|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Фундаментальные науки

КАФЕДРА Прикладная математика

**ОТЧЕТЫ ПО ЛАБОРАТОРНЫМ РАБОТАМ №5-8**

Студент Швецов Григорий Алексеевич

*фамилия, имя, отчество*

Группа ФН2-52Б

Название предприятия: Научно-учебный комплекс «Фундаментальные науки»   
 МГТУ им. Н.Э. Баумана

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Швецов Г.А.

*подпись, дата фамилия и.о.*

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чередниченко А.В.

*подпись, дата фамилия и.о.*

*2022 г.*

Содержание

[1. Задание 3](#_Toc122869779)

[2. Квазиньютоновские методы 3](#_Toc122869780)

[2.1. Результаты 3](#_Toc122869781)

[2.2. Вывод 8](#_Toc122869782)

[3. Методы прямого поиска 8](#_Toc122869783)

[3.1. Результаты 9](#_Toc122869787)

[3.2. Вывод 15](#_Toc122869792)

[4. Симплекс – методы 15](#_Toc122869793)

[4.1. Результаты 15](#_Toc122869794)

[4.2. Вывод 17](#_Toc122869795)

[5. Методы последовательной безусловной минимизации 17](#_Toc122869796)

[5.1. Результаты 17](#_Toc122869797)

[5.2. Вывод 17](#_Toc122869798)

[6. Заключение 17](#_Toc122869799)

# Задание

Квадратичная функция:

Функция Розенброка

Во всех лабораторных работах необходимо найти с заданной точностью точку минимума и минимальное значение целевой функции. Начинать всегда с квадратичной функции (аналитически для нее найти точное решение, с котором сравнивать полученное численное). Далее исследовать функцию Розенброка различными параметрами . При исследовании для каждой функции брать два параметра точности поиска и . Варианты заданий даны в таблице ниже. Также для каждой функции и каждого параметра точности поиска взять две различные (существенно различные) начальные точки. Начальные точки выбрать самостоятельно.

В методах, в которых необходимо проводить одномерную минимизацию (например, в наискорейшем спуске), использовать свой метод золотого сечения, реализованный в лабораторной работе №1.

# Квазиньютоновские методы

Квазиньютоновские методы:

* метод ДФП;
* метод БФШ (БПГШ);
* метод Пауэлла.
  1. Результаты

*Таблица 1. Квадратичная функция.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы** | **ДФП** | **БФШ (БПГШ)** | **Пауэлл** |
| *X0 = [2, -2], eps = 0.01* | | | |
| Iter | 2 | 2 | 2 |
| Value | 105 | 105 | 105 |
| *X0 = [2, -2], eps = 0.000001* | | | |
| Iter | 2 | 2 | 2 |
| Value | 105 | 105 | 105 |

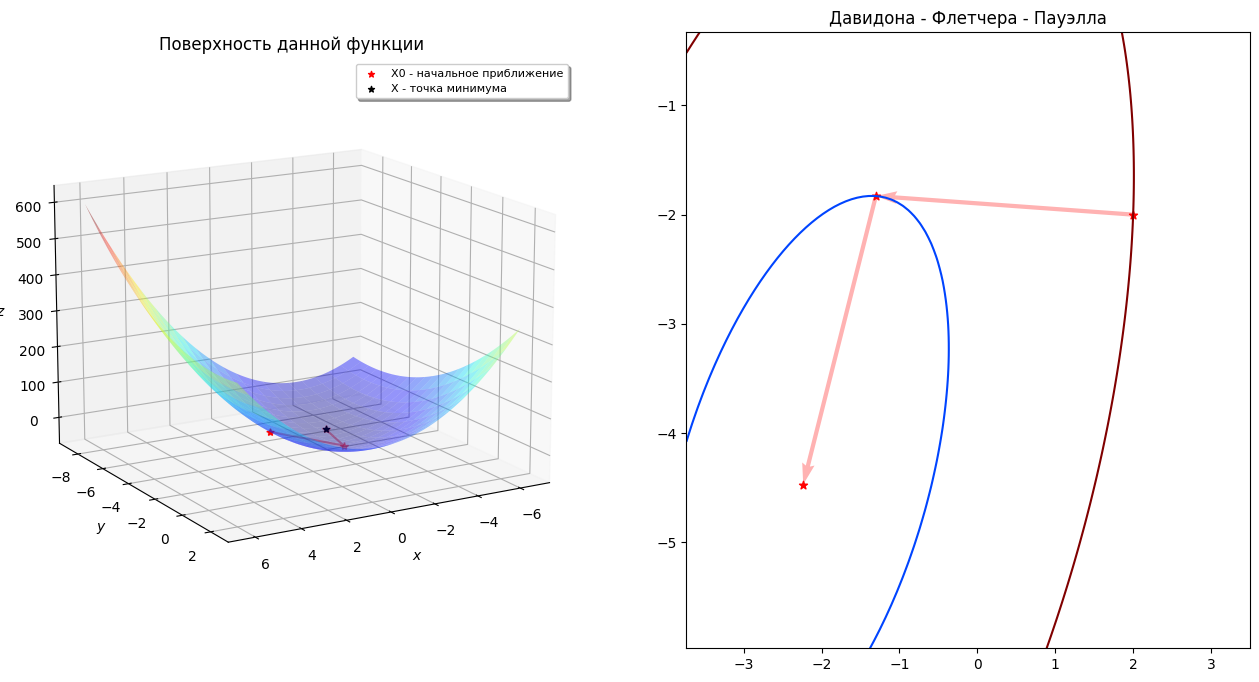


Рисунок 1. Квадратичная функция. Начальная точка - (2, -2), eps = 0.000001.

*Таблица 2. Функция Розенброка.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод** | **ДФП** | **БФШ (БПГШ)** | **Пауэлл** |
| *X0 = [-2, 2], Eps = 0.01, alpha = 30* | | | |
| Iter | 13 | 13 | 13 |
| Value | 718 | 723 | 718 |
| *X0 = [-2, 2], Eps = 0.000001, alpha = 30* | | | |
| Iter | 16 | 16 | 16 |
| Value | 870 | 875 | 870 |
| *X0 = [-2, 2], Eps = 0.01, alpha = 133* | | | |
| Iter | 39 | 27 | 35 |
| Value | 2197 | 1570 | 1974 |
| *X0 = [-2, 2], Eps = 0.000001, alpha = 133* | | | |
| Iter | 41 | 31 | 37 |
| Value | 2306 | 1788 | 2083 |
| *X0 = [20, -20], Eps = 0.01, alpha = 30* | | | |
| Iter | 11 | 31 | 17 |
| Value | 626 | 1779 | 936 |
| *X0 = [20, -20], Eps = 0.000001, alpha = 30* | | | |
| Iter | 15 | 34 | 20 |
| Value | 838 | 1931 | 1088 |
| *X0 = [20, -20], Eps = 0.01, alpha = 133* | | | |
| Iter | 17 | 17 | 17 |
| Value | 977 | 978 | 977 |
| *X0 = [20, -20], Eps = 0.000001, alpha = 133* | | | |
| Iter | 20 | 21 | 22 |
| Value | 1132 | 1133 | 1122 |

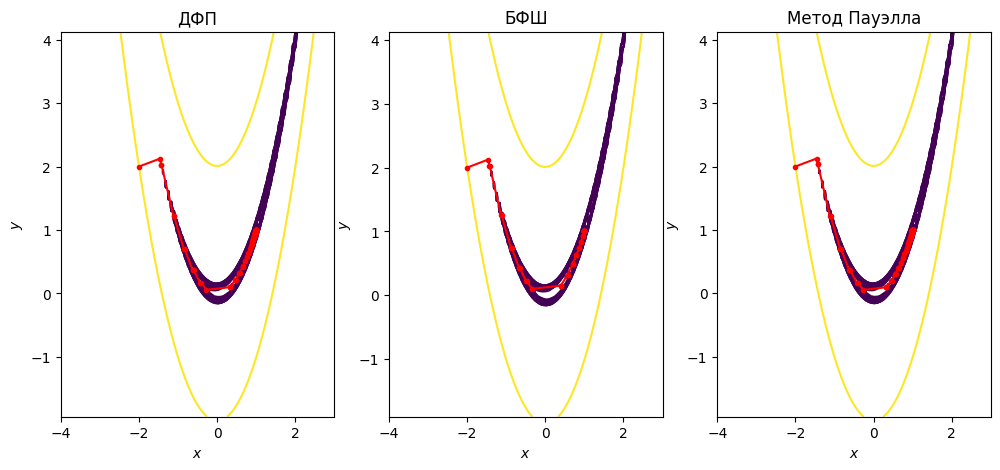


Рисунок 2. Функция Розенброка. Начальная точка - (-2, 2), eps = 0.000001, alpha = 200.



Рисунок 3. Метод ДФП. Начальная точка - (-5, 10), eps = 0.01, alpha = 1.

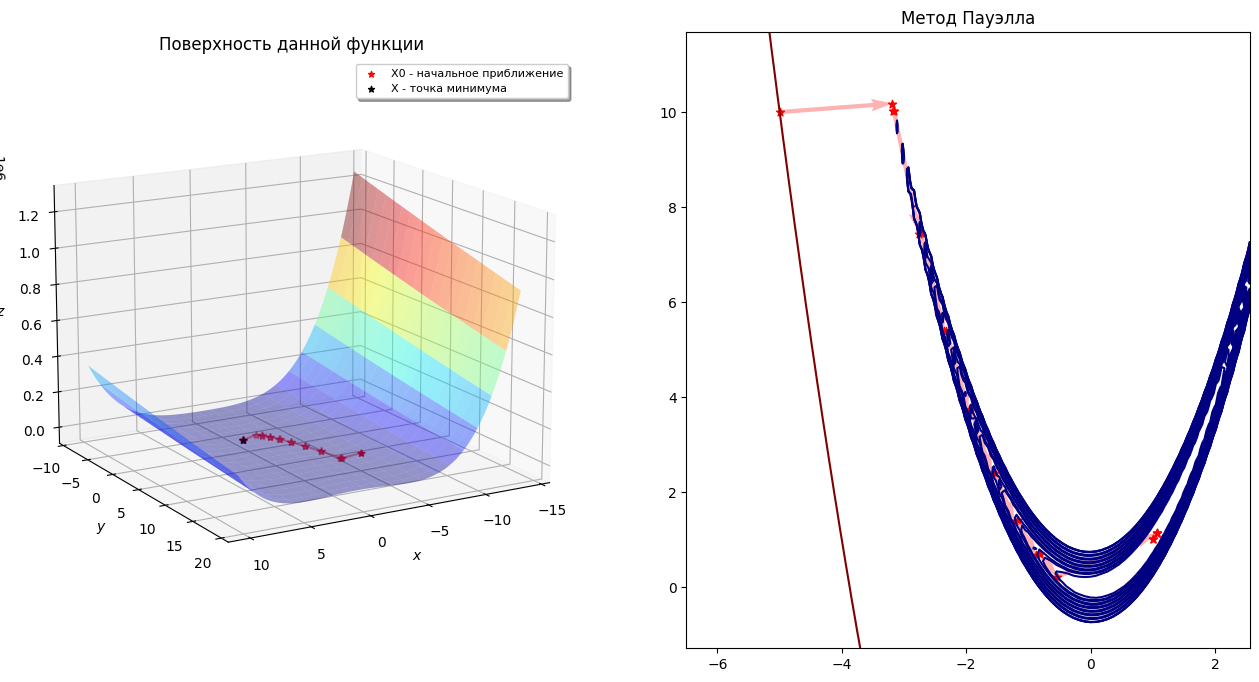


Рисунок 4. Метод Пауэлла. Начальная точка - (-5, 10), eps = 0.01, alpha = 30.

