|  |  |  |
| --- | --- | --- |
| x | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Методы оптимизации» | | |
| **Методы спуска (0-го, 1-го и 2-го порядка и переменной метрики)** | | |
|  | Группа | пм-02 |
| Бригада | дОЛГИХ АЛЕКСАНДР |
|  | дРЯГИН кИРИЛЛ |
| Вариант | 12 |
|  |  |
| Преподаватель | Филиппова Елена Владимировна |
|  |  |
| Дата | 19.04.2023 |
|  | | |

Новосибирск, 2023

1. **Цель работы**

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

1. **Задание (метод Бройдена и метод сопряженных градиентов)**
   1. Реализовать два метода поиска экстремума функции (разного порядка). Включить в реализуемый алгоритм собственную процедуру, реализующую одномерный поиск по направлению. Выбранные методы должны иметь разный порядок.
   2. Исследовать алгоритмы на функциях:

квадратичная функция

функция Розенброка

целевая функция варианта

Осуществлять 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 объёме вычислений в зависимости от задаваемой точности и начального приближения.

* 1. 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.
  2. 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.

1. **Результаты проведенных исследований**
   1. *Сравнение сходимости методов на квадратичной функции с начальными приближениями и*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тест |  |  | Итерации | Вычисления |  |  |
| MSG(Fibonacci) | -3.00000000 4.00000000 | 0.001 | 5 | 268 | 0.99984656 0.99984577 | 2.3606372863644606E-08 |
| MSG(Fibonacci) | 4.00000000 -7.00000000 | 0.001 | 11 | 595 | 0.99997462 0.99997449 | 6.458403011265796E-10 |
| MSG(Parabols) | -3.00000000 4.00000000 | 0.001 | 4 | 139 | 0.99962133 0.99961959 | 1.4369231990747654E-07 |
| MSG(Parabols) | 4.00000000 -7.00000000 | 0.001 | 5 | 179 | 0.99999981 1.00000026 | 2.0278320049512766E-11 |
| Broyden(Fibonacci) | -3.00000000 4.00000000 | 0.001 | 6 | 357 | 1.00000185 1.00000190 | 3.6347552552614526E-12 |
| Broyden(Fibonacci) | 4.00000000 -7.00000000 | 0.001 | 4 | 239 | 1.00000000 1.00000001 | 1.2440695571055016E-14 |
| Broyden(Parabols) | -3.00000000 4.00000000 | 0.001 | 5 | 179 | 0.99999877 0.99999877 | 1.5127468502539295E-12 |
| Broyden(Parabols) | 4.00000000 -7.00000000 | 0.001 | 4 | 148 | 1.00000004 0.99999996 | 7.169032346692852E-13 |
| MSG(Fibonacci) | -3.00000000 4.00000000 | 0.0001 | 6 | 337 | 0.99999994 1.00000004 | 1.1373709450558025E-12 |
| MSG(Fibonacci) | 4.00000000 -7.00000000 | 0.0001 | 11 | 595 | 0.99997462 0.99997449 | 6.458403011265796E-10 |
| MSG(Parabols) | -3.00000000 4.00000000 | 0.0001 | 5 | 179 | 1.00000000 1.00000000 | 5.732848632327099E-17 |
| MSG(Parabols) | 4.00000000 -7.00000000 | 0.0001 | 6 | 211 | 1.00000003 1.00000003 | 1.0497075115070684E-15 |
| Broyden(Fibonacci) | -3.00000000 4.00000000 | 0.0001 | 6 | 357 | 1.00000185 1.00000190 | 3.6347552552614526E-12 |
| Broyden(Fibonacci) | 4.00000000 -7.00000000 | 0.0001 | 4 | 239 | 1.00000000 1.00000001 | 1.2440695571055016E-14 |
| Broyden(Parabols) | -3.00000000 4.00000000 | 0.0001 | 5 | 179 | 0.99999877 0.99999877 | 1.5127468502539295E-12 |
| Broyden(Parabols) | 4.00000000 -7.00000000 | 0.0001 | 4 | 148 | 1.00000004 0.99999996 | 7.169032346692852E-13 |
| MSG(Fibonacci) | -3.00000000 4.00000000 | 1,00E-05 | 7 | 386 | 0.99999999 0.99999999 | 2.9612964219773628E-15 |
| MSG(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-05 | 13 | 713 | 1.00000000 0.99999999 | 3.6433477449971223E-16 |
| MSG(Parabols) | -3.00000000 4.00000000 | 1,00E-05 | 5 | 179 | 1.00000000 1.00000000 | 5.732848632327099E-17 |
| MSG(Parabols) | 4.00000000 -7.00000000 | 1,00E-05 | 6 | 211 | 1.00000003 1.00000003 | 1.0497075115070684E-15 |
| Broyden(Fibonacci) | -3.00000000 4.00000000 | 1,00E-05 | 7 | 406 | 1.00000186 1.00000187 | 3.480627743977036E-12 |
| Broyden(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-05 | 4 | 239 | 1.00000000 1.00000001 | 1.2440695571055016E-14 |
| Broyden(Parabols) | -3.00000000 4.00000000 | 1,00E-05 | 5 | 179 | 0.99999877 0.99999877 | 1.5127468502539295E-12 |
| Broyden(Parabols) | 4.00000000 -7.00000000 | 1,00E-05 | 5 | 180 | 1.00000000 1.00000000 | 2.1163815149618205E-19 |
| MSG(Fibonacci) | -3.00000000 4.00000000 | 1,00E-06 | 8 | 435 | 0.99999999 0.99999999 | 7.27106752059553E-17 |
| MSG(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-06 | 13 | 713 | 1.00000000 0.99999999 | 3.6433477449971223E-16 |
| MSG(Parabols) | -3.00000000 4.00000000 | 1,00E-06 | 5 | 179 | 1.00000000 1.00000000 | 5.732848632327099E-17 |
| MSG(Parabols) | 4.00000000 -7.00000000 | 1,00E-06 | 6 | 211 | 1.00000003 1.00000003 | 1.0497075115070684E-15 |
| Broyden(Fibonacci) | -3.00000000 4.00000000 | 1,00E-06 | 8 | 475 | 1.00000000 1.00000000 | 8.350351526748936E-27 |
| Broyden(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-06 | 5 | 288 | 1.00000000 1.00000000 | 4.094521631979987E-17 |
| Broyden(Parabols) | -3.00000000 4.00000000 | 1,00E-06 | 6 | 219 | 1.00000000 1.00000000 | 3.994767021439722E-24 |
| Broyden(Parabols) | 4.00000000 -7.00000000 | 1,00E-06 | 5 | 180 | 1.00000000 1.00000000 | 2.1163815149618205E-19 |
| MSG(Fibonacci) | -3.00000000 4.00000000 | 1,00E-07 | 8 | 435 | 0.99999999 0.99999999 | 7.27106752059553E-17 |
| MSG(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-07 | 14 | 762 | 0.99999999 0.99999999 | 3.0075363731791694E-17 |
| MSG(Parabols) | -3.00000000 4.00000000 | 1,00E-07 | 6 | 211 | 1.00000000 1.00000000 | 1.0208484481682195E-21 |
| MSG(Parabols) | 4.00000000 -7.00000000 | 1,00E-07 | 6 | 211 | 1.00000003 1.00000003 | 1.0497075115070684E-15 |
| Broyden(Fibonacci) | -3.00000000 4.00000000 | 1,00E-07 | 8 | 475 | 1.00000000 1.00000000 | 8.350351526748936E-27 |
| Broyden(Fibonacci) | 4.00000000 -7.00000000 | 1,00E-07 | 7 | 347 | 1.00000000 1.00000000 | 9.406843113183816E-18 |
| Broyden(Parabols) | -3.00000000 4.00000000 | 1,00E-07 | 6 | 219 | 1.00000000 1.00000000 | 3.994767021439722E-24 |
| Broyden(Parabols) | 4.00000000 -7.00000000 | 1,00E-07 | 5 | 180 | 1.00000000 1.00000000 | 2.1163815149618205E-19 |

* 1. *Исследование сходимости каждого метода*

Будем считать для всех дальнейших тестов: Метод Бройдена: , МСГ: ,

Квадратичная функция (метод Бройдена, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | 4916.000000 | 1455.191523 -1364.242053 | 0.002622 |  |  |  |  | -1455.191523 1364.242053 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | 0.814806 0.423619 | 15.337012 | 2.188304 2.725210 | 0.671089 | 3.814806 | 3.576381 | 4900.662988 | 2.967449 | 77.804430 -77.982065 | 0.470841 0.497830 | 0.497830 0.531643 |
| 2 | 2.283352 2.252477 | 1.742321 | -8.926193 6.306067 | 0.002622 | 1.468546 | 1.828857 | 13.594691 | 0.414796 | 8.926193 -6.306067 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | 2.259952 2.269008 | 1.595680 | -1.296891 -1.302090 | 0.972753 | 0.023400 | 0.016531 | 0.146641 | 1.747948 | 0.732747 1.865175 | 0.500283 0.498779 | 0.498779 0.502157 |
| 4 | 0.998397 1.002396 | 0.001601 | 0.802310 -0.799079 | 0.002622 | 1.261555 | 1.266613 | 1.594079 | 3.141592 | -0.802310 0.799079 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | 1.000500 1.000301 | 0.000004 | -0.000498 -0.000299 | 1.000247 | 0.002103 | 0.002095 | 0.001597 | 1.570777 | 0.040849 -0.039849 | 0.498738 0.498746 | 0.498746 0.503758 |
| 6 | 1.000002 1.000002 | 0.000000 |  |  | 0.000498 | 0.000299 | 0.000004 | 2.896541 | -0.000006 0.000009 |  |  |

Квадратичная функция (метод Бройдена, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | 4916.000000 | 1455.191523 -1364.242053 | 0.002483 |  |  |  |  | -1455.191523 1364.242053 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | 0.613932 0.611939 | 0.149446 | 0.374810 0.399577 | 0.741423 | 3.613932 | 3.388061 | 4915.850554 | 2.967449 | -0.374700 -0.399680 | 0.469371 0.497739 | 0.497739 0.533112 |
| 2 | 0.891825 0.908195 | 0.038501 | 3.488182 -3.270995 | 0.002498 | 0.277893 | 0.296256 | 0.110945 | 0.033598 | -3.488182 3.270995 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | 0.900538 0.900024 | 0.009919 | 0.099137 0.099677 | 1.003124 | 0.008713 | 0.008171 | 0.028582 | 1.547770 | -0.096277 -0.102522 | 0.498610 0.498745 | 0.498745 0.503886 |
| 4 | 0.999985 1.000013 | 0.000000 | 0.005708 -0.005677 | 0.002496 | 0.099447 | 0.099989 | 0.009919 | 0.003001 | -0.005708 0.005677 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | 0.999999 0.999999 | 0.000000 |  |  | 0.000014 | 0.000014 | 0.000000 | 1.568094 | -0.000001 -0.000001 |  |  |

