шага.** Данный метод не требует решения задачи одномерной оптимизации на каждом шаге, в отличие от метода наискорейшего спуска. В нём вводятся дополнительные параметры (обычно ). На каждом шаге для проверяется условие . Если оно выполняется, то принимается и производится шаг градиентного метода. Иначе происходит дробление шага, и условие проверяется для . Дробление шага происходит до тех пор, пока условие не начнёт выполняться.

Метод с дроблением шага также имеет линейную сходимость и порядок сходимости 1. Но в отличие от метода наискорейшего спуска, он не обеспечивает минимальное значение на каждом шаге, из-за чего может потребоваться больше шагов. Но на каждом шаге нужно меньше вычислений, потому что не нужно решать задачу одномерной оптимизации.

***Выбор перечня вариантов запуска программы.***

Проанализировав функцию F(x1,x2,a) = (x2 - x12)2  + a(x1 - 1)2, можно увидеть, что она неотрицательна, потому что оба слагаемых неотрицательны (при условии, что a > 0). Минимальное значение функции равно 0 при любом a. Оно достигается в точке {1;1}. Параметр a влияет на скорость роста функции в зависимости от x1. Таким образом, можно проверить градиентные методы с начальным приближением как близким, так и далёким от точки минимума. При этом можно изменять параметр а на порядок, чтобы проверить действие методов при больших векторах градиента.

Начальные приближения:

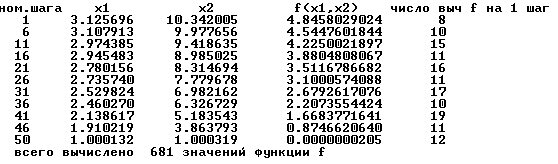
* {10, 10} – обе координаты далеки от точки минимума.
* {10, 3} – первая координата далека, а вторая – близка к точке минимума.
* {3, 10} – вторая координата далека, а первая – близка к точке минимума.

При приближении обеих координат к точке минимума разница в работе методов будет не так заметна. Измерения будут проводиться при a = 1, 10, 100.

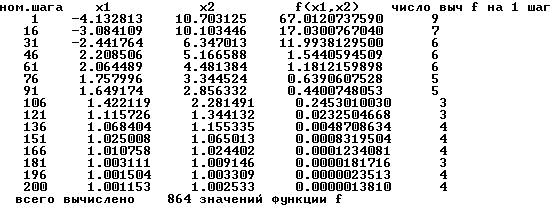
***Протокол запуска программы.***

1. {10; 10}, a = 1.

Метод наискорейшего спуска:

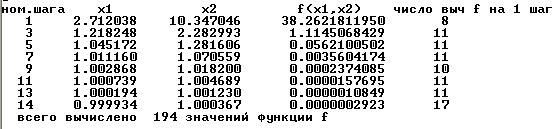


Метод с дроблением шага:

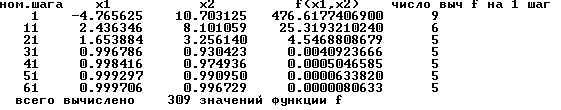


1. {10; 10}, a = 10.

Метод наискорейшего спуска:



Метод с дроблением шага:

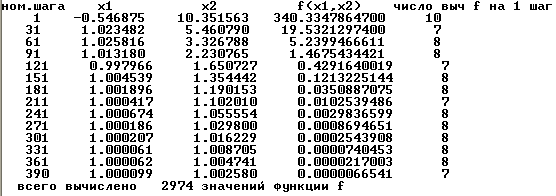


1. {10; 10}, a = 100.

Метод наискорейшего спуска:

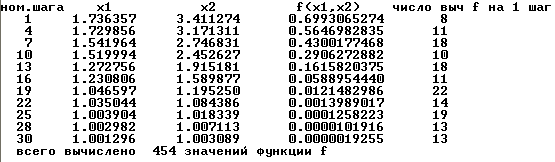


Метод с дроблением шага:

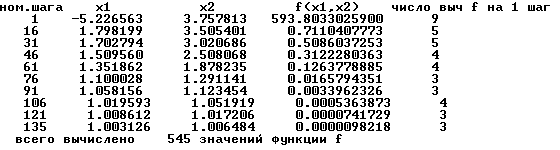


1. {10; 3}, a = 1.

