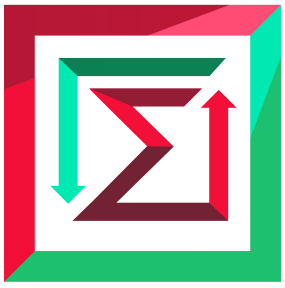
МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»

НГТУ  НЭТИ

Кафедра теоретических и прикладных приколов

Лабораторная работа №2
по дисциплине «Методы оптимизации»
**Методы спуска (0-го, 1-го и 2-го порядка
и переменной метрики)**



ФПМИ

| | |
|---|---|
| Группа | ПМ-92 |
| Бригада | 08 |
| Студенты | БЕГИЧЕВ АЛЕКСАНДР |
| | ШИШКИН НИКИТА |
| Преподаватель | ФИЛИППОВА Е.В. |
| Дата | 22.03.2022 |

Новосибирск

## Цель работы

Ознакомиться с методами поиска минимума функции n переменных в оптимизационных задачах без ограничений.

## Задание

Вариант 8: $f(x,y) = \dfrac{3}{1+\left(\dfrac{x-2}{1}\right)^2+\left(\dfrac{y-2}{2}\right)^2} + \dfrac{2}{1+\left(\dfrac{x-2}{3}\right)^2+\left(\dfrac{y-3}{1}\right)^2}$.

1. Реализовать **два метода** поиска экстремума функции (разного порядка). Включить в реализуемый алгоритм собственную процедуру, реализующую одномерный поиск по направлению. Методы поиска для самостоятельной реализации выбираются студентом в зависимости от уровня сложности. Выбранные методы должны иметь разный порядок (например, метод Гаусса (нулевого порядка) и метод Ньютона (второго порядка).

2. С использованием разработанного програмного обеспечения исследовать алгоритмы на квадратичной функции $f(\overline{x}) = 100(x_2 - x_1)^2 + (1 - x_1)^2$, функции Розенброка $f(\overline{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ и на заданной в соответствии с вариантом тестовой функции, осуществляя спуск из различных исходных точек (не менее двух). Исследовать сходимость алгоритма, фиксируя точность определения минимума/максимума, количество итераций метода и количество вычислений функции в зависимости от задаваемой точности поиска. Результатом выполнения данного пункта должны быть выводы об объеме вычислений в зависимости от задаваемой точности и начального приближения.

3. Построить траекторию спуска различных алгоритмов из одной и той же исходной точки с одинаковой точностью. В отчете наложить эту траекторию на рисунок с линиями равного уровня заданной функции.

4. Реализовать метод квадратичной интерполяции (метод парабол) для приближенного нахождения экстремума при одномерном поиске. Исследовать влияние точности одномерного поиска на общее количество итераций и вычислений функции при разных методах одномерного поиска.

# Исследования

## Сравнение сходимости методов

| № | name | $x_{01}$ | $x_{02}$ | $\epsilon$ | iters | calcs | $x_1$ | $x_2$ | $f(x_1, x_2)$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | bfgs (quad) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-3}$ | 6 | 310 | 1.99977402 | 2.75646882 | −4.51255712 |
| 2 | bfgs (fib) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-3}$ | 6 | 270 | 1.99946121 | 2.75632340 | −4.51255654 |
| 3 | simplex | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-3}$ | 28 | 552 | 1.99689215 | 2.76853641 | −4.51228393 |
| 4 | bfgs (quad) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-4}$ | 9 | 506 | 1.99969911 | 2.75636543 | −4.51255704 |
| 5 | bfgs (fib) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-4}$ | 3 | 178 | 1.99913415 | 2.75588406 | −4.51255501 |
| 6 | simplex | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-4}$ | 31 | 608 | 2.00120401 | 2.76135791 | −4.51251191 |
| 7 | bfgs (quad) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-5}$ | 5 | 390 | 1.99778669 | 2.74825904 | −4.51243592 |
| 8 | bfgs (fib) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-5}$ | 6 | 407 | 1.99911268 | 2.75638067 | −4.51255530 |
| 9 | simplex | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-5}$ | 35 | 678 | 2.00063196 | 2.75770707 | −4.51255326 |
| 10 | bfgs (quad) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-6}$ | 5 | 464 | 1.99881162 | 2.75468524 | −4.51254904 |
| 11 | bfgs (fib) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-6}$ | 4 | 317 | 2.00108923 | 2.75016973 | −4.51249049 |
| 12 | simplex | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-6}$ | 39 | 742 | 2.00026362 | 2.75657612 | −4.51255702 |
| 13 | bfgs (quad) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-7}$ | 6 | 656 | 1.99896028 | 2.75672748 | −4.51255435 |
| 14 | bfgs (fib) | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-7}$ | 4 | 358 | 2.00041918 | 2.75487266 | −4.51255311 |
| 15 | simplex | −1.00000000 | 2.00000000 | $1.00 \cdot 10^{-7}$ | 43 | 812 | 2.00006707 | 2.75644623 | −4.51255725 |
| 16 | bfgs (quad) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-3}$ | 1 | 432 | −667,572,023.48437500 | 3.00000000 | 0.00000000 |
| 17 | bfgs (fib) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-3}$ | 2 | 104 | 2.00006533 | 2.75599759 | −4.51255703 |
| 18 | simplex | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-3}$ | 27 | 532 | 1.99394278 | 2.76381383 | −4.51237302 |
| 19 | bfgs (quad) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-4}$ | 2 | 284 | 2.00009667 | 2.74852039 | −4.51245507 |
| 20 | bfgs (fib) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-4}$ | 2 | 125 | 2.00000567 | 2.75633094 | −4.51255727 |
| 21 | simplex | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-4}$ | 30 | 582 | 2.00331265 | 2.75106432 | −4.51248308 |
| 22 | bfgs (quad) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-5}$ | 3 | 344 | 2.00000575 | 2.75632362 | −4.51255727 |
| 23 | bfgs (fib) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-5}$ | 2 | 149 | 2.00000010 | 2.75636531 | −4.51255727 |
| 24 | simplex | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-5}$ | 33 | 638 | 2.00124821 | 2.75369686 | −4.51254151 |
| 25 | bfgs (quad) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-6}$ | 1 | 440 | −1,335,144,044.96875000 | 3.00000000 | 0.00000000 |
| 26 | bfgs (fib) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-6}$ | 3 | 188 | 2.00000007 | 2.75636541 | −4.51255727 |
| 27 | simplex | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-6}$ | 39 | 744 | 1.99971855 | 2.75606684 | −4.51255692 |
| 28 | bfgs (quad) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-7}$ | 2 | 498 | 2.00000009 | 2.74814196 | −4.51244502 |
| 29 | bfgs (fib) | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-7}$ | 2 | 193 | 2.00000000 | 2.75636537 | −4.51255727 |
| 30 | simplex | −2.00000000 | 3.00000000 | $1.00 \cdot 10^{-7}$ | 42 | 798 | 2.00015276 | 2.75645321 | −4.51255720 |

## Исследование сходимости каждого из методов

Далее начальное приближение $x_0 = \{-1, 2\}$, $\varepsilon = 10^{-3}$.

## Целевая функция варианта (многогранники)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle |
|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | −1.4026634 | NaN | NaN | NaN | NaN | NaN | NaN |
| 1 | 0.9659258 | 0.4829629 | −1.4026634 | 1.9659258 | −1.5170371 | 1.9659258 | 1.5170371 | 0.4359967 | 1.1208710 |
| 2 | 1.5866666 | 0.0866666 | −1.6485032 | 0.6207407 | −0.3962964 | 0.6207407 | 0.3962964 | 0.2458398 | 0.6227626 |
| 3 | 1.5866666 | 0.0866666 | −1.6485032 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 4 | 1.5866666 | 0.0866666 | −1.6485032 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 5 | 1.5866666 | 0.0866666 | −1.6485032 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 6 | 1.5866666 | 0.0866666 | −1.6485032 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 7 | 1.5186342 | 0.2774419 | −1.7571517 | −0.0680324 | 0.1907754 | 0.0680324 | 0.1907754 | 0.1086485 | 1.7326484 |
| 8 | 1.5186342 | 0.2774419 | −1.7571517 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 9 | 1.6962500 | 0.3768747 | −1.9668585 | 0.1776158 | 0.0994327 | 0.1776158 | 0.0994327 | 0.2097068 | 0.2917209 |
| 10 | 1.3668172 | 0.8027484 | −2.0458222 | −0.3294328 | 0.4258738 | 0.3294328 | 0.4258738 | 0.0789637 | 1.6981592 |
| 11 | 1.5573323 | 1.2145508 | −2.6970183 | 0.1905152 | 0.4118023 | 0.1905152 | 0.4118023 | 0.6511961 | 0.4751239 |
| 12 | 1.5573323 | 1.2145508 | −2.6970183 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 13 | 1.4442134 | 2.6769661 | −3.8639573 | −0.1131189 | 1.4624153 | 0.1131189 | 1.4624153 | 1.1669390 | 0.5719403 |
| 14 | 2.0465196 | 2.5564262 | −4.4497393 | 0.6023062 | −0.1205399 | 0.6023062 | 0.1205399 | 0.5857820 | 1.0932477 |
| 15 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 16 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 17 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 18 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 19 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 20 | 2.0465196 | 2.5564262 | −4.4497393 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 21 | 1.9909022 | 2.8262669 | −4.5038186 | −0.0556174 | 0.2698406 | 0.0556174 | 0.2698406 | 0.0540793 | 0.8169574 |
| 22 | 1.9909022 | 2.8262669 | −4.5038186 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 23 | 1.9909022 | 2.8262669 | −4.5038186 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 24 | 1.9909022 | 2.8262669 | −4.5038186 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 25 | 2.0159252 | 2.7353153 | −4.5111890 | 0.0250230 | −0.0909516 | 0.0250230 | 0.0909516 | 0.0073704 | 2.2379825 |
| 26 | 2.0159252 | 2.7353153 | −4.5111890 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 27 | 2.0159252 | 2.7353153 | −4.5111890 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 28 | 1.9873955 | 2.7628651 | −4.5120916 | −0.0285297 | 0.0275498 | 0.0285297 | 0.0275498 | 0.0009025 | 1.4264463 |