Функция Розенброка (метод Бройдена, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | 2516.000000 | 6048.139767 954.969437 | 0.000164 |  |  |  |  | -6048.139767 -954.969437 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | -2.008778 4.156509 | 10.524615 | 1.383067 -8.775570 | 0.041943 | 0.991222 | 0.156509 | 2505.475385 | 2.057696 | 92.903463 23.803182 | 0.024931 -0.155409 | -0.155409 0.975231 |
| 2 | -1.950768 3.788434 | 8.736135 | 19.362290 3.552714 | 0.000328 | 0.058010 | 0.368075 | 1.788481 | 2.847725 | -19.362290 -3.552714 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | -1.944422 3.789598 | 8.677403 | 0.365548 -1.402924 | 0.754939 | 0.006346 | 0.001164 | 0.058732 | 1.864847 | 1.065814 1.776357 | 0.064013 -0.244193 | -0.244193 0.936292 |
| 4 | -1.668455 2.730476 | 7.404394 | 40.856207 10.835777 | 0.000414 | 0.275966 | 1.059122 | 1.273009 | 2.922418 | -40.856207 -10.835777 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | -1.651527 2.734966 | 7.036107 | 0.468387 -1.529658 | 0.671089 | 0.016929 | 0.004490 | 0.368286 | 1.860052 | -0.532907 1.509903 | 0.086093 -0.279824 | -0.279824 0.914322 |
| 6 | -1.337197 1.708429 | 6.097178 | 47.606363 15.987212 | 0.000655 | 0.314329 | 1.026536 | 0.938929 | 2.895481 | -47.606363 -15.987212 | 1.000000 0.000000 | 0.000000 1.000000 |
| 7 | -1.305994 1.718908 | 5.335264 | 0.600202 -1.608609 | 0.344574 | 0.031203 | 0.010479 | 0.761914 | 1.910915 | 2.220446 2.664535 | 0.123459 -0.328138 | -0.328138 0.877160 |
| 8 | -1.099180 1.164622 | 4.596424 | 23.181457 8.704149 | 0.000875 | 0.206814 | 0.554286 | 0.738840 | 2.848981 | -23.181457 -8.704149 | 1.000000 0.000000 | 0.000000 1.000000 |
| 9 | -1.078902 1.172236 | 4.328569 | 0.768151 -1.660894 | 0.335544 | 0.020278 | 0.007614 | 0.267855 | 1.968104 | -0.710543 1.687539 | 0.176950 -0.380685 | -0.380685 0.823922 |
| 10 | -0.821153 0.614933 | 3.668956 | 23.225866 11.857182 | 0.001379 | 0.257749 | 0.557304 | 0.659612 | 2.830826 | -23.225866 -11.857182 | 1.000000 0.000000 | 0.000000 1.000000 |
| 11 | -0.789124 0.631284 | 3.208305 | 0.986967 -1.627624 | 0.183599 | 0.032029 | 0.016351 | 0.460651 | 2.026793 | -0.888178 1.687539 | 0.269925 -0.442790 | -0.442790 0.731448 |
| 12 | -0.607918 0.332454 | 2.723118 | 12.256862 7.460699 | 0.002622 | 0.181206 | 0.298830 | 0.485187 | 2.790627 | -12.256862 -7.460699 | 1.000000 0.000000 | 0.000000 1.000000 |
| 13 | -0.575787 0.352012 | 2.525056 | 1.399072 -1.673742 | 0.167772 | 0.032132 | 0.019558 | 0.198063 | 2.094364 | 1.465494 4.085621 | 0.416046 -0.491672 | -0.491672 0.586027 |
| 14 | -0.341061 0.071205 | 2.002006 | 8.837375 9.015011 | 0.003597 | 0.234725 | 0.280807 | 0.523050 | 2.815774 | -8.837375 -9.015011 | 1.000000 0.000000 | 0.000000 1.000000 |
| 15 | -0.309269 0.103636 | 1.720568 | 1.868174 -1.266257 | 0.088682 | 0.031792 | 0.032431 | 0.281438 | 2.140426 | -1.643130 1.598721 | 0.686877 -0.462586 | -0.462586 0.316608 |
| 16 | -0.143596 -0.008657 | 1.393527 | 3.952394 5.839773 | 0.005243 | 0.165673 | 0.112294 | 0.327040 | 2.869260 | -3.952394 -5.839773 | 1.000000 0.000000 | 0.000000 1.000000 |
| 17 | -0.122874 0.021960 | 1.265556 | 2.107354 -0.603449 | 0.083886 | 0.020722 | 0.030618 | 0.127972 | 2.226001 | -1.887379 1.376677 | 0.925476 -0.261957 | -0.261957 0.079203 |
| 18 | 0.053904 -0.028661 | 0.994741 | 1.199041 6.306067 | 0.005413 | 0.176778 | 0.050621 | 0.270815 | 3.039557 | -1.199041 -6.306067 | 1.000000 0.000000 | 0.000000 1.000000 |
| 19 | 0.060394 0.005476 | 0.883193 | 1.859513 0.201498 | 0.083886 | 0.006491 | 0.034137 | 0.111548 | 1.871584 | -1.920686 0.355271 | 0.988132 0.108022 | 0.108022 0.016827 |
| 20 | 0.216382 0.022379 | 0.673799 | -0.555112 4.884981 | 0.005243 | 0.155987 | 0.016903 | 0.209394 | 0.017514 | 0.555112 -4.884981 | 1.000000 0.000000 | 0.000000 1.000000 |
| 21 | 0.213471 0.047991 | 0.619213 | 1.316394 0.567872 | 0.117348 | 0.002910 | 0.025612 | 0.054585 | 1.580890 | -1.776357 0.488498 | 0.841297 0.364484 | 0.364484 0.162910 |
| 22 | 0.367948 0.114630 | 0.442572 | -1.787459 4.157785 | 0.003417 | 0.154477 | 0.066639 | 0.176642 | 0.186132 | 1.787459 -4.157785 | 1.000000 0.000000 | 0.000000 1.000000 |
| 23 | 0.361841 0.128836 | 0.407685 | 0.828002 0.610554 | 0.167772 | 0.006108 | 0.014207 | 0.034886 | 1.674808 | -0.971445 -0.416334 | 0.648195 0.476339 | 0.476339 0.355044 |
| 24 | 0.500756 0.231270 | 0.287216 | -2.892131 3.896883 | 0.002622 | 0.138916 | 0.102434 | 0.120469 | 0.293315 | 2.892131 -3.896883 | 1.000000 0.000000 | 0.000000 1.000000 |
| 25 | 0.493175 0.241486 | 0.257173 | 0.507039 0.508179 | 0.250988 | 0.007582 | 0.010216 | 0.030043 | 1.776606 | -0.671685 -0.344169 | 0.499320 0.498746 | 0.498746 0.503180 |
| 26 | 0.620435 0.369033 | 0.169372 | -3.180789 3.178013 | 0.002622 | 0.127261 | 0.127547 | 0.087802 | 0.331183 | 3.180789 -3.178013 | 1.000000 0.000000 | 0.000000 1.000000 |
| 27 | 0.612097 0.377364 | 0.151199 | 0.303878 0.373138 | 0.395668 | 0.008338 | 0.008331 | 0.018172 | 1.820046 | -1.440514 0.541234 | 0.394088 0.487428 | 0.487428 0.607887 |
| 28 | 0.732331 0.525003 | 0.084430 | -2.778333 2.259304 | 0.001600 | 0.120235 | 0.147638 | 0.066769 | 0.334873 | 2.778333 -2.259304 | 1.000000 0.000000 | 0.000000 1.000000 |
| 29 | 0.727885 0.528618 | 0.074190 | 0.172375 0.253589 | 0.671089 | 0.004446 | 0.003615 | 0.010240 | 1.836889 | -0.194289 -0.238698 | 0.316986 0.464135 | 0.464135 0.684603 |
| 30 | 0.843564 0.698799 | 0.040860 | -3.979456 2.557676 | 0.001311 | 0.115679 | 0.170181 | 0.033330 | 0.345679 | 3.979456 -2.557676 | 1.000000 0.000000 | 0.000000 1.000000 |
| 31 | 0.838348 0.702152 | 0.026177 | 0.084610 0.142345 | 0.891554 | 0.005216 | 0.003353 | 0.014683 | 1.878539 | -0.096451 -0.135308 | 0.261941 0.438593 | 0.438593 0.739365 |
| 32 | 0.913782 0.829060 | 0.010960 | -1.989901 1.186724 | 0.001311 | 0.075434 | 0.126908 | 0.015217 | 0.337288 | 1.989901 -1.186724 | 1.000000 0.000000 | 0.000000 1.000000 |
| 33 | 0.911174 0.830615 | 0.007904 | 0.041191 0.074477 | 1.342177 | 0.002608 | 0.001556 | 0.003055 | 1.867019 | -0.315026 0.075460 | 0.231544 0.420766 | 0.420766 0.769611 |
| 34 | 0.966460 0.930577 | 0.002328 | -1.272810 0.693065 | 0.001311 | 0.055286 | 0.099961 | 0.005577 | 0.326416 | 1.272810 -0.693065 | 1.000000 0.000000 | 0.000000 1.000000 |
| 35 | 0.964791 0.931485 | 0.001284 | 0.015016 0.028284 | 1.650604 | 0.001668 | 0.000908 | 0.001044 | 1.876485 | -0.326323 0.132403 | 0.211212 0.407147 | 0.407147 0.789844 |
| 36 | 0.989577 0.978171 | 0.000228 | -0.410392 0.218030 | 0.001311 | 0.024785 | 0.046686 | 0.001056 | 0.314905 | 0.410392 -0.218030 | 1.000000 0.000000 | 0.000000 1.000000 |
| 37 | 0.989039 0.978457 | 0.000127 | 0.004523 0.008661 | 1.833733 | 0.000538 | 0.000286 | 0.000101 | 1.873639 | -0.124317 0.051784 | 0.203740 0.401772 | 0.401772 0.797277 |
| 38 | 0.997333 0.994338 | 0.000018 | -0.128222 0.066776 | 0.001311 | 0.008294 | 0.015882 | 0.000109 | 0.309498 | 0.128222 -0.066776 | 1.000000 0.000000 | 0.000000 1.000000 |
| 39 | 0.997165 0.994426 | 0.000009 | 0.001155 0.002215 | 1.873059 | 0.000168 | 0.000088 | 0.000009 | 1.877563 | -0.041077 0.017705 | 0.200152 0.399112 | 0.399112 0.800850 |
| 40 | 0.999328 0.998575 | 0.000001 | -0.031496 0.016511 | 0.001311 | 0.002164 | 0.004149 | 0.000008 | 0.306035 | 0.031496 -0.016511 | 1.000000 0.000000 | 0.000000 1.000000 |
| 41 | 0.999287 0.998596 | 0.000001 | 0.000300 0.000574 | 1.729454 | 0.000041 | 0.000022 | 0.000001 | 1.873737 | -0.010031 0.004303 | 0.201565 0.400170 | 0.400170 0.799438 |
| 42 | 0.999806 0.999589 | 0.000000 | -0.008669 0.004528 | 0.001311 | 0.000519 | 0.000993 | 0.000000 | 0.304262 | 0.008669 -0.004528 | 1.000000 0.000000 | 0.000000 1.000000 |
| 43 | 0.999794 0.999595 | 0.000000 | 0.000084 0.000161 | 1.885631 | 0.000011 | 0.000006 | 0.000000 | 1.874904 | -0.002807 0.001198 | 0.200152 0.399114 | 0.399114 0.800847 |
| 44 | 0.999952 0.999898 | 0.000000 | -0.002376 0.001236 | 0.001311 | 0.000158 | 0.000303 | 0.000000 | 0.305864 | 0.002376 -0.001236 | 1.000000 0.000000 | 0.000000 1.000000 |
| 45 | 0.999949 0.999900 | 0.000000 |  |  | 0.000003 | 0.000002 | 0.000000 | 1.876608 | -0.000767 0.000333 |  |  |