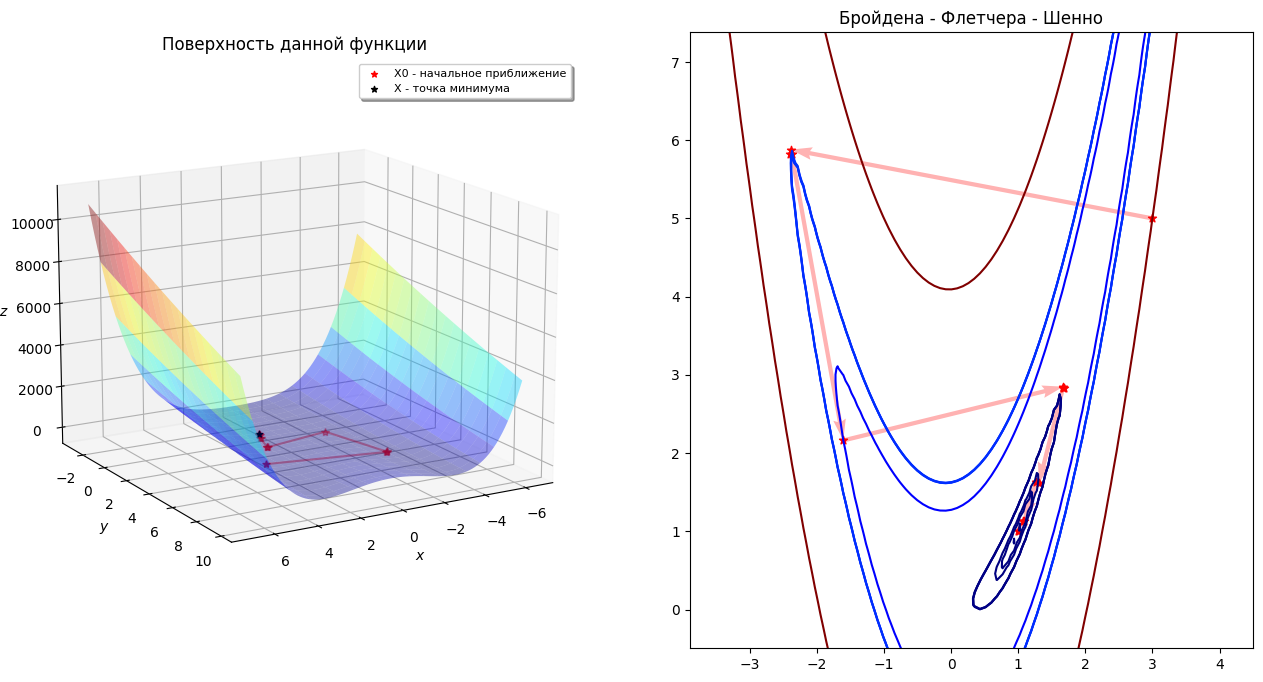


Рисунок 5. БФШ. Начальная точка - (3, 5), eps = 0.01, alpha = 4.

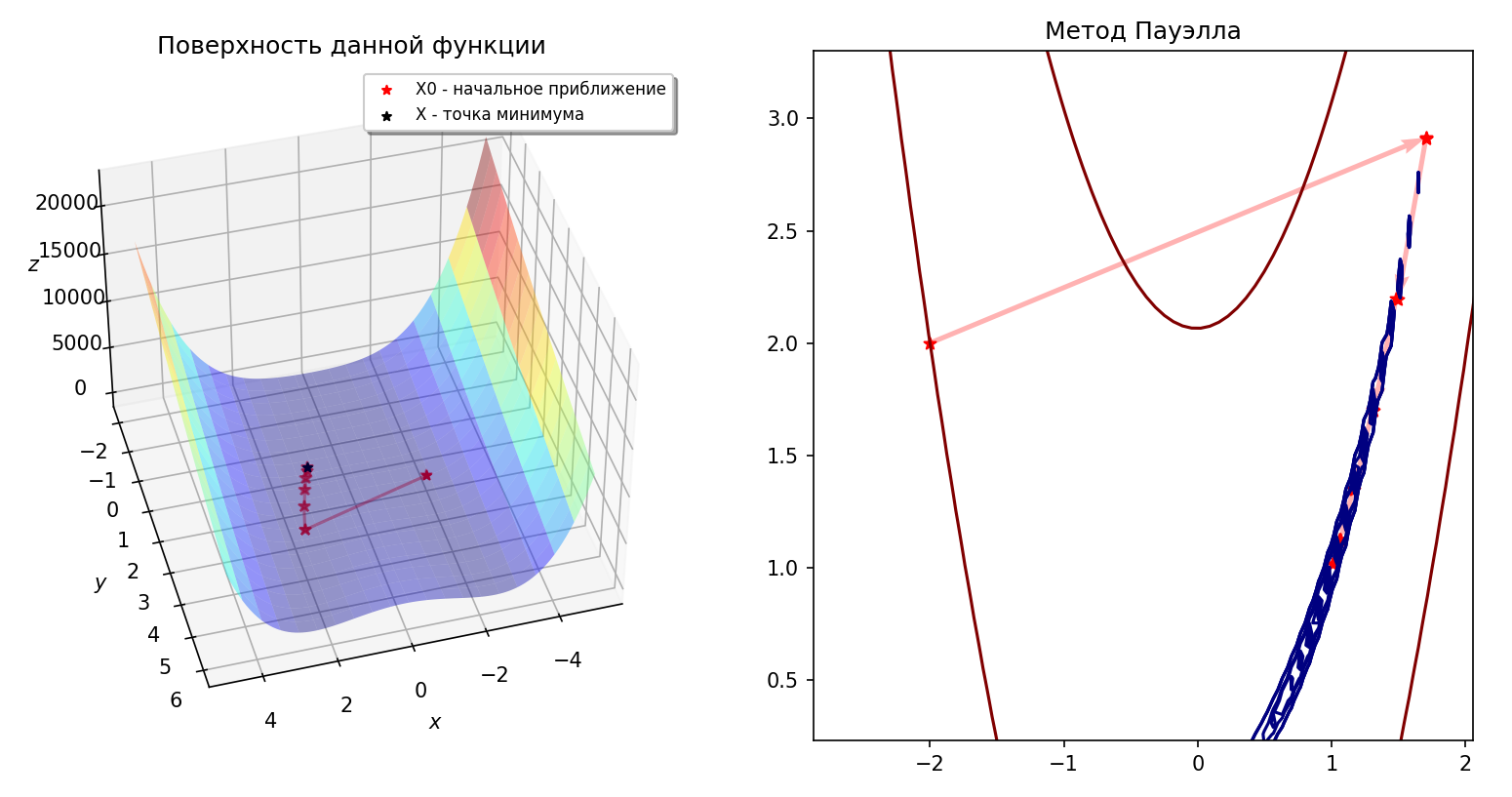


Рисунок 6. Метод Пауэлла. Начальная точка - (-2, 2), eps = 0.01.

* 1. Вывод

Рассмотренные выше алгоритмы объединяют достоинства метода наискорейшего спуска и метода Ньютона. При их применении удается сохранить высокую скорость сходимости алгоритмов, не вычисляя обратную матрицу Гессе. Также, квазиньютоновские методы относятся к методам первого порядка, а значит не требуется вычислять вторые производные.

Положительно определенная матрица Аk обеспечивает направление спуска, в то время как матрица Гессе может быть не определена, что несет за собой дополнительные вычисления. Сложность алгоритма, в отличие от метода Ньютона ()), ()). В результате многочисленных тестов, самым выгодным методом оказался метод БФШ.

Все квазиньютоновские методы отличаются между собой лишь способом обновления матрицы Ak.

Увеличивая овражность функции Розенброка в 4 раза, в два раза возрастают вычисления у методов БФШ и Пауэлла, у ДФП – в три раза. Метод БФШ чувствителен к выбору начальной точки. По результатам графиков, при увеличении овражности метод БФШ оказывается выгодней.

Взяв начальное приближение, отдаленное от точки минимума, количество вычислений у метода БФШ возрастает в 2 раза, у методов ДФП и Пауэлла – в 1.2 раза.

Увеличивая точность с 1е-3 до 1е-6, можно сказать, что количество вычислений функций не зависят от точности.

# Методы прямого поиска

Методы прямого поиска:

* метод циклического покоординатного спуска;
* метод Хука-Дживса;
* метод Розенброка.

3. 1. Результаты

*Таблица 1. Квадратичная функция.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы** | **ЦПС** | **Хука-Дживса** | **Розенброк** |
| *X0 = [-2, -1], eps = 0.01* | | | |
| Iter | 10 | 7 | 4 |
| Value | 480 | 337 | 519 |
| *X0 = [-2, -1], eps = 0.000001* | | | |
| Iter | 22 | 14 | 4 |
| Value | 1253 | 752 | 519 |

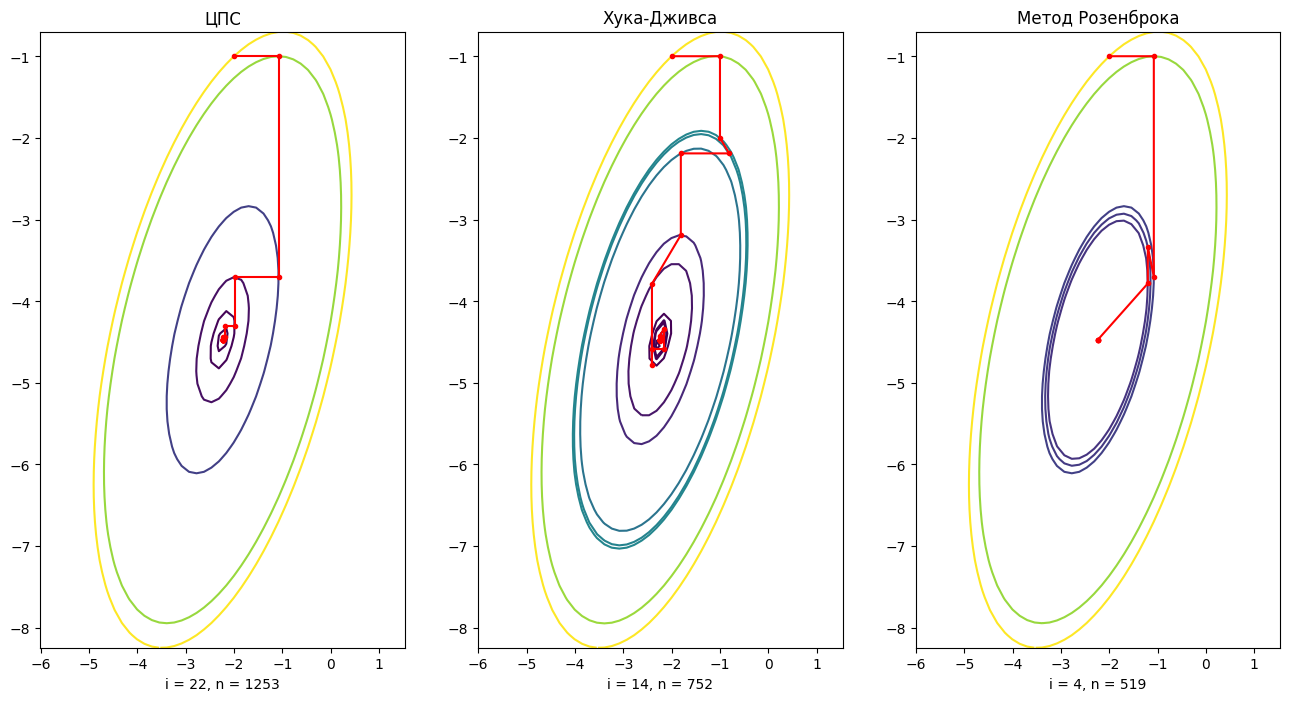
­

Рисунок 7. Квадратичная функция. Начальная точка - (-2, -1), eps = 0.000001.