## Квадратичная функция (многогранники)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle |
|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 23.3264786 | NaN | NaN | NaN | NaN | NaN | NaN |
| 1 | 0.9659258 | 0.4829629 | 23.3264786 | 1.9659258 | −1.5170371 | 1.9659258 | 1.5170371 | 880.6735214 | 1.1208710 |
| 2 | 0.9659258 | 0.4829629 | 23.3264786 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 3 | 0.9659258 | 0.4829629 | 23.3264786 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 4 | 0.9399999 | 0.5649999 | 14.0661000 | −0.0259259 | 0.0820370 | 0.0259259 | 0.0820370 | 9.2603786 | 1.3356899 |
| 5 | 0.7179629 | 0.7449536 | 0.1523949 | −0.2220370 | 0.1799537 | 0.2220370 | 0.1799537 | 13.9137051 | 1.6566558 |
| 6 | 0.7179629 | 0.7449536 | 0.1523949 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 7 | 0.7179629 | 0.7449536 | 0.1523949 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 8 | 0.7179629 | 0.7449536 | 0.1523949 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 9 | 0.7179629 | 0.7449536 | 0.1523949 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 10 | 0.7179629 | 0.7449536 | 0.1523949 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 11 | 0.7347540 | 0.7489850 | 0.0906076 | 0.0167911 | 0.0040314 | 0.0167911 | 0.0040314 | 0.0617872 | 0.5593585 |
| 12 | 0.7450542 | 0.7388492 | 0.0688476 | 0.0103002 | −0.0101358 | 0.0103002 | 0.0101358 | 0.0217601 | 1.5585710 |
| 13 | 0.7508747 | 0.7433988 | 0.0676522 | 0.0058205 | 0.0045496 | 0.0058205 | 0.0045496 | 0.0011953 | 0.1169379 |
| 14 | 0.7508747 | 0.7433988 | 0.0676522 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 15 | 0.7471797 | 0.7496042 | 0.0645059 | −0.0036950 | 0.0062053 | 0.0036950 | 0.0062053 | 0.0031463 | 1.3208480 |
| 16 | 0.7471797 | 0.7496042 | 0.0645059 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 17 | 0.7530002 | 0.7541538 | 0.0611420 | 0.0058205 | 0.0045496 | 0.0058205 | 0.0045496 | 0.0033639 | 0.1227063 |
| 18 | 0.7530002 | 0.7541538 | 0.0611420 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 19 | 0.7592131 | 0.7544634 | 0.0602343 | 0.0062129 | 0.0003096 | 0.0062129 | 0.0003096 | 0.0009077 | 0.7324734 |
| 20 | 0.7615348 | 0.7630983 | 0.0571101 | 0.0023217 | 0.0086349 | 0.0023217 | 0.0086349 | 0.0031242 | 0.5217114 |
| 21 | 0.7751213 | 0.7680349 | 0.0555921 | 0.0135866 | 0.0049366 | 0.0135866 | 0.0049366 | 0.0015180 | 0.4322922 |
| 22 | 0.7865580 | 0.7877731 | 0.0457051 | 0.0114367 | 0.0197382 | 0.0114367 | 0.0197382 | 0.0098870 | 0.2594767 |
| 23 | 0.8001446 | 0.7927097 | 0.0454699 | 0.0135866 | 0.0049366 | 0.0135866 | 0.0049366 | 0.0002352 | 0.4322167 |
| 24 | 0.8298112 | 0.8346543 | 0.0313098 | 0.0296667 | 0.0419446 | 0.0296667 | 0.0419446 | 0.0141602 | 0.1668924 |
| 25 | 0.8718177 | 0.8654999 | 0.0204222 | 0.0420065 | 0.0308455 | 0.0420065 | 0.0308455 | 0.0108876 | 0.1483814 |
| 26 | 0.9014844 | 0.9074445 | 0.0132577 | 0.0296667 | 0.0419446 | 0.0296667 | 0.0419446 | 0.0071645 | 0.1665073 |
| 27 | 0.9434908 | 0.9382901 | 0.0058981 | 0.0420065 | 0.0308455 | 0.0420065 | 0.0308455 | 0.0073596 | 0.1492542 |
| 28 | 0.9731575 | 0.9802347 | 0.0057292 | 0.0296667 | 0.0419446 | 0.0296667 | 0.0419446 | 0.0001689 | 0.1661791 |
| 29 | 1.0151640 | 1.0110803 | 0.0018976 | 0.0420065 | 0.0308455 | 0.0420065 | 0.0308455 | 0.0038316 | 0.1500025 |
| 30 | 0.9688258 | 0.9669738 | 0.0013148 | −0.0463382 | −0.0441065 | 0.0463382 | 0.0441065 | 0.0005828 | 3.1178796 |

## Функция Розенброка (многогранники)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle |
|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 20.2556423 | NaN | NaN | NaN | NaN | NaN | NaN |
| 1 | 0.9659258 | 0.4829629 | 20.2556423 | 1.9659258 | −1.5170371 | 1.9659258 | 1.5170371 | 83.7443577 | 1.1208710 |
| 2 | 0.7244444 | 0.7244444 | 4.0609340 | −0.2414815 | 0.2414815 | 0.2414815 | 0.2414815 | 16.1947083 | 1.5707963 |
| 3 | 0.7244444 | 0.7244444 | 4.0609340 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 4 | 0.7244444 | 0.7244444 | 4.0609340 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 5 | 0.8214814 | 0.6133796 | 0.4095050 | 0.0970370 | −0.1110648 | 0.0970370 | 0.1110648 | 3.6514289 | 1.4940742 |
| 6 | 0.8214814 | 0.6133796 | 0.4095050 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 7 | 0.8214814 | 0.6133796 | 0.4095050 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 8 | 0.8223726 | 0.6319957 | 0.2278090 | 0.0008912 | 0.0186162 | 0.0008912 | 0.0186162 | 0.1816960 | 0.8677219 |
| 9 | 0.8003385 | 0.6468516 | 0.0438463 | −0.0220341 | 0.0148559 | 0.0220341 | 0.0148559 | 0.1839627 | 1.8686441 |
| 10 | 0.8003385 | 0.6468516 | 0.0438463 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 11 | 0.8003385 | 0.6468516 | 0.0438463 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 12 | 0.8003385 | 0.6468516 | 0.0438463 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 13 | 0.8067723 | 0.6476183 | 0.0384018 | 0.0064338 | 0.0007667 | 0.0064338 | 0.0007667 | 0.0054445 | 0.5577934 |
| 14 | 0.8067723 | 0.6476183 | 0.0384018 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |
| 15 | 0.8067723 | 0.6476183 | 0.0384018 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | NaN |