Функция Розенброка (метод Бройдена, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | 2516.000000 | 6048.139767 954.969437 | 0.000166 |  |  |  |  | -6048.139767 -954.969437 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | -1.997046 4.158361 | 11.877969 | 1.911151 -12.086377 | 0.039573 | 1.002954 | 0.158361 | 2504.122031 | 2.057696 | 130.206956 33.217873 | 0.025102 -0.155929 | -0.155929 0.975060 |
| 2 | -1.921416 3.680067 | 8.548536 | 14.743762 2.486900 | 0.000323 | 0.075630 | 0.478294 | 3.329432 | 2.850697 | -14.743762 -2.486900 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | -1.916659 3.680870 | 8.512212 | 0.367573 -1.370120 | 0.823598 | 0.004757 | 0.000802 | 0.036325 | 1.884877 | -0.177636 1.421085 | 0.067460 -0.250224 | -0.250224 0.932859 |
| 4 | -1.613927 2.552441 | 7.106327 | 38.813397 10.658141 | 0.000442 | 0.302732 | 1.128429 | 1.405884 | 2.923622 | -38.813397 -10.658141 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | -1.596767 2.557153 | 6.748807 | 0.457522 -1.449586 | 0.548632 | 0.017159 | 0.004712 | 0.357520 | 1.866639 | -0.266454 1.509903 | 0.091026 -0.286951 | -0.286951 0.909414 |
| 6 | -1.345756 1.761864 | 5.744586 | 31.175063 9.769963 | 0.000613 | 0.251011 | 0.795289 | 1.004221 | 2.889129 | -31.175063 -9.769963 | 1.000000 0.000000 | 0.000000 1.000000 |
| 7 | -1.326653 1.767851 | 5.419464 | 0.561567 -1.522617 | 0.363073 | 0.019103 | 0.005987 | 0.325121 | 1.919391 | -0.355271 1.598721 | 0.120328 -0.324520 | -0.324520 0.880281 |
| 8 | -1.122763 1.215029 | 4.713766 | 24.780178 9.148238 | 0.000836 | 0.203890 | 0.552821 | 0.705698 | 2.851153 | -24.780178 -9.148238 | 1.000000 0.000000 | 0.000000 1.000000 |
| 9 | -1.102047 1.222677 | 4.425276 | 0.713362 -1.597462 | 0.290613 | 0.020716 | 0.007648 | 0.288490 | 1.963093 | -0.710543 1.598721 | 0.166958 -0.372004 | -0.372004 0.833878 |
| 10 | -0.894735 0.758434 | 3.767407 | 18.829382 8.437695 | 0.001215 | 0.207312 | 0.464243 | 0.657869 | 2.828027 | -18.829382 -8.437695 | 1.000000 0.000000 | 0.000000 1.000000 |
| 11 | -0.871862 0.768684 | 3.511162 | 0.909365 -1.620326 | 0.218345 | 0.022873 | 0.010250 | 0.256245 | 2.017173 | -0.710543 1.731948 | 0.240567 -0.426359 | -0.426359 0.760635 |
| 12 | -0.673307 0.414894 | 2.947783 | 13.589130 7.682743 | 0.001845 | 0.198556 | 0.353791 | 0.563379 | 2.804804 | -13.589130 -7.682743 | 1.000000 0.000000 | 0.000000 1.000000 |
| 13 | -0.648239 0.429066 | 2.724528 | 1.198234 -1.599417 | 0.159015 | 0.025068 | 0.014173 | 0.223256 | 2.074785 | -1.021405 1.731948 | 0.360847 -0.479035 | -0.479035 0.640971 |
| 14 | -0.457701 0.174734 | 2.245690 | 9.325873 6.972201 | 0.002860 | 0.190538 | 0.254332 | 0.478838 | 2.798458 | -9.325873 -6.972201 | 1.000000 0.000000 | 0.000000 1.000000 |
| 15 | -0.431028 0.194675 | 2.055745 | 1.638128 -1.488303 | 0.108728 | 0.026673 | 0.019941 | 0.189945 | 2.134929 | -1.376677 1.776357 | 0.549547 -0.496286 | -0.496286 0.453219 |
| 16 | -0.252917 0.032854 | 1.666600 | 5.639933 6.217249 | 0.004366 | 0.178111 | 0.161821 | 0.389145 | 2.828308 | -5.639933 -6.217249 | 1.000000 0.000000 | 0.000000 1.000000 |
| 17 | -0.228294 0.059997 | 1.514915 | 2.033824 -1.013964 | 0.083189 | 0.024622 | 0.027143 | 0.151685 | 2.178365 | -1.754152 1.576517 | 0.802627 -0.397011 | -0.397011 0.201423 |
| 18 | -0.059104 -0.024353 | 1.199241 | 2.775558 5.573320 | 0.005604 | 0.169191 | 0.084350 | 0.315674 | 2.936101 | -2.775558 -5.573320 | 1.000000 0.000000 | 0.000000 1.000000 |
| 19 | -0.043551 0.006878 | 1.091479 | 2.086018 -0.246676 | 0.076296 | 0.015553 | 0.031230 | 0.107762 | 2.423687 | -1.998401 0.999201 | 0.986692 -0.114303 | -0.114303 0.018267 |
| 20 | 0.115605 -0.011943 | 0.846201 | 0.599520 5.062617 | 0.005360 | 0.159155 | 0.018820 | 0.245278 | 3.102671 | -0.599520 -5.062617 | 1.000000 0.000000 | 0.000000 1.000000 |
| 21 | 0.118818 0.015191 | 0.776597 | 1.678711 0.387904 | 0.091090 | 0.003213 | 0.027134 | 0.069604 | 1.555867 | -1.820766 0.222045 | 0.948794 0.219867 | 0.219867 0.055944 |
| 22 | 0.271732 0.050526 | 0.584722 | -1.065814 4.662937 | 0.004166 | 0.152914 | 0.035334 | 0.191875 | 0.099922 | 1.065814 -4.662937 | 1.000000 0.000000 | 0.000000 1.000000 |
| 23 | 0.267292 0.069953 | 0.537084 | 1.136002 0.618588 | 0.127471 | 0.004440 | 0.019427 | 0.047638 | 1.611667 | -1.310063 -0.299760 | 0.771271 0.418962 | 0.418962 0.232589 |
| 24 | 0.412100 0.148805 | 0.389816 | -2.287059 4.202194 | 0.003092 | 0.144808 | 0.078852 | 0.147268 | 0.242667 | 2.287059 -4.202194 | 1.000000 0.000000 | 0.000000 1.000000 |
| 25 | 0.405027 0.161800 | 0.354498 | 0.708024 0.586537 | 0.191595 | 0.007072 | 0.012995 | 0.035319 | 1.722697 | -0.821565 -0.449640 | 0.593671 0.489914 | 0.489914 0.409308 |
| 26 | 0.540681 0.274177 | 0.243948 | -3.011480 3.627654 | 0.002347 | 0.135654 | 0.112378 | 0.110550 | 0.311772 | 3.011480 -3.627654 | 1.000000 0.000000 | 0.000000 1.000000 |
| 27 | 0.533614 0.282691 | 0.217938 | 0.431528 0.470618 | 0.291419 | 0.007067 | 0.008513 | 0.026010 | 1.794342 | -0.496825 -0.410783 | 0.457684 0.496951 | 0.496951 0.544619 |
| 28 | 0.659369 0.419838 | 0.138320 | -3.264056 2.980949 | 0.001840 | 0.125755 | 0.137147 | 0.079618 | 0.341525 | 3.264056 -2.980949 | 1.000000 0.000000 | 0.000000 1.000000 |
| 29 | 0.653363 0.425322 | 0.120401 | 0.252766 0.336103 | 0.444600 | 0.006006 | 0.005485 | 0.017919 | 1.834510 | -0.284495 -0.312250 | 0.362258 0.479442 | 0.479442 0.639565 |
| 30 | 0.765743 0.574754 | 0.068353 | -3.089196 2.320366 | 0.001500 | 0.112380 | 0.149432 | 0.052048 | 0.348925 | 3.089196 -2.320366 | 1.000000 0.000000 | 0.000000 1.000000 |
| 31 | 0.761109 0.578235 | 0.057180 | 0.142559 0.219275 | 0.659992 | 0.004635 | 0.003481 | 0.011174 | 1.853499 | -0.157513 -0.209555 | 0.298059 0.456258 | 0.456258 0.703435 |
| 32 | 0.855196 0.722955 | 0.028034 | -2.584738 1.679906 | 0.001276 | 0.094088 | 0.144719 | 0.029145 | 0.344625 | 2.584738 -1.679906 | 1.000000 0.000000 | 0.000000 1.000000 |
| 33 | 0.851897 0.725099 | 0.021974 | 0.075163 0.128903 | 0.990921 | 0.003299 | 0.002144 | 0.006060 | 1.863466 | -0.080838 -0.125594 | 0.254635 0.434565 | 0.434565 0.746639 |
| 34 | 0.926377 0.852832 | 0.008276 | -1.822500 1.067896 | 0.001133 | 0.074481 | 0.127732 | 0.013698 | 0.337731 | 1.822500 -1.067896 | 1.000000 0.000000 | 0.000000 1.000000 |
| 35 | 0.924312 0.854042 | 0.005738 | 0.034313 0.063429 | 1.428656 | 0.002065 | 0.001210 | 0.002537 | 1.867482 | -0.036603 -0.062190 | 0.227233 0.418000 | 0.418000 0.773899 |
| 36 | 0.973333 0.944659 | 0.001450 | -1.008785 0.543142 | 0.001040 | 0.049021 | 0.090618 | 0.004289 | 0.329011 | 1.008785 -0.543142 | 1.000000 0.000000 | 0.000000 1.000000 |
| 37 | 0.972284 0.945224 | 0.000769 | 0.011516 0.022572 | 2.279027 | 0.001049 | 0.000565 | 0.000680 | 1.877240 | -0.012024 -0.022313 | 0.207355 0.404394 | 0.404394 0.793685 |
| 38 | 0.998530 0.996666 | 0.000018 | -0.155457 0.079197 | 0.001000 | 0.026246 | 0.051442 | 0.000752 | 0.327725 | 0.155457 -0.079197 | 1.000000 0.000000 | 0.000000 1.000000 |
| 39 | 0.998374 0.996745 | 0.000003 | 0.000649 0.001300 | 2.504368 | 0.000156 | 0.000079 | 0.000015 | 1.885951 | -0.000656 -0.001296 | 0.200158 0.399117 | 0.399117 0.800843 |
| 40 | 0.999999 1.000001 | 0.000000 | 0.001063 -0.000530 | 0.000999 | 0.001624 | 0.003255 | 0.000003 | 0.323366 | -0.001063 0.000530 | 1.000000 0.000000 | 0.000000 1.000000 |
| 41 | 1.000000 1.000000 | 0.000000 |  |  | 0.000001 | 0.000001 | 0.000000 | 1.248247 | 0.000000 -0.000000 |  |  |