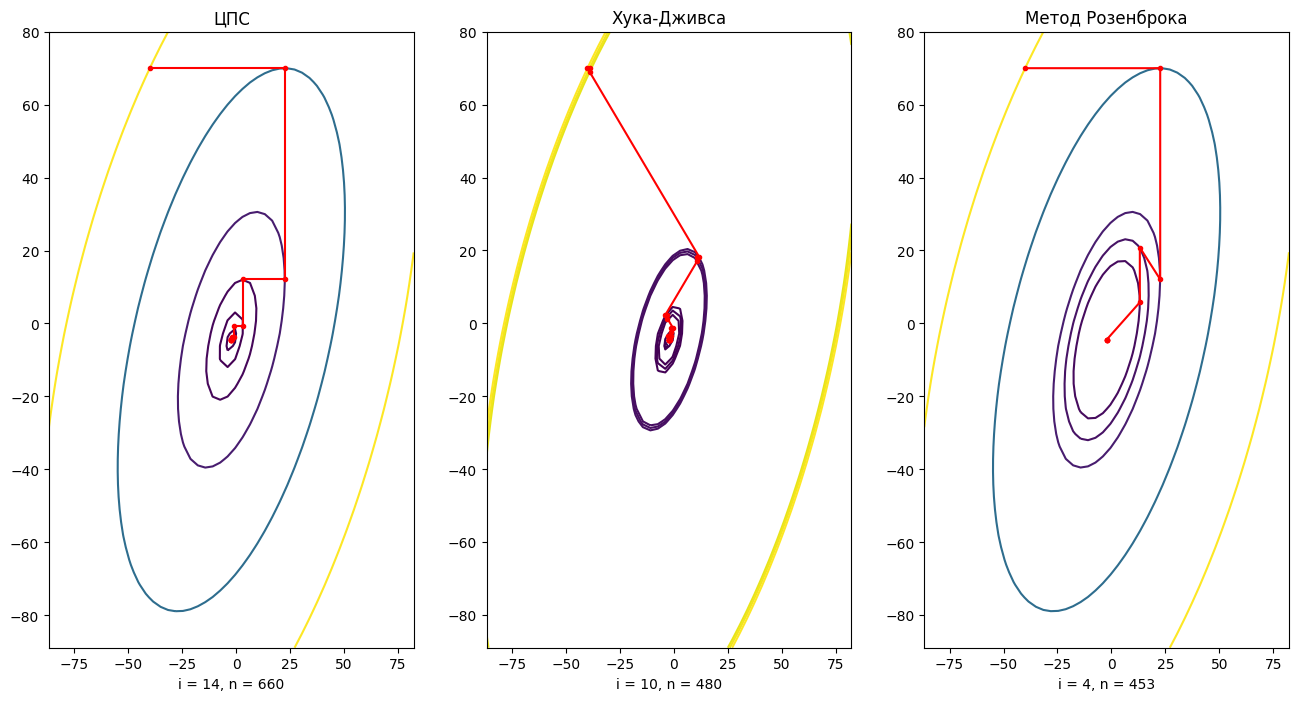


Рисунок 8. Начальная точка - (-40, 70), eps = 0.01.

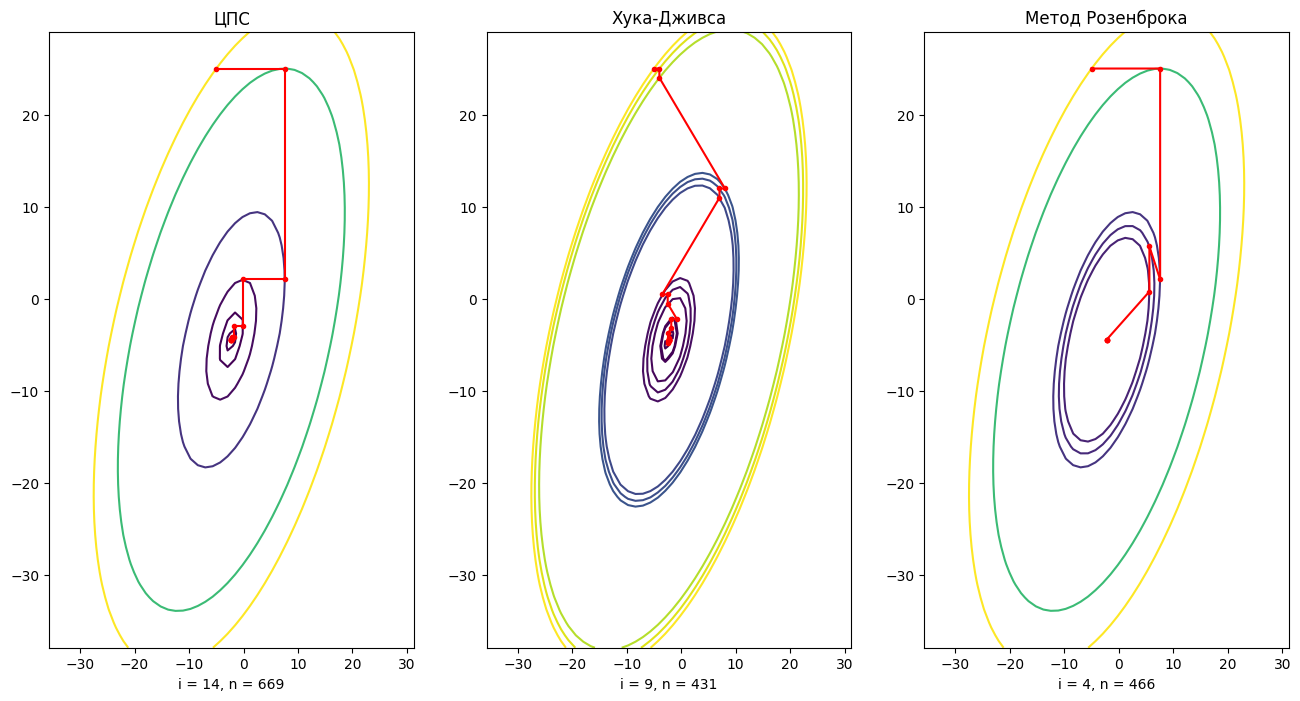
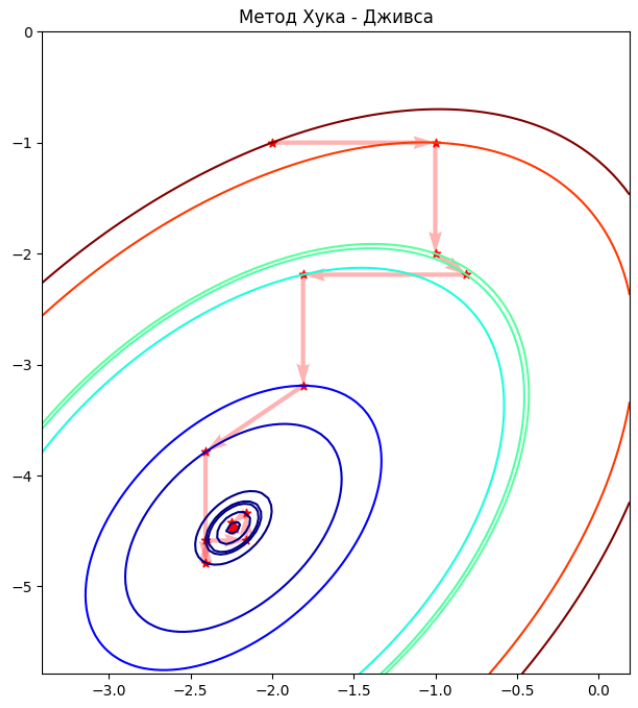
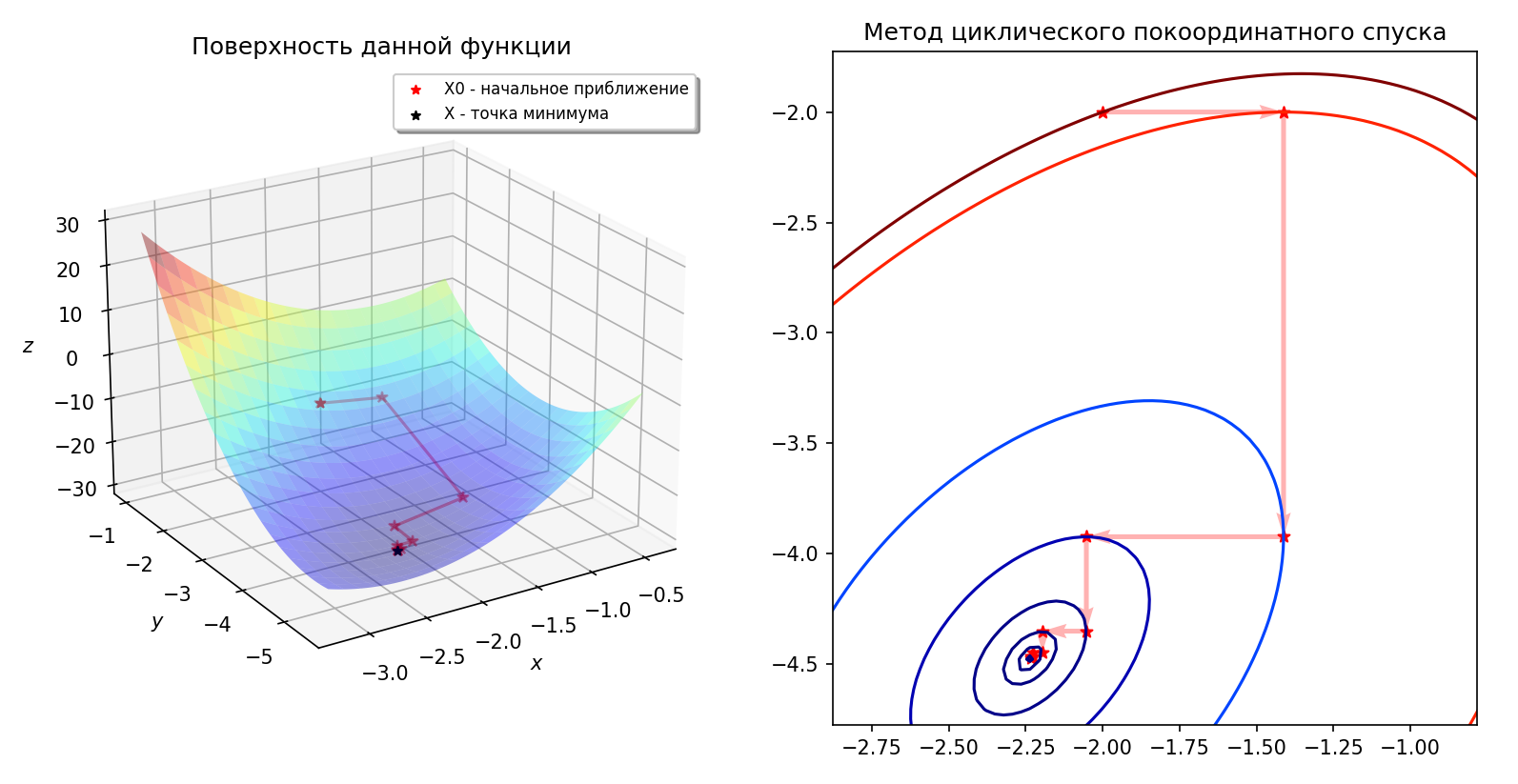