## Целевая функция варианта (BFGS, метод Фибоначчи)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | −0.9666667 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | −0.0863226 | 3.2917508 | −1.7948350 | 0.9136774 | 1.2917508 | 2.8378186 | 0.9136774 | 1.2917508 | 0.8281684 | 0.6418378 | −0.321965 | −0.455191 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 2 | 1.8754245 | 1.9069881 | −3.8584573 | 1.9617471 | −1.3847627 | 2.5985069 | 1.9617471 | 1.3847627 | 2.0636222 | 1.4084058 | −0.754952 | 0.532907 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 3 | 2.2149726 | 2.4108337 | −4.2353306 | 0.3395481 | 0.5038456 | 0.4932867 | 0.3395481 | 0.5038456 | 0.3768733 | 0.1500799 | −0.688338 | −1.021405 | 0.161954 | 0.152576 | 0.152576 | 0.972222 |
| 4 | 2.1682988 | 2.7948206 | −4.4423834 | −0.0466738 | 0.3839869 | 0.6462134 | 0.0466738 | 0.3839869 | 0.2070528 | 0.7807840 | 1.199041 | −0.799361 | 0.117689 | −0.005214 | −0.005214 | 0.409757 |
| 5 | 1.9986034 | 2.7680266 | −4.5123236 | −0.1696953 | −0.0267940 | 0.2011157 | 0.1696953 | 0.0267940 | 0.0699401 | 2.3527581 | 0.843769 | 0.133227 | 0.208779 | −0.072803 | −0.072803 | 0.993301 |
| 6 | 1.9994612 | 2.7563234 | −4.5125565 | 0.0008578 | −0.0117032 | 0.2653103 | 0.0008578 | 0.0117032 | 0.0002330 | 2.4408535 | 0.000000 | 0.044409 | 0.204859 | −0.019315 | −0.019315 | 0.263533 |

## Целевая функция варианта (BFGS, метод парабол)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | −0.9666667 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | −0.0270499 | 3.3755502 | −1.7893340 | 0.9729501 | 1.3755502 | 3.0219158 | 0.9729501 | 1.3755502 | 0.8226673 | 0.6236332 | −0.321965 | −0.455191 | −5.106763 | −1.104240 | −1.104240 | 0.800329 |
| 2 | 1.1848726 | 3.9064913 | −2.2210058 | 1.2119224 | 0.5309411 | −0.3884631 | 1.2119224 | 0.5309411 | 0.4316718 | 0.8633944 | −0.754952 | 0.666134 | −16.826059 | 0.113943 | 0.113943 | 0.673703 |
| 3 | 1.9427073 | 2.5546781 | −4.4457722 | 0.7578347 | −1.3518132 | 0.9224281 | 0.7578347 | 1.3518132 | 2.2247664 | 1.9804869 | −0.821565 | 1.465494 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 4 | 2.0385823 | 2.7190353 | −4.5065113 | 0.0958750 | 0.1643572 | 0.3084163 | 0.0958750 | 0.1643572 | 0.0607391 | 0.1152632 | −0.310862 | −0.532907 | 0.415825 | −0.314553 | −0.314553 | 0.830626 |
| 5 | 1.9953063 | 2.7450009 | −4.5122881 | −0.0432760 | 0.0259656 | 0.1948978 | 0.0432760 | 0.0259656 | 0.0057768 | 1.6589210 | 0.222045 | −0.133227 | 0.234295 | 0.215641 | 0.215641 | 0.939270 |
| 6 | 1.9997740 | 2.7564688 | −4.5125571 | 0.0044677 | 0.0114679 | 0.2235963 | 0.0044677 | 0.0114679 | 0.0002690 | 0.2561294 | −0.044409 | −0.044409 | 0.136356 | −0.035752 | −0.035752 | 0.293986 |

## Квадратичная функция (BFGS, метод Фибоначчи)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 904.0000000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | 0.4929924 | 0.4630961 | 0.3464358 | 1.4929924 | −1.5369039 | 0.0025750 | 1.4929924 | 1.5369039 | 903.6535642 | 1.5540288 | −579.802872 | 596.855898 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 2 | 0.4802206 | 0.4784765 | 0.2704748 | −0.0127718 | 0.0153805 | 0.0025750 | 0.0127718 | 0.0153805 | 0.0759610 | 1.4802163 | 4.959921 | −5.973000 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 3 | 0.4971105 | 0.4870232 | 0.2630731 | 0.0168899 | 0.0085467 | 0.0244387 | 0.0168899 | 0.0085467 | 0.0074017 | 0.3066934 | −0.691114 | −0.349720 | 0.498909 | 0.498746 | 0.498746 | 0.503588 |
| 4 | 1.0002990 | 1.0001522 | 0.0000022 | 0.5031885 | 0.5131290 | 1.0016996 | 0.5031885 | 0.5131290 | 0.2630709 | 0.0098539 | 1.007527 | −2.015055 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 5 | 1.0002219 | 1.0002277 | 0.0000001 | −0.0000771 | 0.0000756 | 0.0025750 | 0.0000771 | 0.0000756 | 0.0000022 | 1.5808737 | 0.029938 | −0.029340 | 0.498754 | 0.498746 | 0.498746 | 0.503742 |
| 6 | 1.0000000 | 1.0000000 | 0.0000000 | −0.0002220 | −0.0002278 | 1.0034830 | 0.0002220 | 0.0002278 | 0.0000001 | 3.1286108 | −0.000720 | 0.001164 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |

## Квадратичная функция (BFGS, метод парабол)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 904.0000000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | 0.4795246 | 0.4769600 | 0.2715524 | 1.4795246 | −1.5230400 | 0.0025518 | 1.4795246 | 1.5230400 | 903.7284476 | 1.5826067 | −579.802872 | 596.855898 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 2 | 0.9599785 | 0.9447703 | 0.0247306 | 0.4804538 | 0.4678103 | 0.9110618 | 0.4804538 | 0.4678103 | 0.2468218 | 0.0053484 | −0.527356 | −0.513478 | 0.499122 | 0.498779 | 0.498779 | 0.503311 |
| 3 | 0.9525911 | 0.9523572 | 0.0022531 | −0.0073873 | 0.0075870 | 0.0024963 | 0.0073873 | 0.0075870 | 0.0224775 | 1.5575881 | 2.959265 | −3.039236 | 0.498753 | 0.498746 | 0.498746 | 0.503742 |
| 4 | 1.0000000 | 1.0000000 | 0.0000000 | 0.0474089 | 0.0476428 | 1.0033549 | 0.0474089 | 0.0476428 | 0.0022531 | 0.0024608 | −0.048008 | −0.046729 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |

## Функция Розенброка (BFGS, метод Фибоначчи)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 104.0000000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | −1.3964828 | 1.7982059 | 8.0522679 | −0.3964828 | −0.2017941 | 0.0010000 | 0.3964828 | 0.2017941 | 95.9477321 | 1.3812791 | 396.482847 | 201.794137 | 0.186347 | −0.388523 | −0.388523 | 0.814479 |
| 2 | −0.8478101 | 0.6692751 | 3.6594961 | 0.5486727 | −1.1289308 | 0.1116372 | 0.5486727 | 1.1289308 | 4.3927719 | 2.6914442 | −89.706020 | −30.375702 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 3 | −0.8273598 | 0.6791560 | 3.3421257 | 0.0204503 | 0.0098810 | 0.0010000 | 0.0204503 | 0.0098810 | 0.3173704 | 2.0041606 | −20.450308 | −9.880985 | 0.255422 | −0.434988 | −0.434988 | 0.745876 |
| 4 | −0.6539857 | 0.3849571 | 2.9183410 | 0.1733741 | −0.2941989 | 0.1915941 | 0.1733741 | 0.2941989 | 0.4237847 | 2.6353368 | −5.395684 | −1.088019 | 0.068629 | −0.107097 | −0.107097 | 0.170305 |
| 5 | −0.6250311 | 0.4020102 | 2.6535999 | 0.0289546 | 0.0170530 | 0.0020000 | 0.0289546 | 0.0170530 | 0.2647411 | 2.0377636 | −14.477308 | −8.526513 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 6 | −0.6216553 | 0.3828809 | 2.6310437 | 0.0033758 | −0.0191293 | 0.0084462 | 0.0033758 | 0.0191293 | 0.0225562 | 2.2975011 | −0.399680 | 2.264855 | 0.390074 | −0.486551 | −0.486551 | 0.611868 |
| 7 | −0.6110205 | 0.3847677 | 2.6084324 | 0.0106348 | 0.0018868 | 0.0025750 | 0.0106348 | 0.0018868 | 0.0226113 | 2.4040191 | −4.130030 | −0.732747 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 8 | −0.6074572 | 0.3654508 | 2.5851814 | 0.0035633 | −0.0193169 | 0.0084462 | 0.0035633 | 0.0193169 | 0.0232510 | 2.2948124 | −0.421885 | 2.287059 | 0.404480 | −0.489561 | −0.489561 | 0.597544 |
| 9 | −0.5970511 | 0.3672805 | 2.5622588 | 0.0104061 | 0.0018296 | 0.0025750 | 0.0104061 | 0.0018296 | 0.0229226 | 2.4160572 | −4.041212 | −0.710543 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 10 | −0.5893030 | 0.3404390 | 2.5305614 | 0.0077481 | −0.0268415 | 0.0124622 | 0.0077481 | 0.0268415 | 0.0316974 | 2.3756764 | −0.621725 | 2.153833 | 0.416203 | −0.491692 | −0.491692 | 0.585882 |
| 11 | −0.5770101 | 0.3439840 | 2.4991563 | 0.0122929 | 0.0035449 | 0.0025750 | 0.0122929 | 0.0035449 | 0.0314051 | 2.3232545 | −4.773959 | −1.376677 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 12 | −0.5695393 | 0.3173026 | 2.4684556 | 0.0074708 | −0.0266814 | 0.0120162 | 0.0074708 | 0.0266814 | 0.0307007 | 2.3520994 | −0.621725 | 2.220446 | 0.431095 | −0.493988 | −0.493988 | 0.571064 |
| 13 | −0.5573035 | 0.3209047 | 2.4358394 | 0.0122358 | 0.0036021 | 0.0025750 | 0.0122358 | 0.0036021 | 0.0326162 | 2.3328404 | −4.751755 | −1.398881 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 14 | −0.5431645 | 0.2843790 | 2.3926964 | 0.0141390 | −0.0365258 | 0.0176879 | 0.0141390 | 0.0365258 | 0.0431429 | 2.4224471 | −0.799361 | 2.065015 | 0.450025 | −0.496254 | −0.496254 | 0.552220 |
| 15 | −0.5292707 | 0.2898679 | 2.3481564 | 0.0138939 | 0.0054889 | 0.0025750 | 0.0138939 | 0.0054889 | 0.0445400 | 2.2642948 | −5.395684 | −2.131628 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 16 | −0.5005418 | 0.2343254 | 2.2779237 | 0.0287289 | −0.0555425 | 0.0287519 | 0.0287289 | 0.0555425 | 0.0702328 | 2.4859809 | −0.999201 | 1.931788 | 0.486741 | −0.498561 | −0.498561 | 0.515716 |
| 17 | −0.4844752 | 0.2426732 | 2.2099979 | 0.0160666 | 0.0083478 | 0.0025750 | 0.0160666 | 0.0083478 | 0.0679258 | 2.1980430 | −6.239453 | −3.241851 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 18 | −0.3555181 | 0.0952936 | 1.9341471 | 0.1289571 | −0.1473795 | 0.0921859 | 0.1289571 | 0.1473795 | 0.2758508 | 2.5515119 | −1.398881 | 1.598721 | 0.632998 | −0.480458 | −0.480458 | 0.371010 |
| 19 | −0.3270076 | 0.1201626 | 1.7784488 | 0.0285105 | 0.0248690 | 0.0040000 | 0.0285105 | 0.0248690 | 0.1556982 | 2.0721635 | −7.127632 | −6.217249 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 20 | −0.3180978 | 0.0948121 | 1.7414446 | 0.0089098 | −0.0253505 | 0.0095538 | 0.0089098 | 0.0253505 | 0.0370043 | 2.1984509 | −0.932587 | 2.653433 | 0.707340 | −0.453846 | −0.453846 | 0.296194 |
| 21 | −0.2988375 | 0.1019571 | 1.7029892 | 0.0192602 | 0.0071449 | 0.0055962 | 0.0192602 | 0.0071449 | 0.0384553 | 2.4575669 | −3.441691 | −1.276756 | 0.727097 | −0.444344 | −0.444344 | 0.276514 |
| 22 | −0.1083749 | −0.0156884 | 1.3037546 | 0.1904627 | −0.1176454 | 0.0994133 | 0.1904627 | 0.1176454 | 0.3992346 | 2.4445109 | −1.088019 | 2.531308 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 23 | −0.0893010 | 0.0150037 | 1.1915175 | 0.0190738 | 0.0306921 | 0.0055962 | 0.0190738 | 0.0306921 | 0.1122371 | 1.9603859 | −3.408385 | −5.484502 | 0.956433 | −0.203614 | −0.203614 | 0.048388 |
| 24 | 0.0732505 | −0.0201455 | 0.9239465 | 0.1625516 | −0.0351492 | 0.0765273 | 0.1625516 | 0.0351492 | 0.2675710 | 0.0554311 | −1.920686 | 1.409983 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 25 | 0.0794635 | 0.0084031 | 0.8478237 | 0.0062130 | 0.0285486 | 0.0055962 | 0.0062130 | 0.0285486 | 0.0761229 | 1.2511533 | −1.110223 | −5.101475 | 0.979303 | 0.142012 | 0.142012 | 0.025604 |
| 26 | 0.2332562 | 0.0305274 | 0.6449268 | 0.1537927 | 0.0221242 | 0.0851801 | 0.1537927 | 0.0221242 | 0.2028969 | 0.0127422 | −1.904032 | 0.416334 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 27 | 0.2298355 | 0.0538750 | 0.5932637 | −0.0034207 | 0.0233476 | 0.0048906 | 0.0034207 | 0.0233476 | 0.0516631 | 1.4860233 | 0.699441 | −4.773959 | 0.819973 | 0.383247 | 0.383247 | 0.184135 |
| 28 | 0.3819893 | 0.1248658 | 0.4262477 | 0.1521537 | 0.0709908 | 0.1208046 | 0.1521537 | 0.0709908 | 0.1670160 | 0.1206159 | −1.632028 | 0.205391 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 29 | 0.3758359 | 0.1379665 | 0.3906607 | −0.0061534 | 0.0131007 | 0.0031094 | 0.0061534 | 0.0131007 | 0.0355870 | 1.6580890 | 1.978973 | −4.213296 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 30 | 0.5658518 | 0.3035318 | 0.2162284 | 0.1900159 | 0.1655654 | 0.2516929 | 0.1900159 | 0.1655654 | 0.1744323 | 0.2243899 | −0.754952 | −0.657807 | 0.407382 | 0.485704 | 0.485704 | 0.601922 |