Целевая функция варианта (метод Бройдена, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | -0.240602 | 0.058287 -0.058287 | 50.358602 |  |  |  |  | -0.058287 0.058287 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | -0.064763 1.064763 | -0.806635 | 4.646469 -3.179163 | 0.671089 | 2.935237 | 2.935237 | 0.566034 | 2.999696 | -0.310862 -0.310862 | 72.511553 -57.564526 | -57.564526 47.337613 |
| 2 | 3.053430 -1.068738 | -0.964386 | -0.044409 0.388578 | 7.786810 | 3.118193 | 2.133501 | 0.157750 | 2.231595 | 0.044409 -0.388578 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | 2.707626 1.957046 | -3.774214 | 4.148922 15.528457 | 0.003369 | 0.345804 | 3.025783 | 2.809828 | 2.021274 | -0.888178 -0.133227 | 3.149194 10.147176 | 10.147176 48.908745 |
| 4 | 2.721606 2.009367 | -3.782562 | 0.754952 -0.177636 | 0.354324 | 0.013979 | 0.052321 | 0.008348 | 0.683855 | -0.754952 0.177636 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | 2.989104 1.946426 | -3.900711 | 0.053644 0.145465 | 0.167914 | 0.267498 | 0.062941 | 0.118149 | 0.867063 | 0.000000 -0.177636 | 0.496438 0.301991 | 0.301991 0.818893 |
| 6 | 2.998111 1.970852 | -3.902648 |  |  | 0.009008 | 0.024426 | 0.001937 | 0.640290 | 0.000000 0.000000 |  |  |

Целевая функция варианта (метод Бройдена, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |  |  |
| 0 | -3.000000 4.000000 | -0.240602 | 0.058287 -0.058287 | 94.482883 |  |  |  |  | -0.058287 0.058287 | 1.000000 0.000000 | 0.000000 1.000000 |
| 1 | 2.507096 -1.507096 | -0.739650 | 1.729013 -1.086371 | 0.113413 | 5.507096 | 5.507096 | 0.499048 | 2.999696 | -0.255351 -0.255351 | 65.458960 -58.687847 | -58.687847 54.433431 |
| 2 | 2.703189 -1.630305 | -0.749806 | 0.177636 0.277556 | 32.306054 | 0.196093 | 0.123209 | 0.010156 | 0.019731 | -0.177636 -0.277556 | 1.000000 0.000000 | 0.000000 1.000000 |
| 3 | 8.441897 7.336426 | -0.109853 | -0.467660 -0.731754 | 18.129343 | 5.738708 | 8.966731 | 0.639953 | 1.544182 | 0.015266 0.024980 | 9.332644 13.018062 | 13.018062 21.338074 |
| 4 | -0.036463 -5.929785 | -0.109611 | 0.026368 0.018041 | 305.028832 | 8.478360 | 13.266211 | 0.000242 | 2.854917 | -0.026368 -0.018041 | 1.000000 0.000000 | 0.000000 1.000000 |
| 5 | 8.006475 -0.426722 | -0.266083 | -0.590739 -0.287469 | 7.991233 | 8.042938 | 5.503063 | 0.156472 | 2.176996 | 0.061062 -0.083267 | 191.428897 133.286653 | 133.286653 94.291155 |
| 6 | 3.285743 -2.723955 | -0.509868 | -0.077716 0.177636 | 25.244741 | 4.720733 | 2.297233 | 0.243785 | 2.635454 | 0.077716 -0.177636 | 1.000000 0.000000 | 0.000000 1.000000 |
| 7 | 1.323832 1.760411 | -1.962897 | 3.722255 -14.475035 | -0.006637 | 1.961910 | 4.484367 | 1.453029 | 2.675396 | -1.021405 -0.466294 | -0.733347 9.589021 | 9.589021 -52.047265 |
| 8 | 1.299128 1.856482 | -1.971948 | 1.043610 0.266454 | 1.491968 | 0.024704 | 0.096070 | 0.009051 | 2.245115 | -1.043610 -0.266454 | 1.000000 0.000000 | 0.000000 1.000000 |
| 9 | 2.856160 2.254022 | -3.641052 | -0.743235 0.245427 | -0.365858 | 1.557032 | 0.397540 | 1.669104 | 0.710240 | -0.355271 1.554312 | 0.621437 0.620219 | 0.620219 -0.016136 |
| 10 | 3.128079 2.164231 | -3.766483 | -0.399680 -1.154632 | 0.176582 | 0.271919 | 0.089791 | 0.125432 | 2.154544 | 0.399680 1.154632 | 1.000000 0.000000 | 0.000000 1.000000 |
| 11 | 3.057502 1.960343 | -3.897159 | -0.185108 0.037536 | 0.307670 | 0.070576 | 0.203888 | 0.130675 | 2.509291 | 0.177636 -0.088818 | 0.982701 -0.118724 | -0.118724 0.185170 |
| 12 | 3.000550 1.971892 | -3.902647 | 0.000000 -0.044409 | 0.031510 | 0.056952 | 0.011549 | 0.005489 | 2.371392 | 0.000000 0.044409 | 1.000000 0.000000 | 0.000000 1.000000 |
| 13 | 3.000550 1.970492 | -3.902653 |  |  | 0.000000 | 0.001399 | 0.000006 | 2.152200 | 0.000000 0.000000 | aa | aa |

Квадратичная функция (МСГ, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | 12109.000000 | -2182.787284 2182.787284 |  |  |  |  |  | 2182.787284 -2182.787284 |
| 1 | -1.722209 -1.277791 | 27.161190 | 90.309560 -84.980489 | 0.002622 | 5.722209 | 5.722209 | 12081.838810 | 2.875341 | -94.146912 88.817842 |
| 2 | -1.485461 -1.500568 | 6.200340 | 2.117770 2.860453 | 0.002622 | 0.236748 | 0.222778 | 20.960850 | 1.748258 | -2.042810 -2.930989 |
| 3 | -1.258723 -1.194316 | 5.516668 | 17.230661 -12.878587 | 0.107064 | 0.226738 | 0.306253 | 0.683672 | 2.998558 | -17.230661 12.878587 |
| 4 | -1.213553 -1.228077 | 4.920911 | 2.013766 2.620775 | 0.002622 | 0.045170 | 0.033761 | 0.595757 | 1.740595 | -1.598721 -2.930989 |
| 5 | -0.875699 -0.788384 | 4.280643 | 160.162132 163.405455 | 0.167772 | 0.337854 | 0.439693 | 0.640268 | 3.017307 | -21.227464 17.408297 |
| 6 | 0.803734 0.925058 | 1.510478 | 24.624747 -24.247271 | 0.010486 | 1.679433 | 1.713442 | 2.770166 | 3.079147 | -24.624747 24.247271 |
| 7 | 0.868288 0.861494 | 0.021965 | -1.031905 1.295850 | 0.002622 | 0.064554 | 0.063565 | 1.488513 | 1.633136 | 1.094611 -1.357595 |
| 8 | 0.865540 0.864944 | 0.018115 | 0.135799 0.136249 | 0.002663 | 0.002748 | 0.003451 | 0.003850 | 1.461816 | -0.149533 -0.119002 |
| 9 | 1.000050 0.999899 | 0.000002 | -0.030171 0.030072 | 0.990504 | 0.134510 | 0.134955 | 0.018113 | 0.001998 | 0.030171 -0.030072 |
| 10 | 0.999971 0.999978 | 0.000000 | 0.001470 -0.001412 | 0.002622 | 0.000079 | 0.000079 | 0.000002 | 1.572525 | -0.001547 0.001489 |
| 11 | 0.999975 0.999974 | 0.000000 | 0.000025 0.000025 | 0.002629 | 0.000004 | 0.000004 | 0.000000 | 1.550616 | -0.000025 -0.000026 |

Квадратичная функция (МСГ, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | 12109.000000 | -2182.787284 2182.787284 |  |  |  |  |  | 2182.787284 -2182.787284 |
| 1 | -1.493766 -1.506234 | 6.234414 | 2.395348 2.489634 | 0.002517 | 5.493766 | 5.493766 | 12102.765586 | 2.875341 | -2.398082 -2.486900 |
| 2 | 0.707952 0.782147 | 0.635787 | 107.216494 80.587716 | 0.919164 | 2.201717 | 2.288382 | 5.598627 | 3.126450 | -15.409896 14.832580 |
| 3 | 1.005755 1.005986 | 0.000038 | 0.034802 -0.046302 | 0.002778 | 0.297803 | 0.223839 | 0.635748 | 0.190602 | -0.034802 0.046302 |
| 4 | 1.005843 1.005868 | 0.000034 | -0.005945 -0.005970 | 0.002548 | 0.000089 | 0.000118 | 0.000004 | 1.711765 | 0.006666 0.005010 |
| 5 | 1.000000 1.000000 | 0.000000 | 0.000089 -0.000091 | 0.982893 | 0.005843 | 0.005868 | 0.000034 | 3.139503 | -0.000090 0.000090 |