Рисунок 9. Начальная точка - (-5, 25), eps = 0.01.



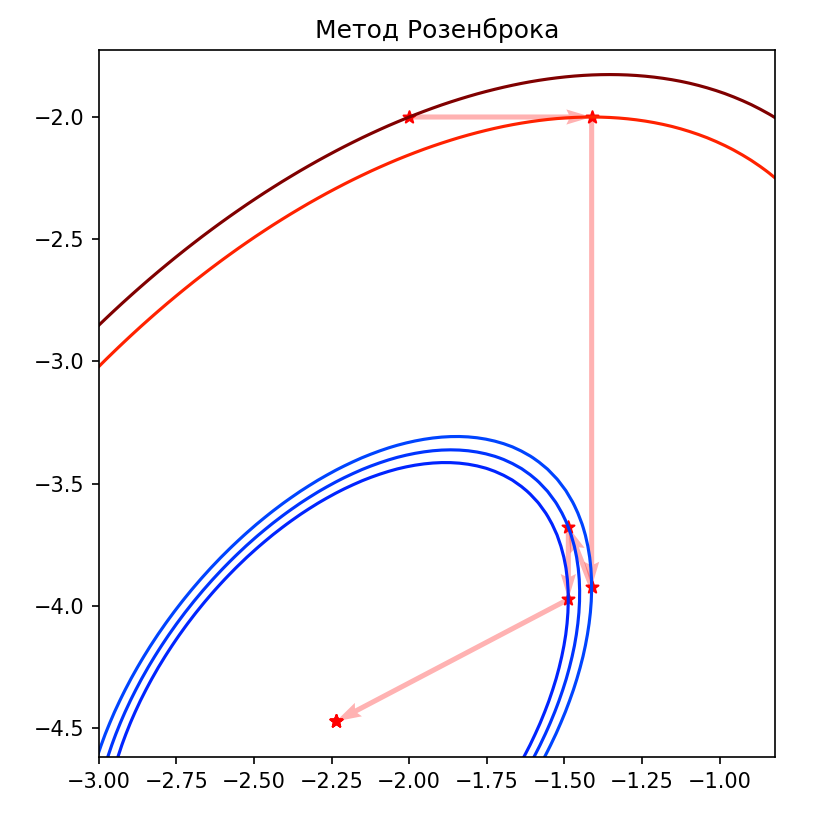


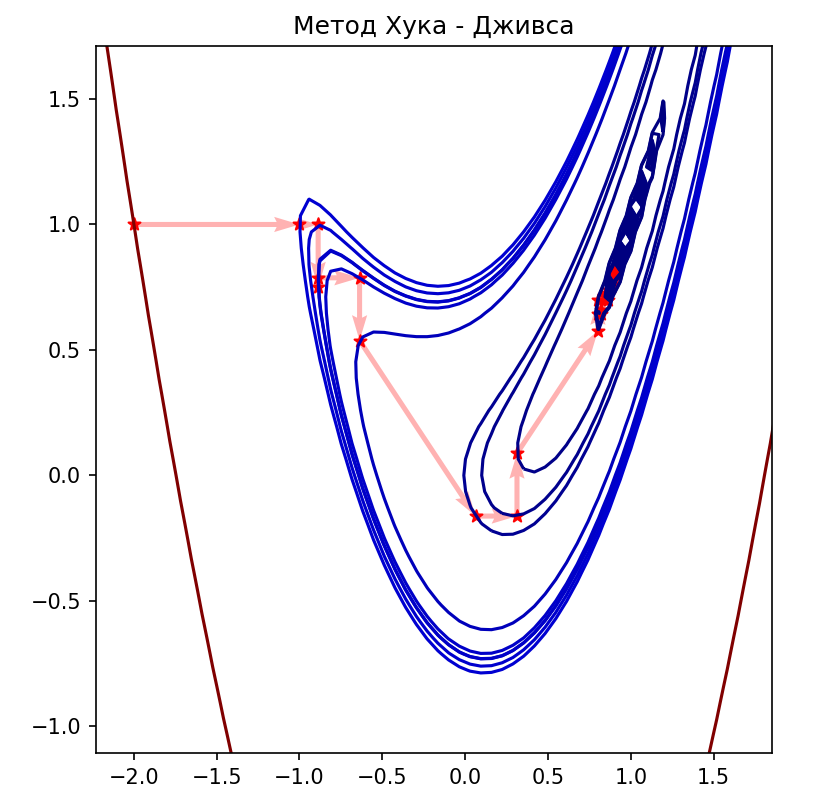
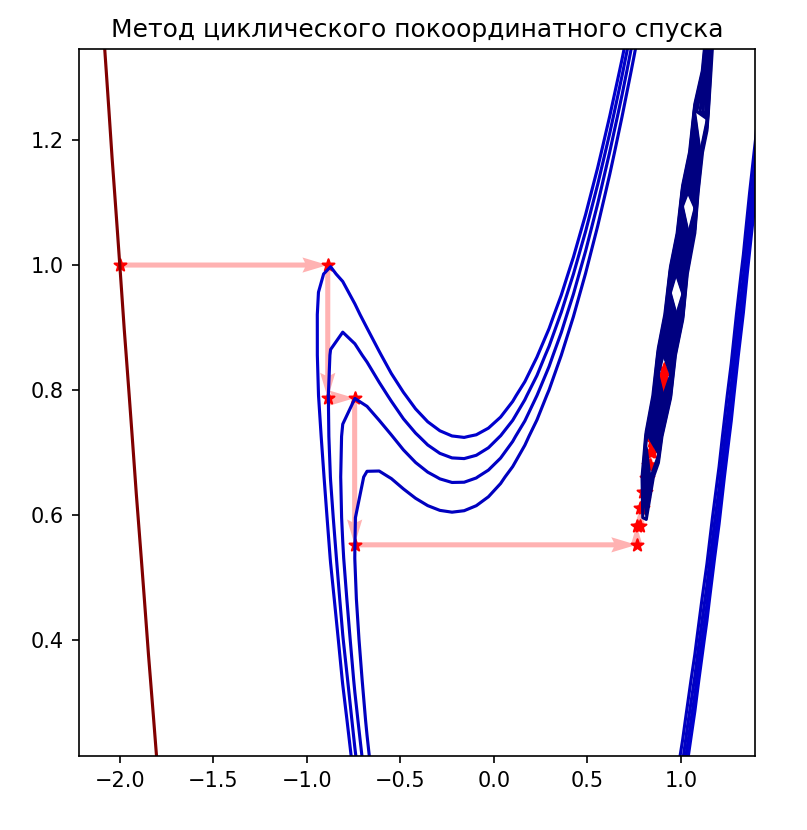
Рисунок 10. Начальная точка - (-2, -2), eps = 0.01.

*Таблица 2. Функция Розенброка.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод** | **ЦПС** | **Хука-Дживса** | **Розенброк** |
| *X0 = [-4, -4], Eps = 0.01, alpha = 30* | | | |
| Iter | 44 | 6 | 13 |
| Value | 2324 | 290 | 1356 |
| *X0 = [-4, -4], Eps = 0.000001, alpha = 30* | | | |
| Iter | 1988 | 809 | 15 |
| Value | 130224 | 39037 | 1705 |
| *X0 = [-4, -4], Eps = 0.01, alpha = 133* | | | |
| Iter | 494 | 30 | 20 |
| Value | 28642 | 1468 | 2192 |
| *X0 = [-4, -4], Eps = 0.000001, alpha = 133* | | | |
| Iter | 7178 | 2811 | 21 |
| Value | 480953 | 135520 | 2398 |
| *X0 = [-50, -50], Eps = 0.01, alpha = 30* | | | |
| Iter | 44 | 6 | 13 |
| Value | 2331 | 293 | 1363 |
| *X0 = [-50, -50], Eps = 0.000001, alpha = 30* | | | |
| Iter | 1988 | 809 | 15 |
| Value | 130231 | 39041 | 1712 |
| *X0 = [-50, -50], Eps = 0.01, alpha = 133* | | | |
| Iter | 494 | 30 | 20 |
| Value | 28649 | 1471 | 2199 |
| *X0 = [-50, -50], Eps = 0.000001, alpha = 133* | | | |
| Iter | 7178 | 2811 | 21 |
| Value | 460960 | 135522 | 2403 |