Метод наискорейшего спуска:

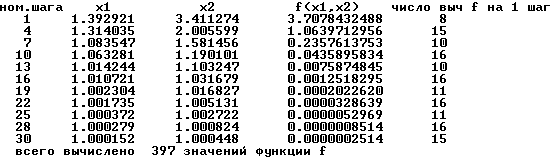


Метод с дроблением шага:

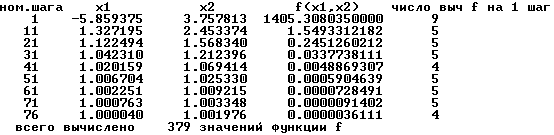


1. {10; 3}, a = 10.

Метод наискорейшего спуска:

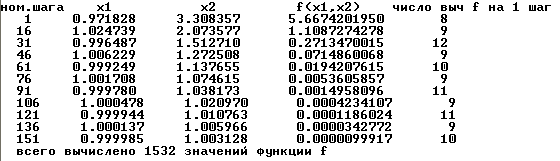


Метод с дроблением шага:

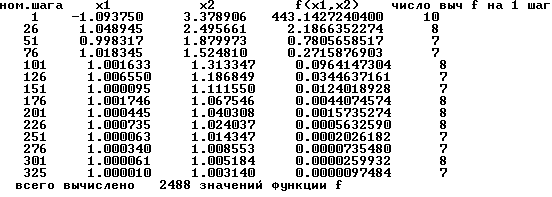


1. {10; 3}, a = 100.

Метод наискорейшего спуска:

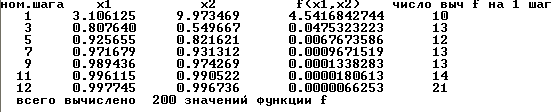


Метод с дроблением шага:

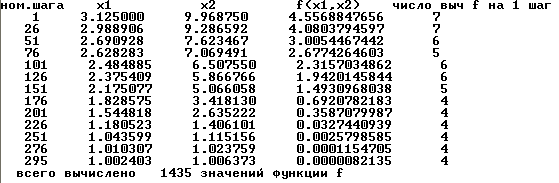


1. {3; 10}, a = 1.

Метод наискорейшего спуска:

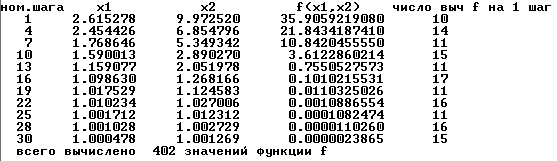


Метод с дроблением шага:

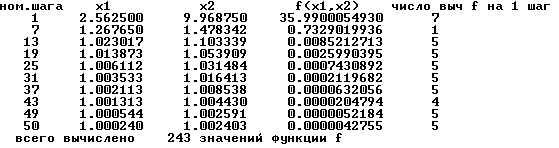


1. {3; 10}, a = 10.

Метод наискорейшего спуска:

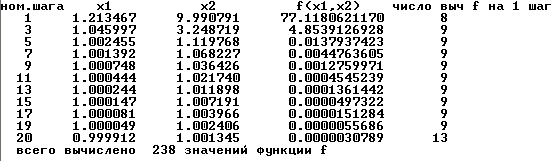


Метод с дроблением шага:

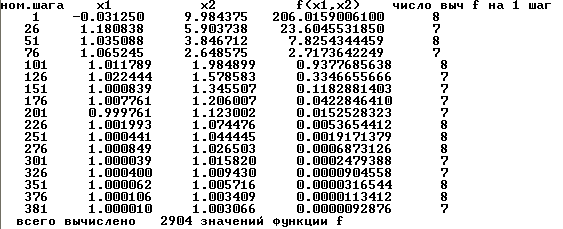


1. {3; 10}, a = 100.

Метод наискорейшего спуска:



Метод с дроблением шага:



***Оценка скорости и порядка сходимости методов.***

Рассчитаем оценку скорости и порядка сходимости для обоих методов при начальном приближении {10; 3} и параметре a = 10.