Функция Розенброка (МСГ, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | 52909.000000 | -36379.788071 4365.574569 |  |  |  |  |  | 36379.788071 -4365.574569 |
| 1 | -0.057567 -6.513092 | 4247.473058 | 137.069746 1278.672086 | 0.000112 | 4.057567 | 0.486908 | 48661.526942 | 2.209371 | -181.898940 -1273.292582 |
| 2 | 0.691995 0.479287 | 0.094886 | 0.734180 -0.085620 | 0.005468 | 0.749562 | 6.992379 | 4247.378172 | 3.043642 | -0.734135 0.086042 |
| 3 | 0.693919 0.479063 | 0.094291 | -0.070777 0.492661 | 0.002622 | 0.001925 | 0.000224 | 0.000595 | 0.721849 | 0.070777 -0.492661 |
| 4 | 0.693668 0.480809 | 0.093853 | 0.434640 0.603010 | 0.003544 | 0.000251 | 0.001746 | 0.000439 | 1.109247 | -0.510703 -0.073552 |
| 5 | 0.793544 0.619375 | 0.053311 | 6.963378 15.707277 | 0.229791 | 0.099876 | 0.138566 | 0.040542 | 0.340151 | 2.869233 -2.065709 |
| 6 | 0.814067 0.665669 | 0.035449 | 1.331574 -0.591888 | 0.002947 | 0.020523 | 0.046294 | 0.017862 | 0.490762 | -1.331574 0.591888 |
| 7 | 0.815923 0.664844 | 0.033963 | 0.102052 0.167126 | 0.001394 | 0.001856 | 0.000825 | 0.001486 | 1.103718 | -0.078410 -0.177636 |
| 8 | 0.895688 0.795471 | 0.015486 | 15.762358 30.800702 | 0.781610 | 0.079765 | 0.130628 | 0.018477 | 0.338875 | 2.217324 -1.356207 |
| 9 | 0.997574 0.994563 | 0.000041 | -0.231189 0.118019 | 0.006464 | 0.101886 | 0.199091 | 0.015445 | 0.371582 | 0.231189 -0.118019 |
| 10 | 0.997271 0.994717 | 0.000010 | 0.050559 -0.022510 | 0.001311 | 0.000303 | 0.000155 | 0.000030 | 1.885705 | -0.072483 0.033702 |
| 11 | 0.997344 0.994684 | 0.000007 | 0.001049 0.002134 | 0.001455 | 0.000074 | 0.000033 | 0.000003 | 1.202986 | -0.000956 -0.002176 |
| 12 | 1.000161 1.000413 | 0.000001 | 0.036373 -0.018347 | 2.684355 | 0.002817 | 0.005729 | 0.000006 | 0.329788 | -0.036373 0.018347 |
| 13 | 1.000208 1.000389 | 0.000000 | -0.007911 0.003731 | 0.001311 | 0.000048 | 0.000024 | 0.000001 | 1.252699 | 0.011452 -0.005518 |
| 14 | 1.000197 1.000395 | 0.000000 | -0.000079 -0.000158 | 0.001453 | 0.000011 | 0.000005 | 0.000000 | 1.915357 | 0.000075 0.000159 |

Функция Розенброка (МСГ, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | 52909.000000 | -36379.788071 4365.574569 |  |  |  |  |  | 36379.788071 -4365.574569 |
| 1 | -0.059188 -6.512897 | 4247.469635 | 137.069746 1278.672086 | 0.000112 | 4.059188 | 0.487103 | 48661.530365 | 2.209371 | -181.898940 -1273.292582 |
| 2 | 0.690067 0.476619 | 0.096077 | 0.736956 -0.084229 | 0.005466 | 0.749255 | 6.989517 | 4247.373558 | 3.043891 | -0.736911 0.084655 |
| 3 | 0.691728 0.476429 | 0.095455 | 0.047184 0.412170 | 0.002254 | 0.001661 | 0.000190 | 0.000621 | 0.718247 | -0.047184 -0.412170 |
| 4 | 0.692065 0.479373 | 0.094842 | 0.881845 1.219148 | 0.007141 | 0.000337 | 0.002943 | 0.000614 | 0.853677 | -0.732747 0.083267 |
| 5 | 0.793676 0.619849 | 0.052714 | 1.814790 8.372544 | 0.115225 | 0.101611 | 0.140476 | 0.042128 | 0.338795 | 2.785272 -2.012973 |
| 6 | 0.797230 0.636247 | 0.041161 | 0.620337 -0.133921 | 0.001959 | 0.003554 | 0.016398 | 0.011553 | 0.694305 | -0.620337 0.133921 |
| 7 | 0.798215 0.636034 | 0.040841 | 0.126553 0.204910 | 0.001589 | 0.000985 | 0.000213 | 0.000320 | 0.886185 | -0.047184 -0.222045 |
| 8 | 0.888022 0.781446 | 0.017633 | 13.789328 27.496961 | 0.709637 | 0.089807 | 0.145412 | 0.023208 | 0.344733 | 2.312039 -1.426290 |
| 9 | 1.018186 1.041002 | 0.002179 | 1.701157 -0.859252 | 0.009439 | 0.130164 | 0.259556 | 0.015454 | 0.384315 | -1.701157 0.859252 |
| 10 | 1.019846 1.040164 | 0.000394 | -0.007653 -0.015657 | 0.000976 | 0.001660 | 0.000838 | 0.001785 | 1.264196 | 0.007795 0.015585 |
| 11 | 1.001573 1.002780 | 0.000016 | -0.856675 -1.371378 | 2.387624 | 0.018273 | 0.037384 | 0.000378 | 2.820701 | 0.150370 -0.073613 |
| 12 | 1.000617 1.001249 | 0.000000 | 0.004924 -0.003078 | 0.001116 | 0.000956 | 0.001531 | 0.000016 | 2.915167 | -0.004924 0.003078 |
| 13 | 1.000622 1.001246 | 0.000000 | -0.000251 -0.000502 | 0.001008 | 0.000005 | 0.000003 | 0.000000 | 1.344430 | 0.000296 0.000473 |

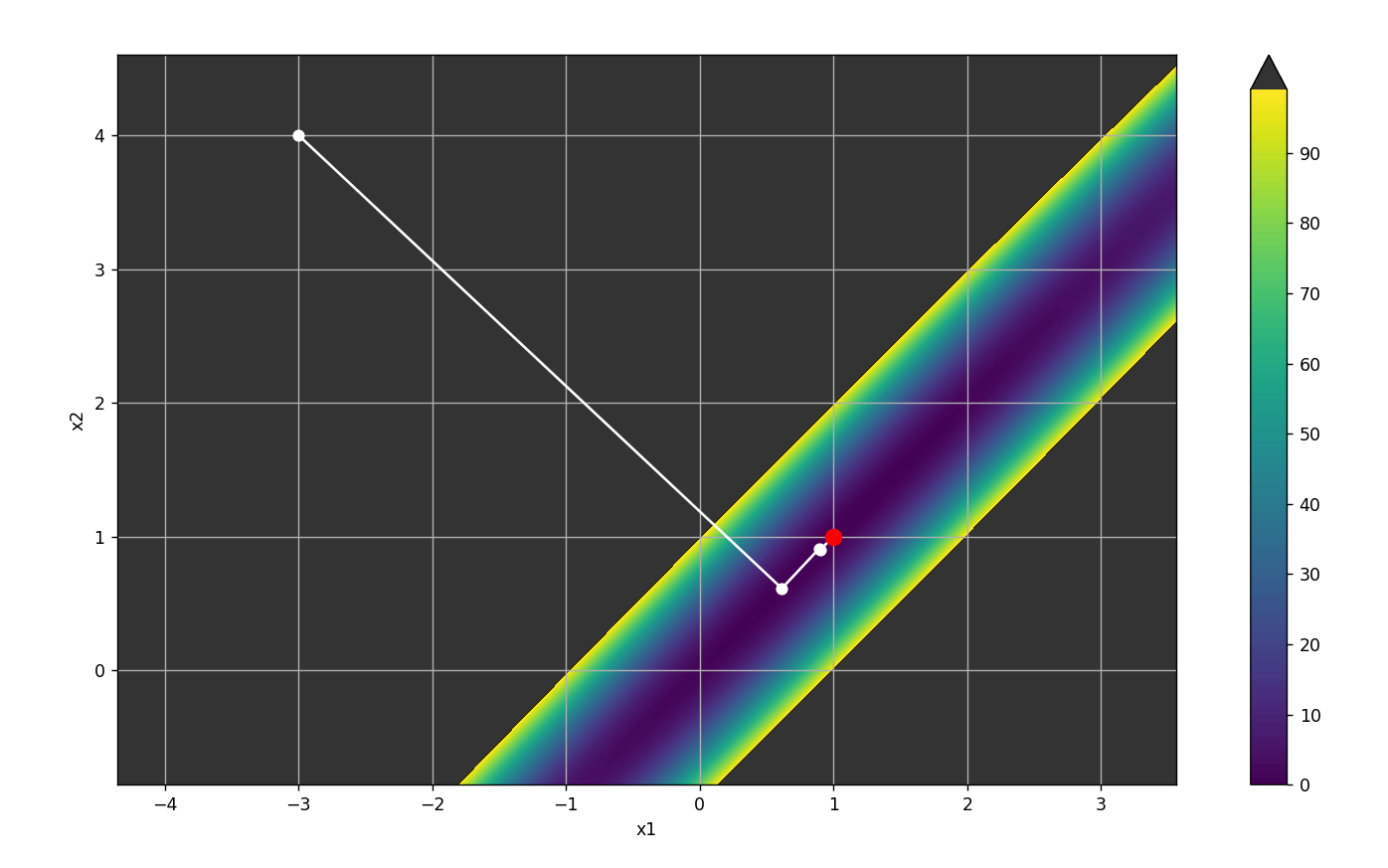
Целевая функция варианта (МСГ, метод Фибоначчи)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | -0.146230 | -0.022204 0.030531 |  |  |  |  |  | 0.022204 -0.030531 |
| 1 | -1.819976 1.002468 | -0.425729 | -0.334597 0.756362 | 262.108437 | 5.819976 | 8.002468 | 0.279499 | 3.031942 | -0.138778 -0.105471 |
| 2 | -2.044521 1.510054 | -0.432411 | 0.132919 0.075753 | 0.671089 | 0.224544 | 0.507586 | 0.006682 | 0.650832 | -0.144329 -0.049960 |
| 3 | 1.148160 3.329632 | -1.026244 | 0.355271 -0.643929 | 24.019810 | 3.192681 | 1.819579 | 0.593833 | 1.987427 | -0.355271 0.643929 |
| 4 | 2.101835 1.601097 | -2.746051 | 3.392931 -2.571993 | 2.684355 | 0.953674 | 1.728535 | 1.719807 | 2.305364 | -1.287859 -1.243450 |
| 5 | 2.259163 1.481835 | -2.769229 | 1.894862 0.982440 | 0.046370 | 0.157329 | 0.119262 | 0.023178 | 1.299616 | -1.199041 -1.509903 |
| 6 | 3.068188 1.901295 | -3.880956 | -0.177636 0.399680 | 0.426957 | 0.809025 | 0.419460 | 1.111727 | 0.102209 | 0.177636 -0.399680 |
| 7 | 3.033965 1.978297 | -3.900663 | -0.208768 0.025638 | 0.192659 | 0.034223 | 0.077002 | 0.019707 | 1.434256 | 0.177636 0.044409 |
| 8 | 2.998940 1.982598 | -3.902206 | -0.083757 -0.122941 | 0.167772 | 0.035025 | 0.004301 | 0.001543 | 2.441583 | 0.000000 0.133227 |
| 9 | 2.991914 1.972285 | -3.902542 | -0.000000 -0.044409 | 0.083886 | 0.007026 | 0.010313 | 0.000336 | 2.752983 | 0.000000 0.044409 |
| 10 | 2.991914 1.970423 | -3.902551 | 0.044409 -0.133227 | 0.041943 | 0.000000 | 0.001863 | 0.000010 | 2.153616 | -0.044409 0.044409 |
| 11 | 2.992379 1.969026 | -3.902556 | 0.004441 0.031086 | 0.010486 | 0.000466 | 0.001397 | 0.000005 | 1.831431 | 0.000000 -0.044409 |
| 12 | 2.992752 1.971633 | -3.902567 | 0.044409 -0.000000 | 0.083886 | 0.000373 | 0.002608 | 0.000011 | 0.846911 | -0.044409 0.000000 |
| 13 | 3.000203 1.971633 | -3.902650 | 0.044409 0.044409 | 0.167772 | 0.007451 | 0.000000 | 0.000082 | 0.582539 | 0.000000 -0.044409 |
| 14 | 2.999271 1.970702 | -3.902653 | 0.000000 0.000000 | -0.020972 | 0.000931 | 0.000931 | 0.000003 | 0.204001 | 0.000000 0.000000 |