**Вывод:** ЦПС проигрывает обоим методам. Методы Хука – Дживса и Розенброка одинаково себя ведут.

**Вывод:** Лучшимметодом оказался метод Розенброка при большой точности.



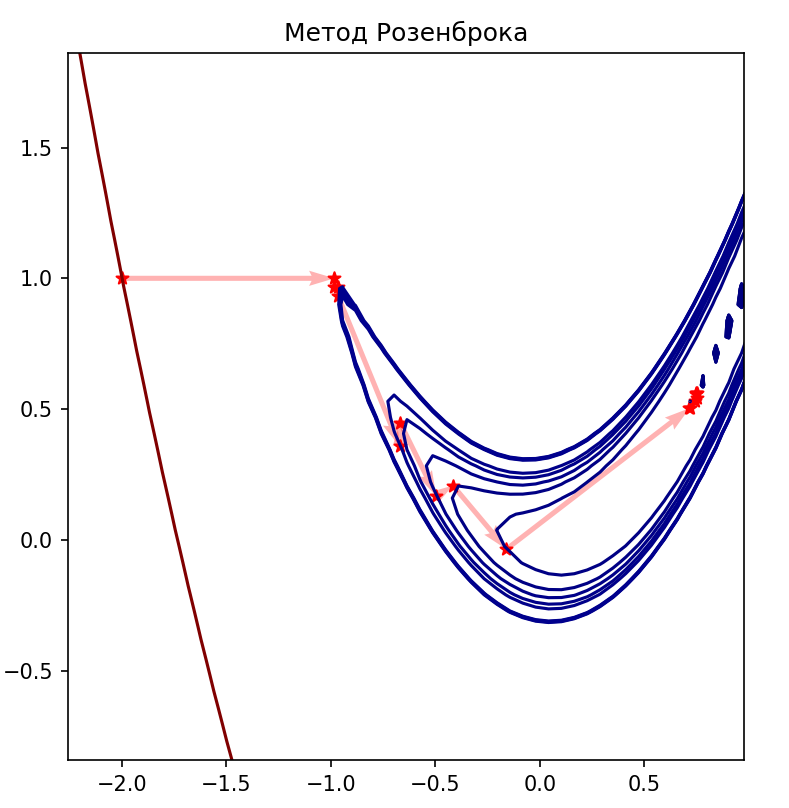
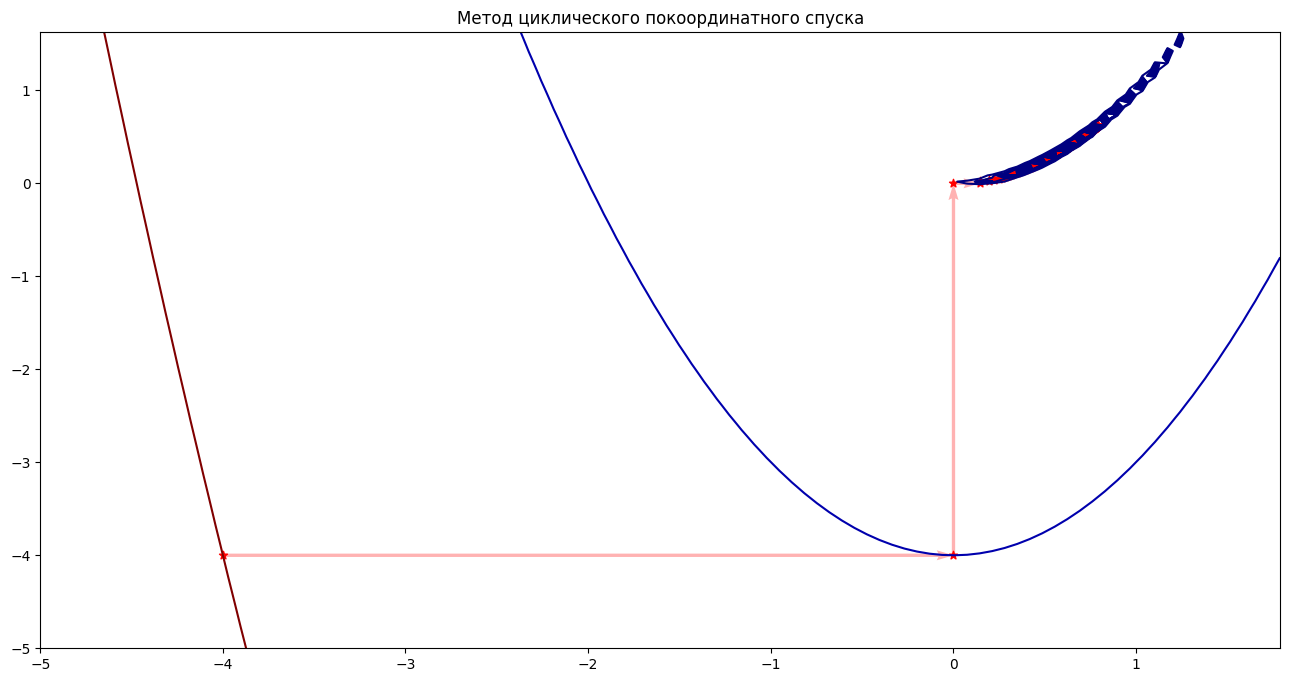
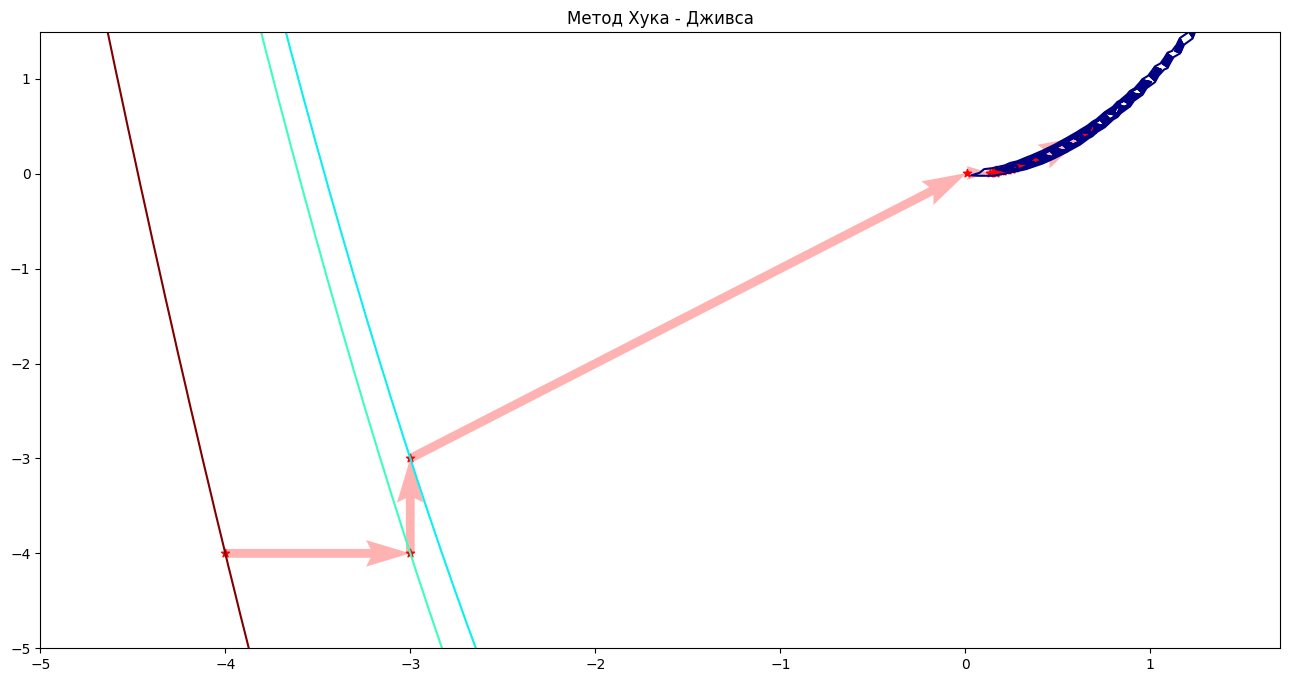


Рисунок 11. Начальная точка - (-2, 1), eps = 30.

****

****

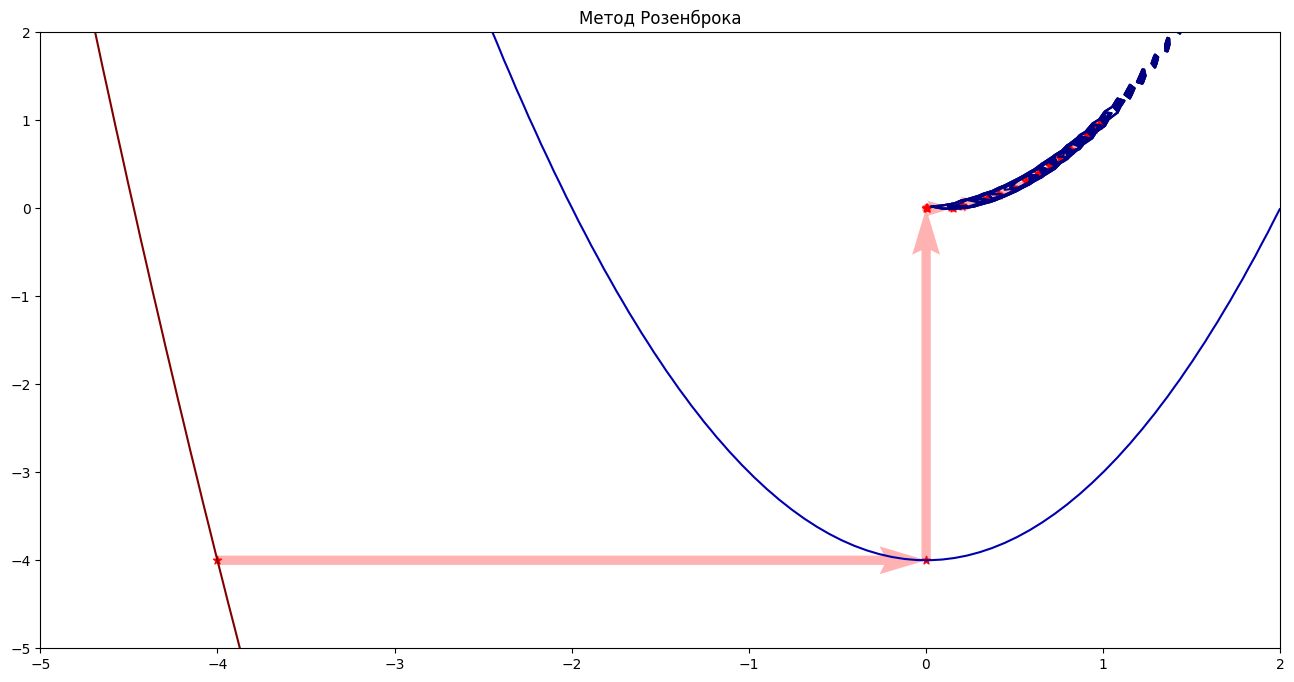
****

Рисунок 12. Функция Розенброка. Начальная точка - (-4, -4), eps = 0.001.

3. 2. Вывод

Методы прямого поиска, по сравнению с методами первого и второго порядков, не требуют вычислений градиента и матрицы Гессе. Следовательно, не требуется дифференцируемости целых функций. При высокой точности, минимум выгодней считать методом Розенброка. Всех хуже себя показывает метод циклического покоординатного спуска.

Таким образом, мы рассмотрели алгоритмы прямого поиска, в которых используется информация только о значениях функции. Достоинством данных методов является то, что нам не нужно требовать дифференцируемости функции и прочих условий. Однако из-за этого появляются недостатки: эффективность такого метода сложно оценить. Метод покоординатного спуска является самым простым для реализации, однако его простота делает его не самым эффективным с точки зрения количества итераций. Методы Хука – Дживса и Розенброка являются эффективными с точки зрения количества итераций и вычислений функции. Наиболее выгодным является метод Хука – Дживса. Следует отметить, что при сильно овражной функции и маленькой точности методы ЦПС и Хука – Дживса ищут минимум не точно. Для точного поиска (методом Розенброка) следует жертвовать производительностью.

Методы ЦПС и Хука – Дживса сильно зависят от точности и начального приближения. Например, повысив точность с 1е-3 до 1е-6, количество вычислений функций возрастает в 150 – 200 раз. Что касается метода Розенброка, количество вычислений возрастают всего в 1.5 раза. Метод Розенброка не зависит от начального приближения. ЦПС проигрывает по всем параметрам.