| 31 | 0.5583974 | 0.3121012 | 0.1950214 | −0.0074544 | 0.0085693 | 0.0025750 | 0.0074544 | 0.0085693 | 0.0212070 | 1.7770577 | 2.894907 | −3.327894 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 32 | 0.5621889 | 0.3118625 | 0.1934374 | 0.0037914 | −0.0002387 | 0.0040000 | 0.0037914 | 0.0002387 | 0.0015840 | 0.5693412 | −0.947853 | 0.059674 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 33 | 0.5619113 | 0.3152153 | 0.1919497 | −0.0002776 | 0.0033529 | 0.0040000 | 0.0002776 | 0.0033529 | 0.0014877 | 1.1421631 | 0.069389 | −0.838218 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 34 | 0.5656910 | 0.3157484 | 0.1904373 | 0.0037797 | 0.0005330 | 0.0049882 | 0.0037797 | 0.0005330 | 0.0015124 | 0.3689879 | −0.757727 | −0.106859 | 0.434662 | 0.494456 | 0.494456 | 0.567539 |
| 35 | 0.5653024 | 0.3191512 | 0.1889793 | −0.0003886 | 0.0034028 | 0.0040000 | 0.0003886 | 0.0034028 | 0.0014580 | 1.1705379 | 0.097145 | −0.850708 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 36 | 0.5690896 | 0.3195584 | 0.1875367 | 0.0037872 | 0.0004072 | 0.0048906 | 0.0037872 | 0.0004072 | 0.0014426 | 0.4045337 | −0.774381 | −0.083267 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 37 | 0.5686122 | 0.3230001 | 0.1861057 | −0.0004774 | 0.0034417 | 0.0040000 | 0.0004774 | 0.0034417 | 0.0014310 | 1.1920311 | 0.119349 | −0.860423 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 38 | 0.5717652 | 0.3232555 | 0.1847246 | 0.0031530 | 0.0002554 | 0.0040000 | 0.0031530 | 0.0002554 | 0.0013811 | 0.4337529 | −0.788258 | −0.063838 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 39 | 0.5718740 | 0.3273483 | 0.1833014 | 0.0001087 | 0.0040928 | 0.0055962 | 0.0001087 | 0.0040928 | 0.0014232 | 1.0243490 | −0.019429 | −0.731359 | 0.431186 | 0.494000 | 0.494000 | 0.570973 |
| 40 | 0.6890449 | 0.4614914 | 0.1143594 | 0.1171709 | 0.1341431 | 0.3170546 | 0.1171709 | 0.1341431 | 0.0689420 | 0.2626918 | −0.927036 | 0.061062 | 0.180555 | 0.220381 | 0.220381 | 0.272256 |
| 41 | 0.6829678 | 0.4668038 | 0.1005223 | −0.0060771 | 0.0053124 | 0.0020000 | 0.0060771 | 0.0053124 | 0.0138371 | 1.8236703 | 3.038542 | −2.656209 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 42 | 0.6848511 | 0.4666198 | 0.0998954 | 0.0018833 | −0.0001840 | 0.0025750 | 0.0018833 | 0.0001840 | 0.0006269 | 0.6955094 | −0.731359 | 0.071471 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 43 | 0.6847400 | 0.4685377 | 0.0993998 | −0.0001110 | 0.0019179 | 0.0040000 | 0.0001110 | 0.0019179 | 0.0004956 | 1.0285375 | 0.027756 | −0.479478 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 44 | 0.6864165 | 0.4687426 | 0.0989227 | 0.0016764 | 0.0002050 | 0.0031094 | 0.0016764 | 0.0002050 | 0.0004772 | 0.4774846 | −0.539152 | −0.065919 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 45 | 0.6862638 | 0.4706800 | 0.0984381 | −0.0001527 | 0.0019373 | 0.0040000 | 0.0001527 | 0.0019373 | 0.0004845 | 1.0482585 | 0.038164 | −0.484335 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 46 | 0.6879769 | 0.4708526 | 0.0979634 | 0.0017131 | 0.0001726 | 0.0031094 | 0.0017131 | 0.0001726 | 0.0004747 | 0.4997637 | −0.550948 | −0.055511 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 47 | 0.6877687 | 0.4728205 | 0.0974926 | −0.0002082 | 0.0019679 | 0.0040000 | 0.0002082 | 0.0019679 | 0.0004708 | 1.0739201 | 0.052042 | −0.491968 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 48 | 0.6895315 | 0.4729499 | 0.0970176 | 0.0017627 | 0.0001295 | 0.0031094 | 0.0017627 | 0.0001295 | 0.0004750 | 0.5278925 | −0.566908 | −0.041633 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 49 | 0.6892484 | 0.4749511 | 0.0965678 | −0.0002831 | 0.0020012 | 0.0040000 | 0.0002831 | 0.0020012 | 0.0004497 | 1.1079707 | 0.070777 | −0.500294 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 50 | 0.6907671 | 0.4750083 | 0.0960876 | 0.0015188 | 0.0000572 | 0.0025750 | 0.0015188 | 0.0000572 | 0.0004802 | 0.5647621 | −0.589806 | −0.022204 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 51 | 0.6909069 | 0.4774119 | 0.0955389 | 0.0001398 | 0.0024036 | 0.0055962 | 0.0001398 | 0.0024036 | 0.0005487 | 0.9080480 | −0.024980 | −0.429518 | 0.343417 | 0.473660 | 0.473660 | 0.658301 |
| 52 | 0.7911469 | 0.6156390 | 0.0541759 | 0.1002400 | 0.1382271 | 0.4730887 | 0.1002400 | 0.1382271 | 0.0413630 | 0.2820867 | −0.634215 | 0.012490 | 0.204618 | 0.294968 | 0.294968 | 0.428249 |
| 53 | 0.7883134 | 0.6176922 | 0.0462143 | −0.0028335 | 0.0020532 | 0.0010000 | 0.0028335 | 0.0020532 | 0.0079616 | 1.8498943 | 2.833497 | −2.053219 | 0.283316 | 0.449476 | 0.449476 | 0.718106 |
| 54 | 0.8798128 | 0.7656520 | 0.0215323 | 0.0914994 | 0.1479597 | 0.7443176 | 0.0914994 | 0.1479597 | 0.0246820 | 0.3008214 | 0.753911 | −0.748707 | 0.210356 | 0.345469 | 0.345469 | 0.569840 |
| 55 | 0.8770903 | 0.7673345 | 0.0154882 | −0.0027225 | 0.0016825 | 0.0010000 | 0.0027225 | 0.0016825 | 0.0060441 | 1.8692851 | 2.722475 | −1.682508 | 0.243519 | 0.428130 | 0.428130 | 0.757700 |
| 56 | 0.9498433 | 0.8976073 | 0.0046272 | 0.0727530 | 0.1302728 | 1.2109382 | 0.0727530 | 0.1302728 | 0.0108610 | 0.3043452 | 0.439492 | −0.390313 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 57 | 0.9482040 | 0.8985255 | 0.0027148 | −0.0016394 | 0.0009183 | 0.0010000 | 0.0016394 | 0.0009183 | 0.0019124 | 1.8724909 | 1.639357 | −0.918276 | 0.218163 | 0.411968 | 0.411968 | 0.782924 |
| 58 | 0.9896615 | 0.9778590 | 0.0003537 | 0.0414576 | 0.0793334 | 1.8596829 | 0.0414576 | 0.0793334 | 0.0023611 | 0.3098536 | 0.111109 | −0.112952 | 0.421908 | 0.820498 | 0.820498 | 1.602070 |
| 59 | 0.9890613 | 0.9781729 | 0.0001201 | −0.0006003 | 0.0003140 | 0.0010000 | 0.0006003 | 0.0003140 | 0.0002336 | 1.8798321 | 0.600263 | −0.313952 | 0.203746 | 0.401776 | 0.401776 | 0.797271 |
| 60 | 0.9998972 | 0.9997095 | 0.0000007 | 0.0108359 | 0.0215366 | 2.4418417 | 0.0108359 | 0.0215366 | 0.0001194 | 0.3193359 | 0.005530 | −0.013849 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 61 | 0.9998635 | 0.9997264 | 0.0000000 | −0.0000337 | 0.0000170 | 0.0010000 | 0.0000337 | 0.0000170 | 0.0000007 | 1.8901811 | 0.033723 | −0.016964 | 0.200160 | 0.399120 | 0.399120 | 0.800840 |