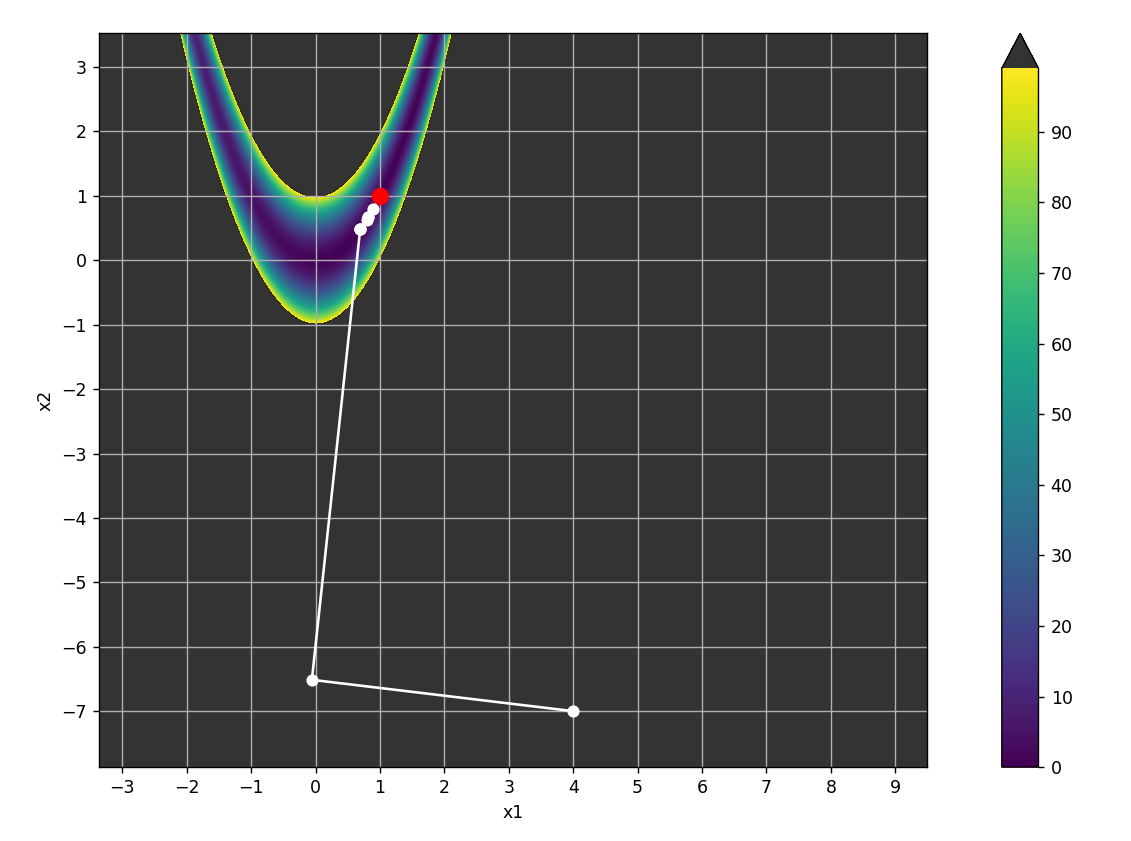
Целевая функция варианта (МСГ, метод Парабол)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Угол |  |
| 0 | 4.000000 -7.000000 | -0.146230 | -0.022204 0.030531 |  |  |  |  |  | 0.022204 -0.030531 |
| 1 | -1.819976 1.002468 | -0.425729 | -0.265043 0.634357 | 262.108436 | 5.819976 | 8.002468 | 0.279499 | 3.031942 | -0.127676 -0.094369 |
| 2 | -1.993545 1.417888 | -0.434687 | 0.121546 0.083468 | 0.654867 | 0.173568 | 0.415420 | 0.008958 | 0.671561 | -0.133227 -0.055511 |
| 3 | 0.574481 3.181412 | -0.910810 | 0.333067 -0.499600 | 21.128046 | 2.568026 | 1.763525 | 0.476123 | 1.921603 | -0.333067 0.499600 |
| 4 | 1.582464 1.669439 | -2.188360 | 2.775444 -1.742879 | 3.026366 | 1.007982 | 1.511974 | 1.277550 | 2.374941 | -1.110223 -0.754952 |
| 5 | 2.317182 1.208062 | -2.400240 | 1.847923 1.007401 | 0.264721 | 0.734718 | 0.461377 | 0.211880 | 1.372866 | -0.976996 -1.554312 |
| 6 | 7.392738 3.975018 | -0.355485 | -0.088818 -0.133227 | 2.746627 | 5.075556 | 2.766956 | 2.044755 | 0.018533 | 0.088818 0.133227 |
| 7 | 4.074672 -1.002081 | -0.675800 | -0.869390 -0.587991 | 37.358106 | 3.318066 | 4.977100 | 0.320315 | 2.652144 | 0.321965 -0.233147 |
| 8 | 3.314096 -1.516479 | -0.778894 | -0.302276 0.230480 | 0.874839 | 0.760576 | 0.514398 | 0.103094 | 2.305789 | 0.199840 -0.299760 |
| 9 | 1.078447 0.188169 | -0.786115 | 0.266454 0.410783 | 7.396062 | 2.235649 | 1.704648 | 0.007221 | 2.919293 | -0.266454 -0.410783 |
| 10 | 4.415224 5.332367 | -0.434521 | -0.074725 -0.093459 | 12.522924 | 3.336777 | 5.144198 | 0.351594 | 0.822629 | 0.122125 0.166533 |
| 11 | 1.979227 2.285650 | -2.684630 | -12.270381 -17.923243 | 32.599668 | 2.435996 | 3.046717 | 2.250109 | 3.124473 | -1.243450 1.021405 |
| 12 | 1.952126 2.246063 | -2.687515 | 1.332268 -0.888178 | 0.002209 | 0.027102 | 0.039587 | 0.002885 | 3.028240 | -1.332268 0.888178 |
| 13 | 2.714161 1.738040 | -3.630179 | 1.798220 0.503529 | 0.571983 | 0.762035 | 0.508023 | 0.942664 | 1.443302 | -0.754952 -1.199041 |
| 14 | 3.068573 1.837280 | -3.842979 | 0.097289 0.844370 | 0.197090 | 0.354412 | 0.099241 | 0.212799 | 0.296546 | 0.222045 -0.754952 |
| 15 | 3.083383 1.965816 | -3.891722 | -0.310862 -0.000000 | 0.152227 | 0.014810 | 0.128536 | 0.048743 | 0.916589 | 0.310862 0.000000 |
| 16 | 3.000000 1.965816 | -3.902587 | -0.006344 0.044409 | 0.268230 | 0.083383 | 0.000000 | 0.010866 | 2.574018 | 0.000000 -0.044409 |
| 17 | 2.999339 1.970444 | -3.902653 | -0.006217 0.087930 | 0.104195 | 0.000661 | 0.004627 | 0.000066 | 1.132621 | 0.000000 -0.044409 |
| 18 | 2.999337 1.970468 | -3.902653 | -0.000000 -0.000000 | 0.000279 | 0.000002 | 0.000025 | 0.000000 | 1.060134 | 0.000000 0.000000 |