**Выгодным методом оказался метод Розенброка.**

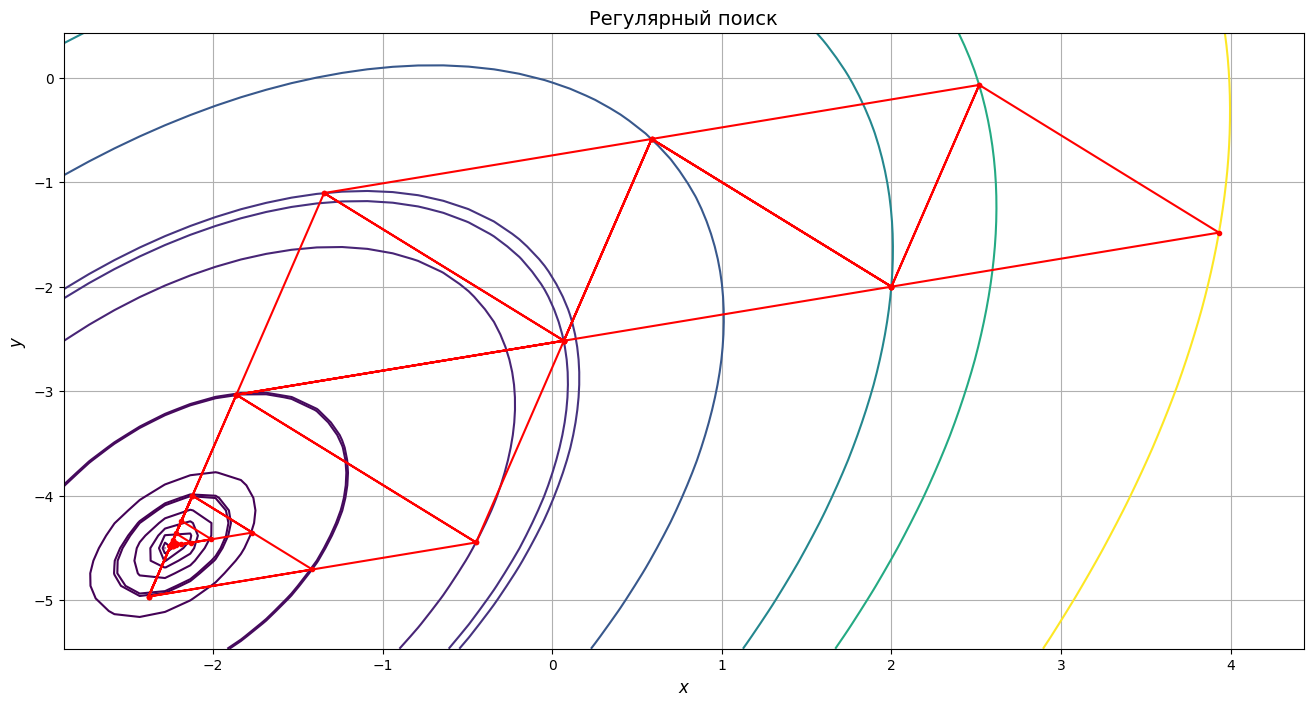
# Симплекс – методы

Симплекс – метод:

* регулярный симплекс;
* нерегулярный симплекс (метод Недлера – Мида).
  1. Результаты

*Таблица 1. Квадратичная функция.*

|  |  |  |
| --- | --- | --- |
| **Методы** | **Регулярный симплекс** | **Недлера – Мида** |
| *X0 = [5, -5], eps = 0.001* | | |
| Iter | 26 | 20 |
| Value | 82 | 180 |
| *X0 = [5, -5], eps = 0.000001* | | |
| Iter | 41 | 36 |
| Value | 122 | 320 |
| *X\* = [-2.23, -4.47], F(X\*) = -27.99* | | |

****

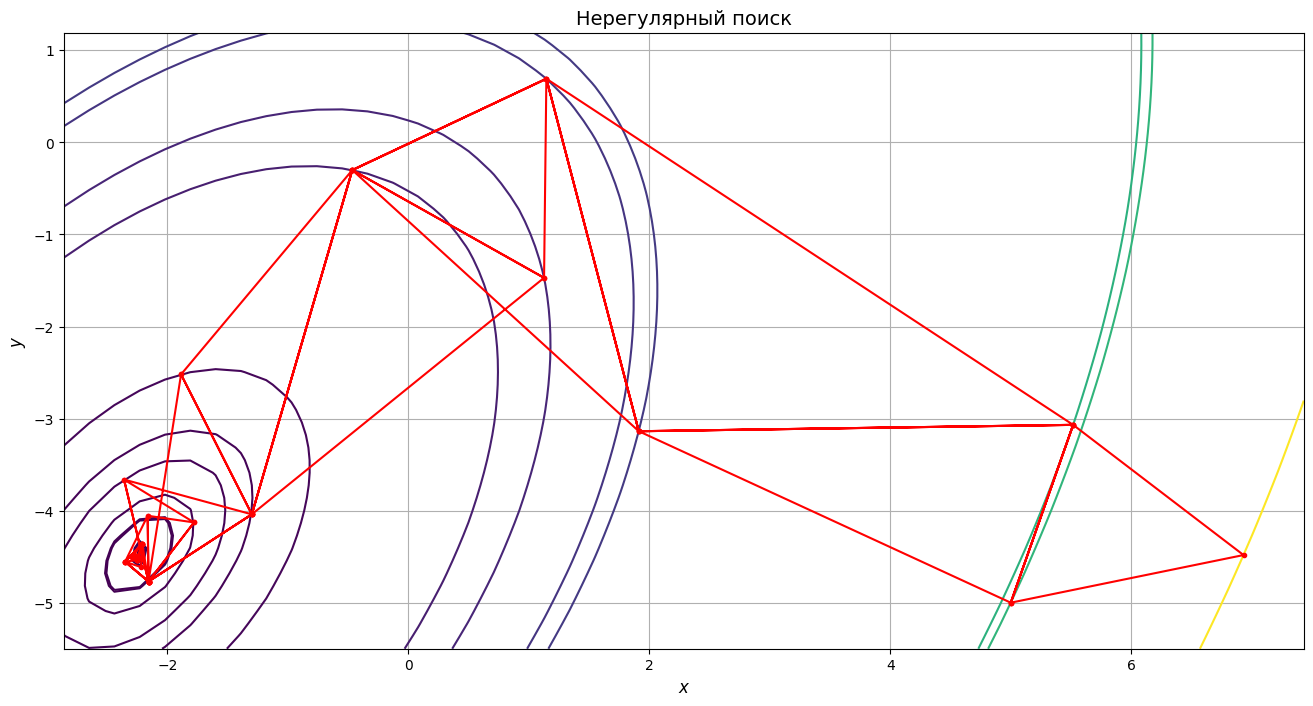
****

Рисунок 13. Квадратичная функция. Начальная точка - (5, -5), eps = 0.001.

*Таблица 2. Функция Розенброка.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы** | **Регулярный симплекс** | | **Недлера – Мида** |
| *X0 = [2, -2], Eps = 0.001, alpha = 30* | | | |
| Iter | 20 | | 55(62) |
| Value | 66 | | 490(118) |
| *X\* = [0.98253, 0.96353], F(X\*) = 0.00025* | | *X\* = [0.9677, 0.9347], F(X\*) = 0.0011* | |
| *X0 = [2, -2], Eps = 0.000001, alpha = 30* | | | |
| Iter | 3334 | | 75(62) |
| Value | 13294 | | 660(118) |
| *X\* = [0.99, 0.99], F(X\*) = 4.51e-08* | | *X\* = [0.9984, 0.9968], F(X\*) = 2.57e-06* | |
| *X0 = [2, -2], Eps = 0.001, alpha = 133* | | | |
| Iter | 18 | | 94(76) |
| Value | 50 | | 830(142) |
| *X\* = 1.022, 1.0471], F(X\*) = 0.0006* | | *X\* = 0.9981, 0.9984], F(X\*) = 0.0006* | |
| *X0 = [2, -2], Eps = 0.000001, alpha = 133* | | | |
| Iter | 11491 | | 129(76) |
| Value | 45922 | | 1130(142) |
| *X\* = [1.00, 1.00], F(X\*) = 8.27e-07* | | *X\* = [0.999, 0.999], F(X\*) = 9.55e-08* | |
| *X0 = [50, -70], Eps = 0.001, alpha = 30* | | | |
| Iter | 255 | | 60(91) |
| Value | 998 | | 526(167) |
| *X\* = [1.0249, 1.0507], F(X\*) = 0.0006* | | *X\* = [1.0249, 1.0507], F(X\*) = 0.0006* | |
| *X0 = [50, -70], Eps = 0.000001, alpha = 30* | | | |
| Iter | 3975 | | 366(91) |
| Value | 15858 | | 3206(167) |
| *X\* = [1.00, 1.00], F(X\*) = 4.69e-08* | | *X\* = [0.9989, 0.9978], F(X\*) = 1.22e-06* | |
| *X0 = [50, -70], Eps = 0.001, alpha = 133* | | | |
| Iter | 252 | | 91(131) |
| Value | 986 | | 798(242) |
| *X\* = [0.802, 0.643], F(X\*) = 0.039* | | *X\* = [0.802, 0.643], F(X\*) = 0.039* | |
| *X0 = [50, -70], Eps = 0.000001, alpha = 133* | | | |
| *X\* = [-0.69, 0.47], F(X\*) = 2.86* | | | |
| Iter | 17234 | | 1411(131) |
| Value | 68894 | | 12348(242) |
| *X\* = [0.99, 0.99], F(X\*) = 9.57e-08* | | *X\* = [1.005, 1.010], F(X\*) = 2.79e-05* | |