Для метода наискорейшего спуска:

|  |  |  |
| --- | --- | --- |
| k |  |  |
| 16 | 1.0203226004407169 | 0.9332764342902399 |
| 17 | 1.1555466033322985 | 0.583173588178649 |
| 18 | 1.0172919844409585 | 0.9330697814884805 |
| 19 | 1.1315688303990237 | 0.5849630569310058 |
| 20 | 1.0150087581496272 | 0.9331253462349217 |
| 21 | 1.1148107329482642 | 0.5842547257259969 |
| 22 | 1.0132844618995969 | 0.9330256135870694 |
| 23 | 1.102080118058116 | 0.5828860386243996 |
| 24 | 1.0119651446849833 | 0.9326495121085602 |
| 25 | 1.0913925667866502 | 0.5833578973255702 |

Для метода с дроблением шага:

|  |  |  |
| --- | --- | --- |
| k |  |  |
| 61 | 1.0198293095300845 | 0.9117733803547228 |
| 62 | 1.0194084508654409 | 0.911926295334282 |
| 63 | 1.0190027628217666 | 0.912086044294203 |
| 64 | 1.0186292082331398 | 0.9121723883146333 |
| 65 | 1.0182243524546084 | 0.9124665873992636 |
| 66 | 1.0359632673556756 | 0.8318842538123953 |
| 67 | 1.0176736111508766 | 0.9105490166176893 |
| 68 | 1.017261379291117 | 0.9110665192433907 |
| 69 | 1.0168833115686071 | 0.9114925296440817 |
| 70 | 1.0165614689523577 | 0.911703838004697 |

Результаты совпали с теоретическими: оба метода имеют линейную сходимость, порядок сходимости примерно равен 1.

***Сравнение методов.***

Составим итоговые таблицы. При a =1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Метод наискорейшего спуска | | Метод с дроблением шага | |
| шагов | вычислений F | шагов | вычислений F |
| {10, 10} | 50 | 681 | 200 | 864 |
| {10, 3} | 30 | 454 | 135 | 545 |
| {3, 10} | 12 | 200 | 295 | 1435 |

При a=10:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Метод наискорейшего спуска | | Метод с дроблением шага | |
| шагов | вычислений F | шагов | вычислений F |
| {10, 10} | 14 | 194 | 61 | 309 |
| {10, 3} | 30 | 397 | 76 | 379 |
| {3, 10} | 30 | 402 | 50 | 243 |

При a=100:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Метод наискорейшего спуска | | Метод с дроблением шага | |
| шагов | вычислений F | шагов | вычислений F |
| {10, 10} | 301 | 2900 | 390 | 2974 |
| {10, 3} | 151 | 1532 | 325 | 2488 |
| {3, 10} | 20 | 238 | 381 | 2904 |

В среднем один шаг для метода наискорейшего спуска требует 15-16 вычислений F, а метод с дроблением шага – 5-7.

Видно, что при a = 100 и x1 = 10 оба метода работают дольше, потому что функция круто возрастает при увеличении x1, и точка находится достаточно далеко от точки минимума.

Метод наискорейшего спуска при каждом запуске сходится за меньшее число шагов, чем метод с дроблением шага. Так происходит, потому что в методе наискорейшего спуска на каждом шаге обеспечивается минимальное значение в определённом направлении, в то время как в методе с дроблением шага при выборе шаг делится на фиксированное число *λ*, что не всегда оптимально.

Однако метод с дроблением шага требует меньше вычислений на одном шаге, потому что не требует минимизации функции , как метод наискорейшего спуска.

***Вывод.***

В ходе работы была проведена минимизация функции  
F(x1,x2,a) = (x2 - x12)2  + a(x1 - 1)2 с точностью до 10-5  с помощью программы методами наискорейшего спуска и с дроблением шага.

Были выбраны параметры запуска программы для исследования методов. Программа запускалась для начальных приближений {10;10}, {10; 3} и {3; 10} с параметрами a=1,10,100.

Было установлено, что метод наискорейшего спуска сходится за меньшее количество шагов, но требует большее количество вычислений на одном шаге, чем метод с дроблением шага.