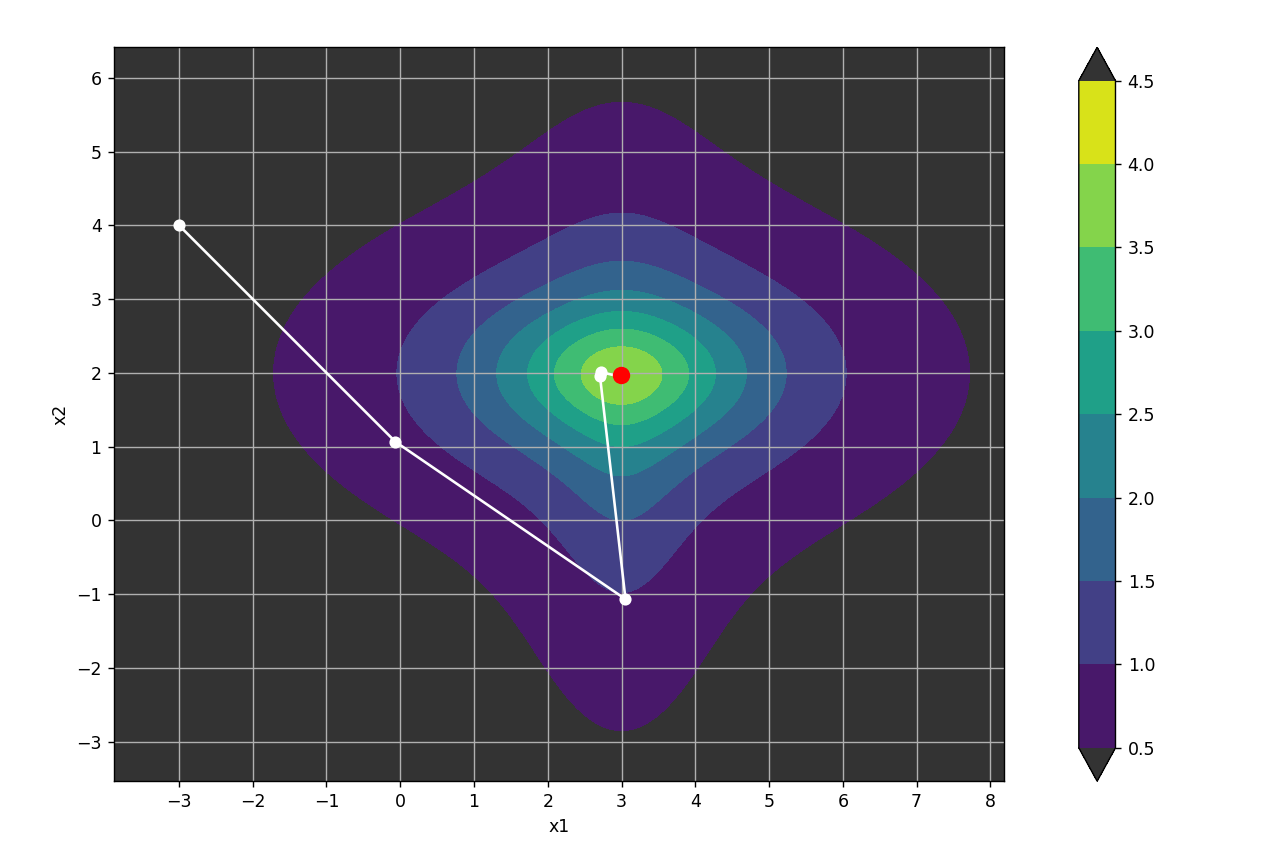
1. **Графика**



Квадратичная функция (Бройден, метод Парабол)



Функция Розенброка (МСГ, метод Фибоначчи)



Целевая функция варианта (метод Бройдена, метод Фибоначчи)

1. **Выводы**

По итогам исследования можно сделать следующие выводы:

* В среднем методБройдена делает меньшее количество итераций, чем метод сопряженных градиентов, однако производит больше вычислений функции (при равном количестве итераций).
* Выбор начального приближения может повлиять на количество итераций для обоих методов, но для метода сопряженных градиентов влияние значительнее.
* Точность вычислений не влияет на количество итераций метода Бройдена, когда для метода сопряженных градиентов наблюдается прямо пропорциональная зависимость.
* Метод парабол позволяет обоим методам сойтись за меньшее число итераций в сравнении с методом Фибоначчи, к тому же требует меньшего количества вычислений функций.

1. **Листинг**

**Functions.cs**

namespace OM2;

public interface IFunc

{

   public double Func(Vector x);

}

public class QuadraticFunc : IFunc

{

   public double Func(Vector x)

      => 100 \* Math.Pow(x[1] - x[0], 2) + Math.Pow(1 - x[0], 2);

}

public class RosenbrokeFunc : IFunc

{

   public double Func(Vector x)

      => 100 \* Math.Pow(x[1] - Math.Pow(x[0], 2), 2) + Math.Pow(1 - x[0], 2);

}

public class VariantFunc : IFunc

{

   public double Func(Vector x)

      => -(3 / (1 + Math.Pow((x[0] - 3) / 2, 2) + Math.Pow((x[1] - 2) / 1, 2)) + 1 / (1 + Math.Pow((x[0] - 3) / 1, 2) + Math.Pow((x[1] - 1) / 3, 2)));

}

**Matrix.cs**

namespace OM2;

public class Matrix

{

   private double[,] mat;

   public int Size { get; }

   public double this[int i, int j]

   {

      get => mat[i, j];

      set => mat[i, j] = value;

   }

   public Matrix(int size)

   {

      mat = new double[size, size];

      Size = size;

   }

   public void Clear()

       => Array.Clear(mat, 0, mat.Length);

   public void Copy(Matrix destination)

   {

      for (int i = 0; i < destination.Size; i++)

      {

         for (int j = 0; j < destination.Size; j++)

         {

            destination[i, j] = mat[i, j];

         }

      }

   }

   public static Matrix operator +(Matrix fstMatrix, Matrix sndMatrix)

   {

      Matrix resultMatrix = new(fstMatrix.Size);

      for (int i = 0; i < resultMatrix.Size; i++)

      {

         for (int j = 0; j < resultMatrix.Size; j++)

         {

            resultMatrix[i, j] = fstMatrix[i, j] + sndMatrix[i, j];

         }

      }

      return resultMatrix;

   }

   public static Matrix operator \*(double coef, Matrix Mat)

   {

      Matrix resultMatrix = new(Mat.Size);

      for (int i = 0; i < resultMatrix.Size; i++)

      {

         for (int j = 0; j < resultMatrix.Size; j++)

         {

            resultMatrix[i, j] = coef \* Mat[i, j];

         }

      }

      return resultMatrix;

   }

}

**MethodsND.cs**

using System.Runtime.CompilerServices;

using System.Security.Cryptography;

namespace OM2;

public abstract class MethodND

{

   public delegate double LinearSearch((double, double) interval, Vector x, Func<Vector, double> f, Vector direction, ref int funcCalc);

   protected LinearSearch LinearMethod;

   protected double eps;

   protected int maxIter;

   protected Func<Vector, double> func;

   protected IFunc Func;

   protected Vector x0;

   protected int funcCalc;

   protected string path = "results ";

   public MethodND(double eps, int maxIter, IFunc func, Vector x0, LinearSearch method)

   {

      this.eps = eps;

      this.maxIter = maxIter;

      this.func = func.Func;

      this.x0 = x0;

      funcCalc = 0;

      this.LinearMethod = method;

      Func = func;

   }

   public abstract Vector Compute();

   public Vector Gradient(Vector x, Func<Vector, double> func)

   {

      Vector grad = new(2);

      double fx = func(x);

      funcCalc++;

      double h = 1e-14;

      for (int i = 0; i < x.Length; i++)

      {

         grad[i] = (func(x + new Vector(h \* ((i + 1) % 2), h \* (i % 2))) - fx) / h;

         funcCalc++;

      }

      return grad;

   }

   public static void Issledovanie(List<Vector> coords, List<double> funcs, int iters, int funcCalc, double eps, string name)

   {

      StreamWriter sw = new("Issledovanie.csv", true);

      sw.WriteLine($"{name};{coords[0].ToString()}; {eps}; {iters}; {funcCalc}; {coords[^1].ToString()}; {funcs[^1]}");

      sw.Close();

   }

}

public class MSG : MethodND

{

   public MSG(double eps, int maxIter, IFunc func, Vector x0, LinearSearch method) : base(eps, maxIter, func, x0, method)

   {

   }

   public override Vector Compute()

   {

      List<Vector> coords = new(); // Координаты для графики.

      List<double> funcs = new(); // Значения функции.

      List<Vector> dirs = new(); // Значения направлений поиска.

      List<double> lambdas = new();

      List<double> corner = new(); // Угол между векторами (xi, yi) и (s1, s2)

      List<Vector> gradfs = new();

      Vector S = new(2), gradient = new(2), gradient0;

      int iter;

      double omegaK = 0, lambdaK;

      for (iter = 0; iter < maxIter;)

      {

         coords.Add(1 \* x0);

         funcs.Add(func(x0));

         gradient0 = Gradient(x0, func);

         if (iter % 3 == 0) S = -1 \* gradient0;

         else S = -1 \* gradient + omegaK \* S;

         gradfs.Add(gradient0);

         dirs.Add(S);

         if (S.Norm() < eps) break;

         omegaK = 1 / Math.Pow(S.Norm(), 2);

         var interval = Methods1D.FindInterval(x0, func, S, ref funcCalc);

         lambdaK = LinearMethod(interval, x0, func, S, ref funcCalc);

         lambdas.Add(lambdaK);

         var x = x0 + lambdaK \* S;

         gradient = Gradient(x, func);

         omegaK \*= Math.Pow(gradient.Norm(), 2);

         Vector.Copy(x, x0);

         iter++;

      }

      for (int i = 0; i < iter; i++)

      {

         double crn =

            Math.Acos

            (

             (coords[i][0] \* dirs[i][0] + coords[i][1] \* dirs[i][1])

             / (coords[i].Norm() \* dirs[i].Norm())

            );

         corner.Add(crn);

      }

      //Output(coords, funcs, dirs, corner, iter, funcCalc, lambdas, gradfs);

      Issledovanie(coords, funcs, iter, funcCalc, eps, Name());

      return x0;

   }

   public string Name()

   {

      string name = "MSG";

      if (LinearMethod == Methods1D.Fibonacci)

         name += "(Fibonacci)";

      else

         name += "(Parabols)";

      name += Func.GetType().ToString();

      return name;

   }

   private void Output(List<Vector> coords, List<double> funcs, List<Vector> dirs, List<double> corners, int iters, int funcCalc, List<double> lambdas, List<Vector> gradfs)

   {

      StreamWriter sw = new(path + Name() + ".csv");

      sw.WriteLine("i; Xi; f(Xi); S; Lambda; |Xi - Xi-1|; |Yi - Yi-1|; |fi - fi-1|; Angel above X and S; Gradient");

      for (int i = 0; i <= iters; i++)

      {

         if (i > 0)

            sw.WriteLine($"{i}; {coords[i].ToString()}; {funcs[i]:F6}; {dirs[i].ToString()}; {lambdas[i - 1]:F6}; {Math.Abs(coords[i][0] - coords[i - 1][0]):F6}; " +

               $"{Math.Abs(coords[i][1] - coords[i - 1][1]):F6}; {Math.Abs(funcs[i] - funcs[i - 1]):F6}; {corners[i - 1]:F6}; {gradfs[i].ToString()}");

         if (i == 0)

            sw.WriteLine($"{i}; {coords[i].ToString()}; {funcs[i]:F6}; {dirs[i].ToString()}; ; ; ; ; ; {gradfs[i].ToString()}");

      }

      sw.Close();

   }

}

public class Broyden : MethodND

{

   public Broyden(double eps, int maxIter, IFunc func, Vector x0, LinearSearch method) : base(eps, maxIter, func, x0, method)

   {

   }

   public override Vector Compute()

   {

      List<Vector> coords = new(); // Координаты для графики.

      List<double> funcs = new(); // Значения функции.

      List<Vector> dirs = new(); // Значения направлений поиска.