## Функция Розенброка (BFGS, метод парабол)

| № | $x_i$ | $y_i$ | $f_i$ | $s_1$ | $s_2$ | $\lambda$ | $\lvert x_i - x_{i-1}\rvert$ | $\lvert y_i - y_{i-1}\rvert$ | $\lvert f_i - f_{i-1}\rvert$ | angle | $\nabla_x$ | $\nabla_y$ | $H_{11}$ | $H_{12}$ | $H_{21}$ | $H_{22}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −1.0000000 | 2.0000000 | 104.0000000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 1 | −1.3111075 | 1.8416586 | 6.8456618 | −0.3111075 | −0.1583414 | 0.0007847 | 0.3111075 | 0.1583414 | 97.1543382 | 1.4229076 | 396.482847 | 201.794137 | 0.217729 | −0.411832 | −0.411832 | 0.783188 |
| 2 | −1.4177628 | 2.0396104 | 5.9329508 | −0.1066553 | 0.1979519 | 0.0366587 | 0.1066553 | 0.1979519 | 0.9127110 | 0.1132403 | 59.729999 | 24.513724 | 0.016286 | −0.036351 | −0.036351 | 0.083308 |
| 3 | −1.4244506 | 2.0362292 | 5.8831012 | −0.0066878 | −0.0033812 | 0.0005598 | 0.0066878 | 0.0033812 | 0.0498496 | 1.4284689 | 11.946000 | 6.039613 | 0.113843 | −0.316856 | −0.316856 | 0.886705 |
| 4 | −1.2303385 | 1.4935212 | 5.0152612 | 0.1941121 | −0.5427080 | 0.3463768 | 0.1941121 | 0.5427080 | 0.8678400 | 2.7960109 | −0.843769 | 1.465494 | 0.112633 | −0.311384 | −0.311384 | 0.861946 |
| 5 | −1.2199205 | 1.4964563 | 4.9348538 | 0.0104180 | 0.0029351 | 0.0007263 | 0.0104180 | 0.0029351 | 0.0804074 | 1.9801251 | −14.344081 | −4.041212 | 0.144289 | −0.350515 | −0.350515 | 0.856422 |
| 6 | −1.0158460 | 0.9980654 | 4.1784042 | 0.2040745 | −0.4983909 | 0.3174922 | 0.2040745 | 0.4983909 | 0.7564496 | 2.7360065 | −0.355271 | 1.687539 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 7 | −0.9979960 | 1.0048760 | 3.9998734 | 0.0178500 | 0.0068106 | 0.0010024 | 0.0178500 | 0.0068106 | 0.1785308 | 1.9882621 | −17.807977 | −6.794565 | 0.196364 | −0.396251 | −0.396251 | 0.804620 |
| 8 | −0.8271766 | 0.6582647 | 3.4059480 | 0.1708193 | −0.3466113 | 0.2171485 | 0.1708193 | 0.3466113 | 0.5939254 | 2.7008619 | −0.466294 | 1.754152 | 0.108447 | −0.208211 | −0.208211 | 0.402435 |
| 9 | −0.8101351 | 0.6655020 | 3.2850220 | 0.0170416 | 0.0072372 | 0.0013929 | 0.0170416 | 0.0072372 | 0.1209260 | 2.0522948 | −12.234658 | −5.195844 | 0.269347 | −0.442505 | −0.442505 | 0.732005 |
| 10 | −0.6653161 | 0.4262293 | 2.8002272 | 0.1448189 | −0.2392726 | 0.1479043 | 0.1448189 | 0.2392726 | 0.4847948 | 2.6848330 | −0.643929 | 1.820766 | 0.155038 | −0.242039 | −0.242039 | 0.380445 |
| 11 | −0.6503674 | 0.4326236 | 2.7330168 | 0.0149488 | 0.0063943 | 0.0019458 | 0.0149488 | 0.0063943 | 0.0672104 | 2.1504130 | −7.682743 | −3.286260 | 0.366541 | −0.480658 | −0.480658 | 0.635285 |
| 12 | −0.4719954 | 0.1973131 | 2.2316249 | 0.1783720 | −0.2353105 | 0.1460237 | 0.1783720 | 0.2353105 | 0.5013919 | 2.6153767 | −0.799361 | 1.931788 | 0.116182 | −0.139691 | −0.139691 | 0.170918 |
| 13 | −0.4496936 | 0.2120528 | 2.1112714 | 0.0223018 | 0.0147397 | 0.0028862 | 0.0223018 | 0.0147397 | 0.1203535 | 2.1169501 | −7.727152 | −5.107026 | 0.536004 | −0.497449 | −0.497449 | 0.466686 |
| 14 | −0.3032945 | 0.0752750 | 1.7265076 | 0.1463991 | −0.1367778 | 0.0927165 | 0.1463991 | 0.1367778 | 0.3847638 | 2.6334332 | −1.132427 | 1.953993 | 0.160519 | −0.133368 | −0.133368 | 0.113663 |
| 15 | −0.2828114 | 0.0901093 | 1.6558608 | 0.0204832 | 0.0148343 | 0.0044244 | 0.0204832 | 0.0148343 | 0.0706468 | 2.2063449 | −4.629630 | −3.352874 | 0.740343 | −0.437349 | −0.437349 | 0.263361 |
| 16 | −0.1174431 | −0.0084460 | 1.2981359 | 0.1653682 | −0.0985553 | 0.0852126 | 0.1653682 | 0.0985553 | 0.3577249 | 2.5323462 | −1.421085 | 2.031708 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 17 | −0.0985302 | 0.0171760 | 1.2123453 | 0.0189130 | 0.0256220 | 0.0057552 | 0.0189130 | 0.0256220 | 0.0857906 | 2.0340853 | −3.286260 | −4.451994 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 18 | −0.0620925 | −0.0115906 | 1.1518986 | 0.0364377 | −0.0287666 | 0.0191931 | 0.0364377 | 0.0287666 | 0.0604468 | 2.2887607 | −1.898481 | 1.498801 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 19 | −0.0445219 | 0.0101006 | 1.0976168 | 0.0175707 | 0.0216912 | 0.0070028 | 0.0175707 | 0.0216912 | 0.0542818 | 2.0285344 | −2.509104 | −3.097522 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 20 | −0.0181677 | −0.0118863 | 1.0515894 | 0.0263542 | −0.0219869 | 0.0135644 | 0.0263542 | 0.0219869 | 0.0460274 | 1.8669314 | −1.942890 | 1.620926 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 21 | −0.0001744 | 0.0088389 | 1.0081614 | 0.0179932 | 0.0207252 | 0.0084853 | 0.0179932 | 0.0207252 | 0.0434279 | 0.7346874 | −2.120526 | −2.442491 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 22 | 0.0209699 | −0.0099429 | 0.9692800 | 0.0211443 | −0.0187818 | 0.0106397 | 0.0211443 | 0.0187818 | 0.0388815 | 0.2835384 | −1.987299 | 1.765255 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 23 | 0.0398744 | 0.0110372 | 0.9307661 | 0.0189045 | 0.0209801 | 0.0101055 | 0.0189045 | 0.0209801 | 0.0385139 | 0.5673534 | −1.870726 | −2.076117 | 0.996946 | 0.055036 | 0.055036 | 0.008068 |
| 24 | 0.1864994 | 0.0184215 | 0.6885501 | 0.1466249 | 0.0073843 | 0.0747945 | 0.1466249 | 0.0073843 | 0.2422160 | 0.0481363 | −2.070566 | 1.887379 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 25 | 0.1887596 | 0.0366890 | 0.6582231 | 0.0022602 | 0.0182676 | 0.0055776 | 0.0022602 | 0.0182676 | 0.0303270 | 1.2557178 | −0.405231 | −3.275158 | 0.877123 | 0.327474 | 0.327474 | 0.127261 |
| 26 | 0.3004351 | 0.0783006 | 0.5036969 | 0.1116755 | 0.0416115 | 0.0783298 | 0.1116755 | 0.0416115 | 0.1545262 | 0.1017224 | −1.704192 | 0.210942 | 0.198257 | 0.089649 | 0.089649 | 0.043944 |
| 27 | 0.3002450 | 0.0900348 | 0.4896584 | −0.0001901 | 0.0117342 | 0.0048932 | 0.0001901 | 0.0117342 | 0.0140385 | 1.2956602 | 0.038858 | −2.398082 | 0.734847 | 0.440313 | 0.440313 | 0.268818 |
| 28 | 0.4195472 | 0.1615324 | 0.3579143 | 0.1193023 | 0.0714976 | 0.1161045 | 0.1193023 | 0.0714976 | 0.1317441 | 0.1723804 | −1.385003 | −0.022204 | 0.171959 | 0.117304 | 0.117304 | 0.083462 |
| 29 | 0.4154984 | 0.1707728 | 0.3419903 | −0.0040488 | 0.0092405 | 0.0031920 | 0.0040488 | 0.0092405 | 0.0159239 | 1.5938008 | 1.268430 | −2.894907 | 0.584584 | 0.491556 | 0.491556 | 0.418349 |
| 30 | 0.5384973 | 0.2745355 | 0.2368361 | 0.1229989 | 0.1037626 | 0.1794155 | 0.1229989 | 0.1037626 | 0.1051543 | 0.2292997 | −0.857647 | −0.374700 | 0.161555 | 0.148938 | 0.148938 | 0.140858 |
| 31 | 0.5327625 | 0.2818993 | 0.2186859 | −0.0057348 | 0.0073638 | 0.0023859 | 0.0057348 | 0.0073638 | 0.0181501 | 1.7457846 | 2.403633 | −3.086420 | 0.460480 | 0.497183 | 0.497183 | 0.541833 |
| 32 | 0.6400779 | 0.3982497 | 0.1426542 | 0.1073154 | 0.1163504 | 0.2475990 | 0.1073154 | 0.1163504 | 0.0760317 | 0.2691957 | −0.523193 | −0.387190 | 0.194195 | 0.222657 | 0.222657 | 0.258812 |
| 33 | 0.6358040 | 0.4026766 | 0.1328853 | −0.0042739 | 0.0044269 | 0.0019345 | 0.0042739 | 0.0044269 | 0.0097689 | 1.7740390 | 2.209344 | −2.288447 | 0.377731 | 0.483603 | 0.483603 | 0.624163 |
| 34 | 0.7188089 | 0.5094186 | 0.0843502 | 0.0830049 | 0.1067421 | 0.3002665 | 0.0830049 | 0.1067421 | 0.0485350 | 0.2933107 | −0.330291 | −0.313638 | 0.247127 | 0.328742 | 0.328742 | 0.440538 |
| 35 | 0.7162635 | 0.5118501 | 0.0806464 | −0.0025454 | 0.0024315 | 0.0016742 | 0.0025454 | 0.0024315 | 0.0037038 | 1.7586011 | 1.520312 | −1.452310 | 0.326889 | 0.467903 | 0.467903 | 0.674744 |
| 36 | 0.7987257 | 0.6304117 | 0.0462133 | 0.0824622 | 0.1185615 | 0.4454719 | 0.0824622 | 0.1185615 | 0.0344331 | 0.2949166 | −0.227596 | −0.236616 | 0.237691 | 0.352099 | 0.352099 | 0.524400 |
| 37 | 0.7958762 | 0.6325572 | 0.0417408 | −0.0028496 | 0.0021456 | 0.0014220 | 0.0028496 | 0.0021456 | 0.0044725 | 1.8246523 | 2.003953 | −1.508862 | 0.282051 | 0.448872 | 0.448872 | 0.719358 |
| 38 | 0.8743110 | 0.7579699 | 0.0199579 | 0.0784349 | 0.1254126 | 0.6825252 | 0.0784349 | 0.1254126 | 0.0217829 | 0.2976557 | −0.133574 | −0.172085 | 0.246819 | 0.402183 | 0.402183 | 0.657485 |
| 39 | 0.8718328 | 0.7595637 | 0.0164548 | −0.0024782 | 0.0015939 | 0.0012364 | 0.0024782 | 0.0015939 | 0.0035031 | 1.8533565 | 2.004300 | −1.289073 | 0.246026 | 0.429615 | 0.429615 | 0.755205 |
| 40 | 0.9331901 | 0.8672209 | 0.0057760 | 0.0613572 | 0.1076572 | 0.9712604 | 0.0613572 | 0.1076572 | 0.0106788 | 0.3040063 | −0.071991 | −0.105818 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 41 | 0.9318287 | 0.8680302 | 0.0046549 | −0.0013613 | 0.0008092 | 0.0011178 | 0.0013613 | 0.0008092 | 0.0011211 | 1.8553065 | 1.217863 | −0.723943 | 0.223001 | 0.415217 | 0.415217 | 0.778114 |
| 42 | 0.9722063 | 0.9435758 | 0.0010315 | 0.0403775 | 0.0755457 | 1.3294809 | 0.0403775 | 0.0755457 | 0.0036234 | 0.3094931 | −0.034044 | −0.054861 | 0.429853 | 0.821671 | 0.821671 | 1.576775 |
| 43 | 0.9716096 | 0.9439126 | 0.0008073 | −0.0005966 | 0.0003368 | 0.0010472 | 0.0005966 | 0.0003368 | 0.0002242 | 1.8567892 | 0.569759 | −0.321596 | 0.209296 | 0.405789 | 0.405789 | 0.791749 |
| 44 | 0.9960044 | 0.9914413 | 0.0000500 | 0.0243947 | 0.0475287 | 2.0588935 | 0.0243947 | 0.0475287 | 0.0007573 | 0.3134925 | −0.012951 | −0.022519 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 45 | 0.9957787 | 0.9915585 | 0.0000178 | −0.0002257 | 0.0001172 | 0.0010050 | 0.0002257 | 0.0001172 | 0.0000322 | 1.8794095 | 0.224531 | −0.116582 | 0.201018 | 0.399758 | 0.399758 | 0.799987 |
| 46 | 0.9999983 | 0.9999914 | 0.0000000 | 0.0042195 | 0.0084329 | 2.4885046 | 0.0042195 | 0.0084329 | 0.0000178 | 0.3214603 | −0.001784 | −0.003345 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| 47 | 0.9999962 | 0.9999924 | 0.0000000 | −0.0000020 | 0.0000010 | 0.0009992 | 0.0000020 | 0.0000010 | 0.0000000 | 1.8918663 | 0.002039 | −0.001021 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |

## Вывод

BFGS соверашает меньшее количество итераций, чем метод многогранников и производит меньше вычислений функций (но это связано с неоптимальной реализацией алгоритма, в теории должно быть меньше вычислений). Правильно выбранная начальная точка может сильно сократить объём вычислений для BFGS, но не для метода многогранников. Точность вычислений не влияет на количество итераций у BFGS, но увеличение точности для метода многогранников увеличивает количество итераций. Метод парабол позволяет сойтись BFGS за меньшее количество итераций, но требует больше вычислений целевой функции.

## Листинги

### simplex_algorithm.py

```
import numpy as np
from math import sqrt

#---------------------------------------------------
# General computation class
#---------------------------------------------------
class SimplexMethod:
    iters = 0
    calcs = 0

    def __init__(self, n, f, t=1., max_iters=100, x0=None, alpha=1.,
    ↪  beta=.5, gamma=2.0, omega=.5, eps=1e-7):
        # n is number of dimensions
        # f is function taken numpy array as arg with n variables
        # t is edge length
        # max_iters is max possible iterations
        self.n = n
        self.f = f
        self.max_iters = max_iters
        self.alpha = alpha
        self.beta = beta
        self.gamma = gamma
        self.omega = omega
        self.eps = eps

        # Compute simplex coeficients
        d1 = t / (n * sqrt(2.)) * (sqrt(n + 1.) + n - 1.)
        d2 = t / (n * sqrt(2.) * (sqrt(n + 1.) - 1.))

        # Setup begining coordinates
        self.x = np.array([[d1 if j == i else d2 for j in range(n+1)] for
        ↪  i in range(1,n+1)])
        self.x[:,0] = np.zeros(n)

```

```
        if not x0 is None:
            self.x[0,0] = x0[0]
            self.x[0,1] = x0[1]

        self.finished = False
        self.x_min = self.x[:,0]

    def __iter__(self):
        self.calcs = 0
        self.iters = 0
        return self

    def __next__(self):
        self.iters += 1

        if self.finished:
            raise StopIteration

        # Sort verteces by function values
        self.x_res = self.f(self.x)
        self.calcs += self.n
        self.x = self.x.take(self.x_res.argsort(), 1)
        self.x_res = self.x_res.take(self.x_res.argsort())
        self.s_k = self.x_min - self.x[:,0]
        self.x_min = self.x[:,0]


        # Calculate centroid
        x_o = np.sum(self.x[:,:-1], 1) / self.n

        self.calcs += 1
        if np.sqrt(np.sum(np.power(self.x_res - self.f(x_o), 2)) / (self.n
        ↪  + 1)) < self.eps:
            self.finished = True
            return self.x_min

        # Calculate reflected point
        x_r = x_o + self.alpha * (x_o - self.x[:,-1])
        self.calcs += 1
        x_r_res = self.f(x_r)

        if self.x_res[0] ⩽ x_r_res and x_r_res < self.x_res[-2]:
            self.x[:,-1] = x_r
            return self.termination()

        # Calculate expanded point
        elif x_r_res < self.x_res[0]:
            x_e = x_o + self.gamma * (x_r - x_o)

```

```
            self.calcs += 1
            if self.f(x_e) < x_r_res:
                self.x[:,-1] = x_e
                return self.termination()

            else:
                self.x[:,-1] = x_r
                return self.termination()

        # Calculate contracted point
        else:
            if x_r_res < self.x_res[-1]:
                x_c = x_o + self.beta * (x_r - x_o)

            else:
                x_c = x_o + self.beta * (self.x[:,-1] - x_o)

            self.calcs += 1
            if self.f(x_c) < x_r_res:
                self.x[:,-1] = x_c
                return self.termination()

        # Shrink
        self.x[:,1:] = self.x[:,0] + self.omega * (self.x[:,1:] -
        ↪  self.x[:,0])

        return self.termination()

    def termination(self):
        if self.iters > self.max_iters:
            raise StopIteration
        return self.x_min
```

**BFGS_algorithm.py**

```
import numpy as np

from linear_search import FibonacciSearch, QuadraticInterpolationSearch
from interval_search import IntervalSearch

class BFGSMethod:
    iters = 0
    calcs = 0

    def __init__(self, n, f, eps=1e-7, h=1e-14, x0=None, max_iters=5):
        # n is number of dimensions
        # f is function taken numpy array as arg with n variables
        # t is edge length
```

```
        # max_iters is max possible iterations
        self.n = n
        self.f = f
        self.eps = eps
        self.h = h
        self.max_iters = max_iters

        self.H_k = np.eye(n) # approx matrix

        if x0 is None:
            self.x_k = np.ones(n)

        elif x0.shape ≠ (n,):
            raise ValueError('Initial approx vector has invalid shape:
            ↪  needed ({}, 1), but {} was given'.format(n, x0.shape))

        else:
            self.x_k = x0


    def __iter__(self):
        self.calcs = 0
        self.iters = 0
        return self


    def __next__(self):
        self.iters += 1

        if self.iters % self.n == 0:
            self.H_k = np.eye(self.n)

        grad_xk = self.grad(self.x_k, self.h)
        self.g = grad_xk

        if np.linalg.norm(grad_xk) < self.eps:
            raise StopIteration

        if self.iters ⩾ self.max_iters:
            raise StopIteration

        p_k = -np.dot(self.H_k, grad_xk)

        interval = IntervalSearch(0., p_k, self.x_k, self.f, eps=self.eps)
        a, b = interval.compute()

        minimizer = FibonacciSearch(a, b, p_k, self.x_k, self.f,
        ↪  eps=self.eps)
        self.alpha = minimizer.compute()
```

```
        print(self.alpha)

        x_k1 = self.x_k + self.alpha * p_k

        self.calcs += interval.calcs
        self.calcs += minimizer.calcs

        grad_xk1 = self.grad(x_k1, self.h)

        self.s_k = x_k1 - self.x_k
        y_k = grad_xk1 - grad_xk

        if np.equal(self.s_k, self.alpha * p_k).all():
            self.H_k = np.eye(self.n)

        btm = np.dot((self.s_k - self.H_k @ y_k), y_k)

        if btm == 0.:
            self.x_k = x_k1
            self.H_k = np.eye(self.n)

            return self.x_k

        self.H_k += (self.s_k - self.H_k @ y_k)[:, np.newaxis] * \
                (self.s_k - self.H_k @ y_k)[np.newaxis] / btm

        self.x_k = x_k1

        return self.x_k

    def grad(self, x, h):
        n = np.size(x, 0)
        g = np.zeros(n)
        hx = np.zeros(n)

        for i in range(0, n):
            hx[i] = h
            g[i] = (self.f(x + hx) - self.f(x - hx)) / (2. * h)
            hx[i] = 0.
            self.calcs += 2


        return g
```