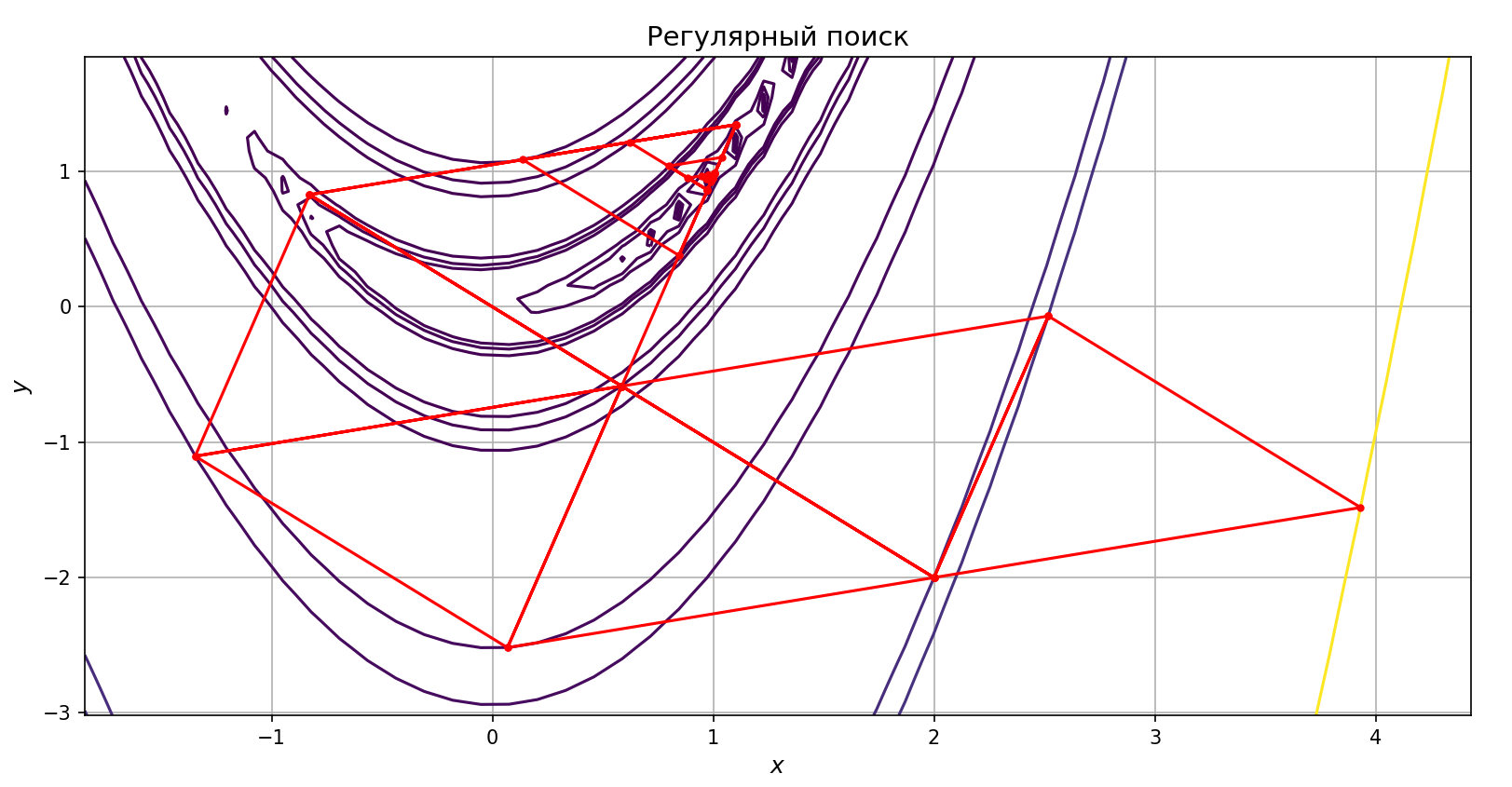


Рисунок 14. Функция Розенброка. Начальная точка - (2, -2), eps = 0.001.

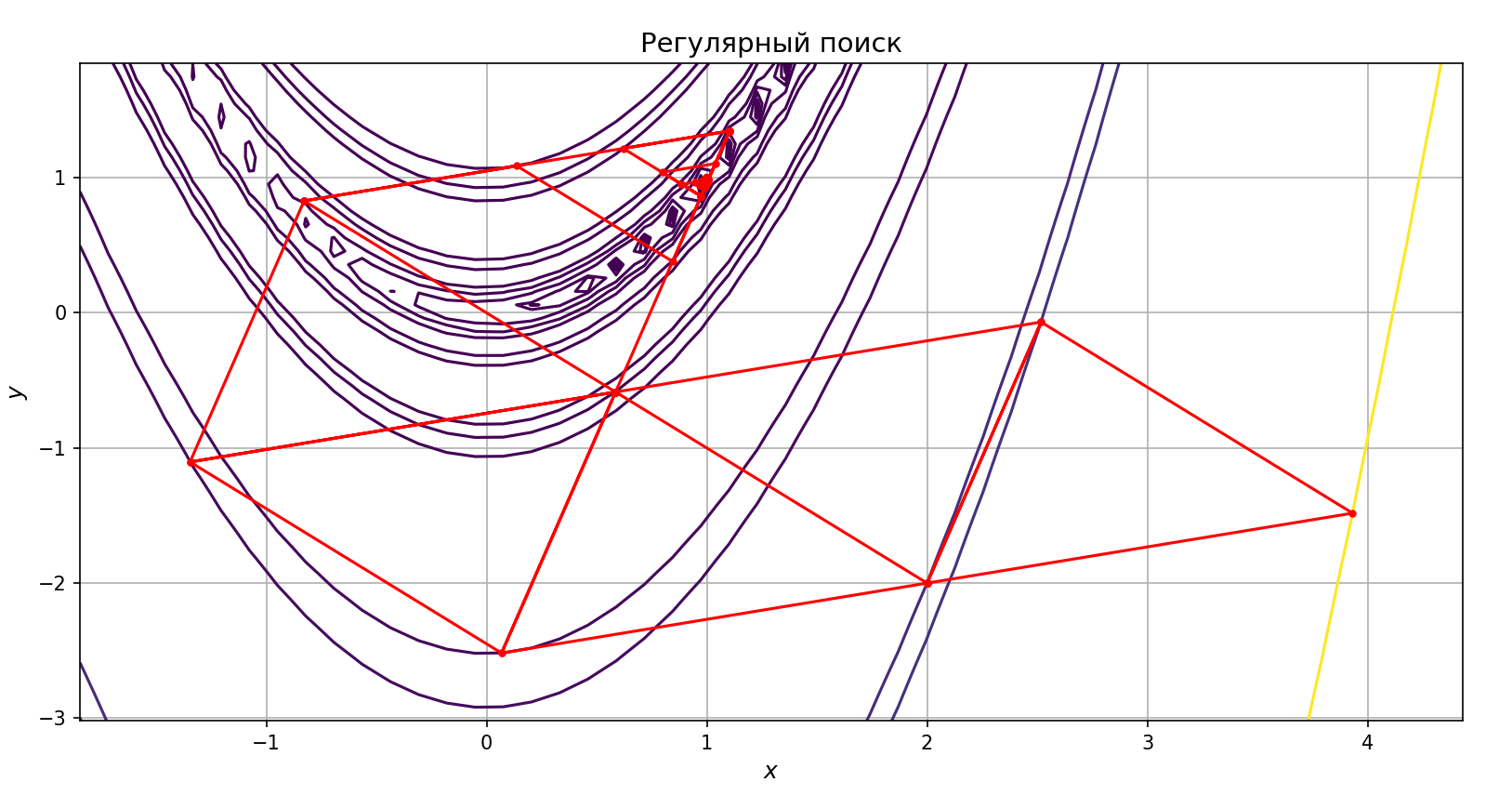


Рисунок 15. Функция Розенброка. Начальная точка - (2, -2), eps = 0.000001.

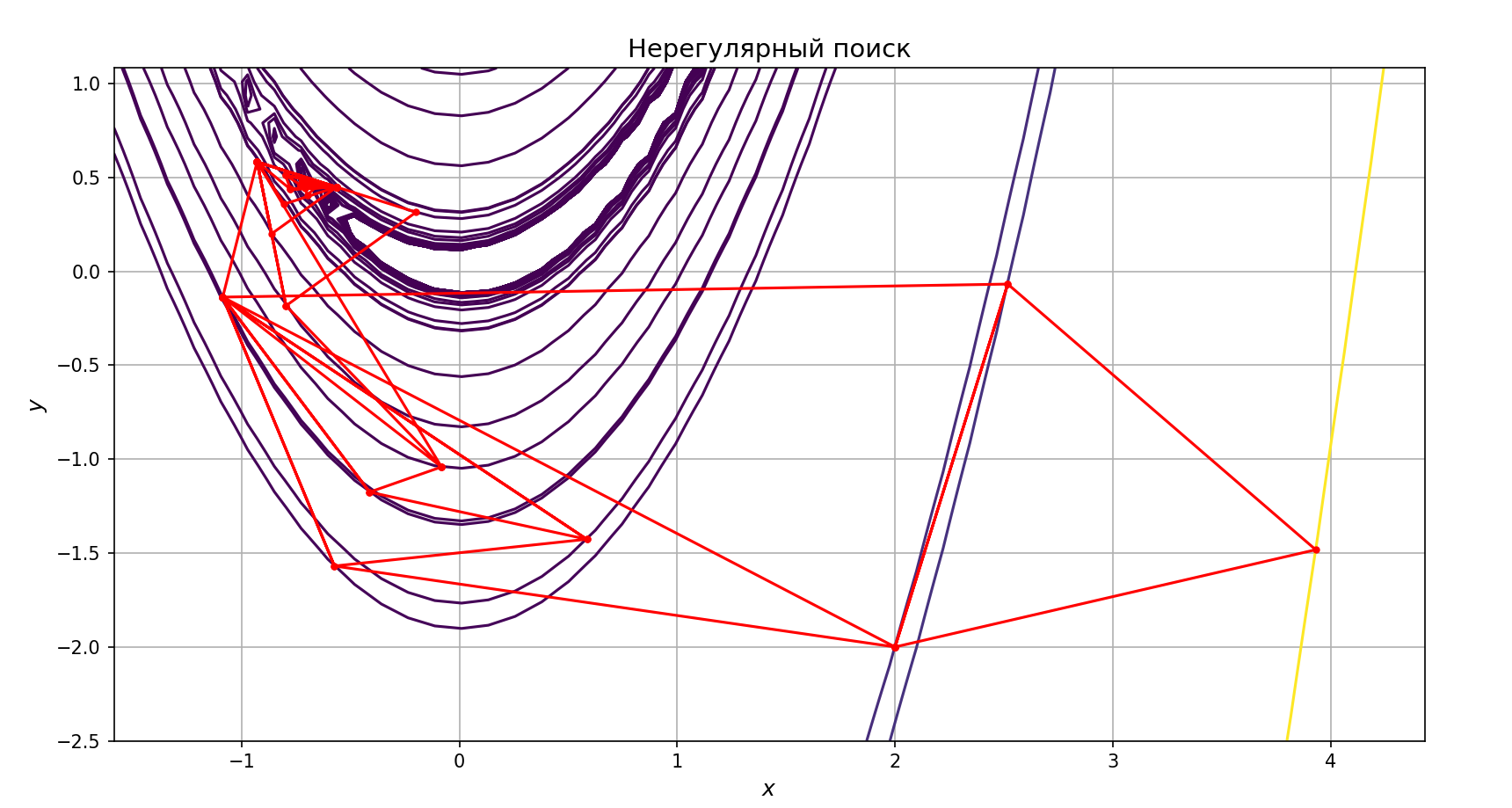


Рисунок 16. Функция Розенброка. Начальная точка - (2, -2), eps = 0.001.