      List<double> lambdas = new();

      List<double> corner = new(); // Угол между векторами (xi, yi) и (s1, s2)

      List<Vector> gradfs = new();

      List<Matrix> matrices = new();

      int iter;

      double omegaK, lambdaK;

      Matrix H = new(2);

      Matrix deltaH = new(2);

      H[0, 0] = 1;

      H[1, 1] = 1;

      for (iter = 0; iter < maxIter; iter++)

      {

         coords.Add(1 \* x0);

         if (iter % 2 == 0 && iter != 0)

         {

            H.Clear();

            H[0, 0] = 1;

            H[1, 1] = 1;

         }

         var gradient0 = Gradient(x0, func);

         funcs.Add(func(x0));

         gradfs.Add(gradient0);

         matrices.Add(H);

         if (gradient0.Norm() < eps)

            break;

         var S = -1 \* H \* gradient0;

         var interval = Methods1D.FindInterval(x0, func, S, ref funcCalc);

         lambdaK = LinearMethod(interval, x0, func, S, ref funcCalc);

         var x1 = x0 + lambdaK \* S;

         dirs.Add(S);

         lambdas.Add(lambdaK);

         var gradient1 = Gradient(x1, func);

         var deltaGrad = gradient1 - gradient0;

         var deltaX = x1 - x0;

         var denominatorAsVec = deltaX - H \* deltaGrad;

         var denominator = denominatorAsVec \* deltaGrad;

         for (int i = 0; i < 2; i++)

            for (int j = 0; j < 2; j++)

               deltaH[i, j] = denominatorAsVec[i] \* denominatorAsVec[j];

         deltaH = 1 / denominator \* deltaH;

         H += deltaH;

         Vector.Copy(x1, x0);

      }

      for (int i = 0; i < iter; i++)

      {

         double crn =

            Math.Acos

            (

             (coords[i][0] \* dirs[i][0] + coords[i][1] \* dirs[i][1])

             / (coords[i].Norm() \* dirs[i].Norm())

            );

         corner.Add(crn);

      }

      //Output(coords, funcs, dirs, corner, iter, funcCalc, lambdas, gradfs, matrices);

      Issledovanie(coords, funcs, iter, funcCalc, eps, Name());

      return x0;

   }

   public string Name()

   {

      string name = "Broyden";

      if (LinearMethod == Methods1D.Fibonacci)

         name += "(Fibonacci)";

      else

         name += "(Parabols)";

      name += Func.GetType().ToString();

      return name;

   }

   private void Output(List<Vector> coords, List<double> funcs, List<Vector> dirs, List<double> corners, int iters, int funcCalc, List<double> lambdas, List<Vector> gradfs, List<Matrix> matrices)

   {

      StreamWriter sw = new(path + Name() + ".csv");

      sw.WriteLine("i; Xi; f(Xi); S; Lambda; |Xi - Xi-1|; |Yi - Yi-1|; |fi - fi-1|; Angel above X and S; Gradient; Eta 1 stroka; Eta 2 stroka");

      for (int i = 0; i <= iters; i++)

      {

         if (i > 0 && i < iters)

            sw.WriteLine($"{i}; {coords[i].ToString()}; {funcs[i]:F6}; {dirs[i].ToString()}; {lambdas[i]:F6}; {Math.Abs(coords[i][0] - coords[i - 1][0]):F6}; " +

               $"{Math.Abs(coords[i][1] - coords[i - 1][1]):F6}; {Math.Abs(funcs[i] - funcs[i - 1]):F6}; {corners[i - 1]:F6}; {gradfs[i].ToString()};" +

               $" {matrices[i][0, 0]:F6} {matrices[i][0, 1]:F6}; {matrices[i][1, 0]:F6} {matrices[i][1, 1]:F6}");

         if (i == 0)

            sw.WriteLine($"{i}; {coords[i].ToString()}; {funcs[i]:F6}; {dirs[i].ToString()}; {lambdas[i]:F6}; ; ; ; ; {gradfs[i].ToString()};" +

               $" {matrices[i][0,0]:F6} {matrices[i][0, 1]:F6}; {matrices[i][1, 0]:F6} {matrices[i][1, 1]:F6}");

         if (i == iters && iters < 100)

            sw.WriteLine($"{i}; {coords[i].ToString()}; {funcs[i]:F6}; ; ; {Math.Abs(coords[i][0] - coords[i - 1][0]):F6}; " +

            $"{Math.Abs(coords[i][1] - coords[i - 1][1]):F6}; {Math.Abs(funcs[i] - funcs[i - 1]):F6}; {corners[i - 1]:F6}; {gradfs[i].ToString()}");

      }

      sw.Close();

   }

}

**Methods1D.cs**

namespace OM2;

public static class Methods1D

{

   public static (double, double) FindInterval(Vector x, Func<Vector, double> f, Vector direction, ref int funcCalc)

   {

      double delta = 1e-8;

      double a = 0, h;

      double f1 = f(x), f2 = f(x + delta \* direction);

      funcCalc += 2;

      if (f1 == f2) return (a, a + delta);

      if (f1 > f2) h = delta;

      else h = -delta;

      f2 = f(x + (a + h) \* direction);

      funcCalc++;

      while (f1 > f2)

      {

         a += h;

         h \*= 2;

         f1 = f2;

         f2 = f(x + (a + h) \* direction);

         funcCalc++;

      }

      return (Math.Min(a, a + h), Math.Max(a, a + h));

   }

   public static double Fibonacci((double, double) interval, Vector x, Func<Vector, double> f, Vector direction, ref int funcCalc)

   {

      double eps = 1e-7;

      double a, b;

      (a, b) = interval;

      int n = 1;

      while ((b - a) / eps > Fibonacci(n + 2))

         n++;

      double x1, x2, f1, f2;

      x1 = a + Fibonacci(n) / Fibonacci(n + 2) \* (b - a);

      x2 = a + Fibonacci(n + 1) / Fibonacci(n + 2) \* (b - a);

      f1 = f(x + x1 \* direction);

      f2 = f(x + x2 \* direction);

      funcCalc += 2;

      for (int iter = 1; iter < n; iter++)

      {

         if (f1 == f2) break;

         if (f1 > f2)

         {

            a = x1;

            x1 = x2;

            x2 = a + Fibonacci(n - iter + 2) / Fibonacci(n - iter + 3) \* (b - a);

            f1 = f2;

            f2 = f(x + x2 \* direction);

         }

         else

         {

            b = x2;

            x2 = x1;

            x1 = a + Fibonacci(n - iter + 1) / Fibonacci(n - iter + 3) \* (b - a);

            f2 = f1;

            f1 = f(x + x1 \* direction);

         }

         funcCalc++;

      }

      return (a + b) / 2;

   }

   public static double Fibonacci(int i)

   {

      return (Math.Pow((1 + Math.Sqrt(5)) / 2, i) - Math.Pow((1 - Math.Sqrt(5)) / 2, i)) / Math.Sqrt(5);

   }

   public static double QuadraticSearch((double, double) interval, Vector x, Func<Vector, double> f, Vector direction, ref int funcCalc)

   {

      double eps = 1e-7;

      double a, b;

      (a, b) = interval;

      double lmbd1 = a, lmbd2 = (a + b) / 2, lmbd3 = b;

      double lmbd = 0, lmbdPrev;

      double f1 = f(x + lmbd1 \* direction), f2 = f(x + lmbd2 \* direction), f3 = f(x + lmbd3 \* direction);

      funcCalc += 3;

      while (true)

      {

         var c2 = (f2 - f1) / (lmbd2 - lmbd1);

         var c3 = ((f3 - f1) / (lmbd3 - lmbd1) - (f2 - f1) / (lmbd2 - lmbd1)) / (lmbd3 - lmbd2);

         lmbdPrev = lmbd;

         lmbd = (lmbd1 + lmbd2 - c2 / c3) / 2;

         var fx = f(x + lmbd \* direction);

         funcCalc++;

         if (Math.Abs(lmbd - lmbdPrev) < eps) break;

         if (lmbd > lmbd2)

         {

            if (fx > f2)

            {

               lmbd3 = lmbd;

               f3 = fx;

            }

            else

            {

               lmbd1 = lmbd2;

               f1 = f2;

               lmbd2 = lmbd;

               f2 = fx;

            }

         }

         else

         {

            if (fx < f2)

            {

               lmbd3 = lmbd2;

               f3 = f2;

               lmbd2 = lmbd;

               f2 = fx;

            }

            else

            {

               lmbd1 = lmbd;

               f1 = fx;

            }

         }

      }

      return lmbd;

   }

}

**Program.cs**

using OM2;

using System.Xml.Linq;

Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo("en-US");

//List<IFunc> funcs = new List<IFunc>();

//funcs.Add(new QuadraticFunc());

//funcs.Add(new RosenbrokeFunc());

//funcs.Add(new VariantFunc());

//foreach (var f in funcs)

//{

//   var test = new Broyden(1e-3, 100, f, new Vector(-3, 4), Methods1D.Fibonacci);

//   Console.WriteLine(test.Compute().ToString());

//}

StreamWriter sw = new("Issledovanie.csv");

sw.WriteLine($"Method name; X0; eps; Iters; Func calculation; Xi; f(Xi)");

sw.Close();

double[] eps = new double[] { 1e-3, 1e-4, 1e-5, 1e-6, 1e-7 };

MethodND test;

foreach (var acc in eps)

{

   test = new MSG(acc, 100, new QuadraticFunc(), new Vector(-3, 4), Methods1D.Fibonacci);

   Console.WriteLine(test.Compute().ToString());

   test = new MSG(acc, 100, new QuadraticFunc(), new Vector(4, -7), Methods1D.Fibonacci);

   Console.WriteLine(test.Compute().ToString());

   test = new MSG(acc, 100, new QuadraticFunc(), new Vector(-3, 4), Methods1D.QuadraticSearch);

   Console.WriteLine(test.Compute().ToString());

   test = new MSG(acc, 100, new QuadraticFunc(), new Vector(4, -7), Methods1D.QuadraticSearch);

   Console.WriteLine(test.Compute().ToString());

   test = new Broyden(acc, 100, new QuadraticFunc(), new Vector(-3, 4), Methods1D.Fibonacci);

   Console.WriteLine(test.Compute().ToString());

   test = new Broyden(acc, 100, new QuadraticFunc(), new Vector(4, -7), Methods1D.Fibonacci);

   Console.WriteLine(test.Compute().ToString());

   test = new Broyden(acc, 100, new QuadraticFunc(), new Vector(-3, 4), Methods1D.QuadraticSearch);

   Console.WriteLine(test.Compute().ToString());

   test = new Broyden(acc, 100, new QuadraticFunc(), new Vector(4, -7), Methods1D.QuadraticSearch);

   Console.WriteLine(test.Compute().ToString());

}