### linear_search.py

```
from math import log, sqrt, fabs

```

```
class QuadraticInterpolationSearch:
    iters = 0
    calcs = 0

    def __init__(self, a0, b0, dx, X0, f, eps=1e-7, max_iters=10000):
        self.f = f
        self.ai = a0
        self.bi = b0
        self.dx = dx
        self.X0 = X0
        self.eps = eps
        self.max_iters = max_iters

        self.x0 = (a0 + b0) / 2.
        self.h = (b0 - a0) / 2.
        self.xk = 0.0


    def __iter__(self):
        self.iters = 0
        return self

    def __next__(self):
        self.iters += 1

        if (fabs(self.xk - self.x0) < self.eps):
            raise StopIteration

        else:
            self.x0 = self.xk

        self.x1 = self.x0 - self.h
        self.x2 = self.x0 + self.h

        c = (self.f(self.X0 + self.x1 * self.dx) - \
            2. * self.f(self.X0 + self.x0 * self.dx) + \
            self.f(self.X0 + self.x2 * self.dx)) / \
            (2. * self.h * self.h)

        b = (-self.f(self.X0 + self.x1 * self.dx) * \
            (2. * self.x0 + self.h) + 4. * self.f(self.X0 + self.x0 *
            ↪  self.dx) * \
            self.x0 - self.f(self.X0 + self.x2 * self.dx) * \
            (2. * self.x0 - self.h)) / (2. * self.h * self.h)

        self.xk = -b / (2. * c)

        if self.iters ⩾ self.max_iters:
            raise StopIteration
```

```
        return self.iters

    def compute(self):
        for i in self:
            pass
        return (self.ai + self.bi) / 2.


class FibonacciSearch:
    iters = 0
    Fib = lambda n: (((1. + sqrt(5.)) / 2.) ** n - ((1. - sqrt(5.)) / 2.) ** n) / sqrt(5.)

    def __init__(self, a0, b0, dx, X0, f, eps=1e-7, max_iters=10000):
        self.f = f
        self.ai = a0
        self.bi = b0
        self.dx = dx
        self.X0 = X0
        self.eps = eps
        self.delta = eps / 10.

        n = 0

        while (b0 - a0) / eps ⩾ FibonacciSearch.Fib(n + 2):
            n += 1

        self.max_iters = max_iters if max_iters < n else n

        self.x1 = self.ai + FibonacciSearch.Fib(self.max_iters-2) * \
                (self.bi - self.ai) / FibonacciSearch.Fib(self.max_iters)
        self.x2 = self.ai + FibonacciSearch.Fib(self.max_iters-1) * \
                (self.bi - self.ai) / FibonacciSearch.Fib(self.max_iters)

        self.f_x1 = self.f(X0 + self.x1 * dx)
        self.f_x2 = self.f(X0 + self.x2 * dx)

        self.calcs = 2

    def __iter__(self):
        self.iters = 0
        return self

    def __next__(self):
        self.iters += 1

        # Algorithm from Wikipedia
        if self.f_x1 < self.f_x2:
            self.bi = self.x2
```

```
            self.x2 = self.x1
            self.x1 = self.ai + FibonacciSearch.Fib(self.max_iters -
            ↪  self.iters + 0) / \
                    FibonacciSearch.Fib(self.max_iters - self.iters + 2) *
                    ↪  (self.bi - self.ai)
            self.f_x2 = self.f_x1
            self.f_x1 = self.f(self.X0 + self.x1 * self.dx)

        else:
            self.ai = self.x1
            self.x1 = self.x2
            self.x2 = self.ai + FibonacciSearch.Fib(self.max_iters -
            ↪  self.iters + 1) / \
                    FibonacciSearch.Fib(self.max_iters - self.iters + 2) *
                    ↪  (self.bi - self.ai)
            self.f_x1 = self.f_x2
            self.f_x2 = self.f(self.X0 + self.x2 * self.dx)

        self.calcs += 1

        if self.iters ⩾ self.max_iters:
            raise StopIteration
        return (self.ai + self.bi) / 2.

    def compute(self):
        for i in self:
            pass
        return (self.ai + self.bi) / 2.
```

**interval_search.py**

```
import numpy as np

class IntervalSearch:
    iters = 0

    def __init__(self, x0, dx, X0, f, eps=1e-7, max_iters=100):
        self.f = f
        self.xk1 = x0
        self.eps = eps
        self.max_iters = max_iters
        self.dx = dx
        self.X0 = X0

        self.h = eps if f(X0 + x0 * dx) > f(X0 + (x0 + eps) * dx) else -eps
        self.calcs = 2

        self.xk0 = x0
```

```
        self.xk2 = self.xk1 + self.h

    def __iter__(self):
        self.iters = 0
        return self

    def __next__(self):
        self.calcs += 2
        if self.f(self.X0 + self.xk1 * self.dx) ⩽ \
                self.f(self.X0 + (self.xk2 + self.eps) * self.dx):
            raise StopIteration

        if self.iters ⩾ self.max_iters:
            raise Exception("There is no extremum.")

        self.iters += 1

        self.h *= 2.
        self.xk0 = self.xk1
        self.xk1 = self.xk2
        self.xk2 = self.xk1 + self.h

        return (self.xk0, self.xk2) if self.xk0 ⩽ self.xk2 else
        ↪  (self.xk2, self.xk0)

    def compute(self):
        for i in self:
            pass
        return (self.xk0, self.xk2) if self.xk0 ⩽ self.xk2 else
        ↪  (self.xk2, self.xk0)

```

**graphics.py**

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from decimal import Decimal as dcm

def rosen():
    xList = np.arange(-10, 10, 0.05)
    yList = np.arange(-10, 10, 0.05)
    X, Y = np.meshgrid(xList, yList)

    Z = 100 * (Y - X**2)**2 + (1 - X)**2

    return X, Y, Z

```

```
def quadratic():
    xList = np.arange(-10, 10, 0.05)
    yList = np.arange(-10, 10, 0.05)
    X, Y = np.meshgrid(xList, yList)

    Z = 100 * (Y - X)**2 + (1 - X)**2

    return X, Y, Z

def variant():
    xList = np.arange(-10, 10, 0.05)
    yList = np.arange(-10, 10, 0.05)
    X, Y = np.meshgrid(xList, yList)

    x1 = 1 + (X - 2)**2 + ((Y - 2) / 2)**2
    x2 = 1 + ((X - 2)/3)**2 + (Y - 3)**2
    Z  = (3.0/x1) + (2.0/x2)

    return X, Y, Z


def testing(num):
    switch = {
        "1": rosen(),
        "2": quadratic(),
        "3": variant(),
    }
    return switch.get(num, "Invalid input")

def main():
    x = []
    y = []

    with open("coords.txt") as file:
        for line in file:
            xC, yC = line.split()
            x.append(dcm(xC))
            y.append(dcm(yC))

    _levels = np.arange(0, 900, 30)
    figure, axes = plt.subplots(1, 1)

    num = input("Enter the test: \n1) Rosenbrock \n2) Quadratic \n3)
    ↪  Function for 8 variant\n")

    X, Y, Z = testing(num)

    plt.xlim(-5, 5)
    plt.ylim(-5, 5)
```

```
63
64      axes.xaxis.set_major_locator(ticker.MultipleLocator(1))
65      axes.xaxis.set_minor_locator(ticker.MultipleLocator(1))
66      axes.yaxis.set_major_locator(ticker.MultipleLocator(1))
67      axes.yaxis.set_minor_locator(ticker.MultipleLocator(1))
68
69      plt.xlabel("x1")
70      plt.ylabel("x2")
71
72      plt.plot(x, y, '-o', markersize=6, color='c')
73      plt.plot(x[-1], y[-1], 'o', markersize=9, color='r')
74
75      _contourf = axes.contourf(X, Y, Z, levels=_levels, extend='max')
76      cs = _contourf
77      cs.cmap.set_over('blue', alpha = 0.2)
78      cs.changed()
79
80      figure.colorbar(_contourf, shrink=1)
81
82      plt.grid()
83      axes.set_aspect(1)
84      plt.savefig('kek.eps')
85
86  if __name__ == "__main__":
87      main()
```