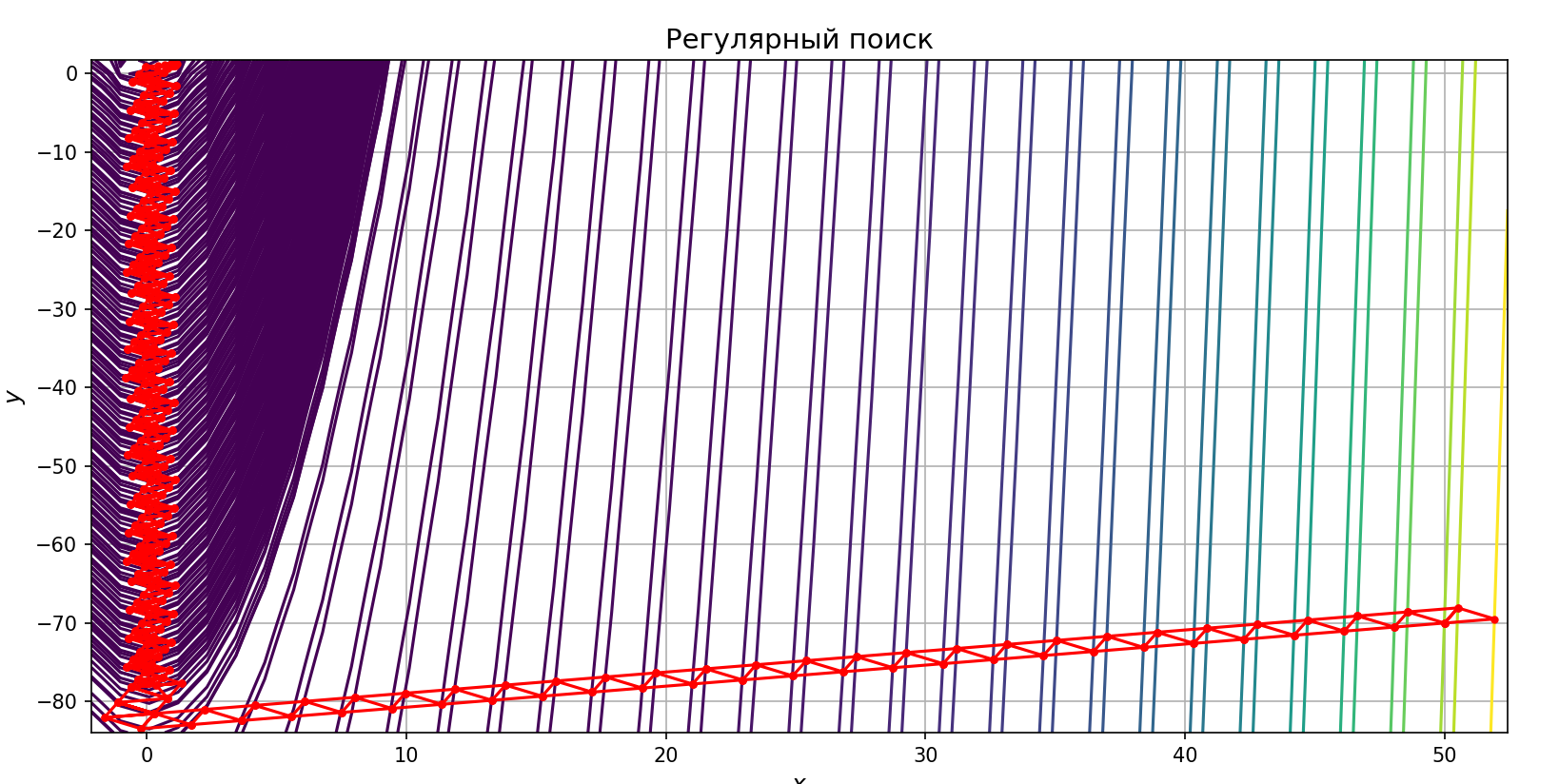


Рисунок 17.Функция Розенброка. Начальная точка - (50, -70), eps = 0.001.

* 1. Вывод

Методы симплексного поиска имеют преимущества перед методами 1го и 2го-порядка, во-первых, не нужно считать производную и матрицы Гесса, во-вторых, методы симплексного поиска всегда сходятся, если есть глобальный минимум, но есть вероятность попасть в локальный минимум при одномерной минимизации определенной координаты. Симплексный поиск с использованием регулярного симплекса эффективен лишь в случаях, когда график целевой функции имеет достаточно простую топографию. Если график целевой функции имеет овражную структуру, использование регулярного симплекса менее эффективно: регулярный симплекс, скатываясь по поверхности плохо в нее вписывается, поэтому лучше использовать нерегулярный симплекс. Функция с овражной структурой – к примеру, функция Розенброка. На квадратичной функции методы симплексного поиска показывают хорошие результаты, на Розенброка лучшие результаты у метода нерегулярного симплекса. Но в общем случае хоть и класс функций для нахождения экстремумов шире, чем класс при использовании методов 1го и 2го порядка, нет гарантий, что метод не придет в локальный минимум, вероятность этого выше, чем при использовании методов более высокого порядка.

Минимизируя квадратичную функцию, при повышении точности в два раза, в два раза возрастают вычисления функции.

При увеличении точности с 1e-3 до 1e-6, количество вычислений функции регулярного симплекса возрастают в 200 раз.

Выбор точки влияет на количество вычислений функций обоих методов, особенно на регулярном симплексе (количество вычислений увеличивается в 10 раз).  
При увеличении овражности регулярный симплекс не стоит использовать, кол-во вычислений функции. Нерегулярный симплекс показывает лучше результаты, чем регулярный.

Рассматривая квадратичную функцию, регулярный более выгодный, чем нерегулярный.

# Методы последовательной безусловной минимизации

Методы:

* внутренних штрафных функций;
* внешних штрафных функций.
  1. Результаты

***Треугольник***

*Таблица 1. Квадратичная функция.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы штрафных функций** | **Внутренних** | | **Внешних** |
| *X0 = [5, -5], eps = 0.001* | | | |
| Iter | 64 | | 3 |
| Value | 2944 | | 158 |
| *X\* = [0.00, 0.00], F(X\*) = 22.003* | | *X\* = [0.00, 0.00], F(X\*) = 22.00* | |
| *X0 = [5, -5], eps = 0.000001* | | | |
| Iter | 123 | | 11 |
| Value | 9102 | | 911 |
| *X\* = [0.00, 0.00], F(X\*) = 22.003* | | *X\* = [0.00, 0.00], F(X\*) = 22.00* | |

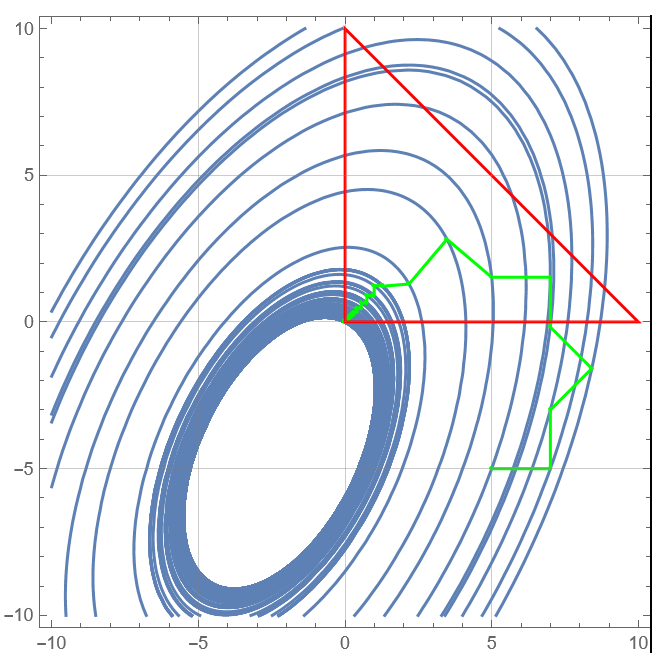


Рисунок 18. Внутренние гирафные функции. Область - треугольник.

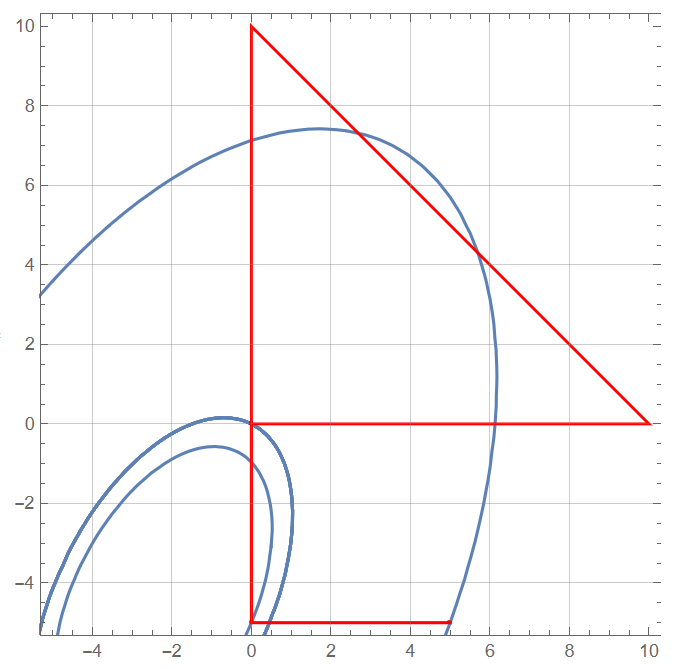


Рисунок 19. Метод внешних штрафных функций. Область треугольник.

***Эллипс***

*Таблица 1. Квадратичная функция.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы штрафных функций** | **Внутренних** | | **Внешних** |
| *X0 = [-5, -5], eps = 0.001* | | | |
| Iter | 6 | | 6 |
| Value | 276 | | 314 |
| *X\* = [-2.236, -4.472], F(X\*) = -28.00* | | *X\* = [-2.236, -4.472], F(X\*) = -28.00* | |
| *X0 = [-5, -5], eps = 0.000001* | | | |
| Iter | 7 | | 8 |
| Value | 518 | | 658 |
| *X\* = [-2.23607, -4.47214], F(X\*) = -28.00* | | *X\* = [-2.23686, -4.47224], F(X\*) = -28.00* | |

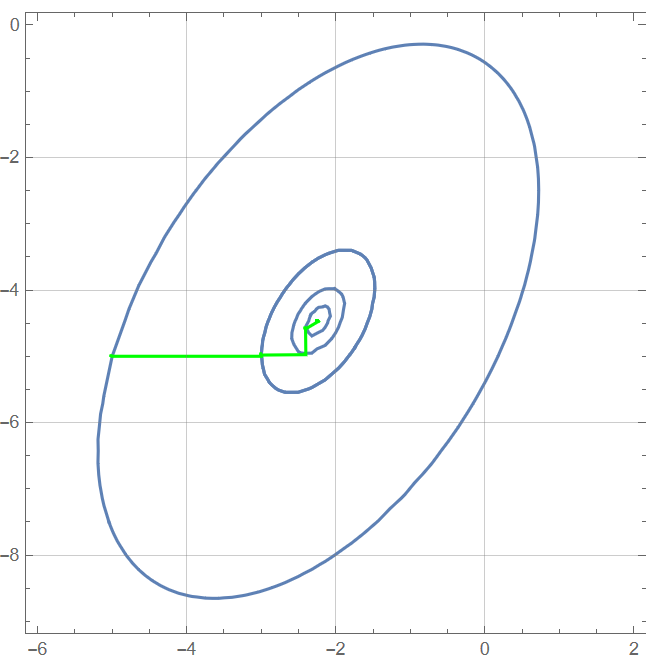


Рисунок 20. Метод внутренних штрафных функций.

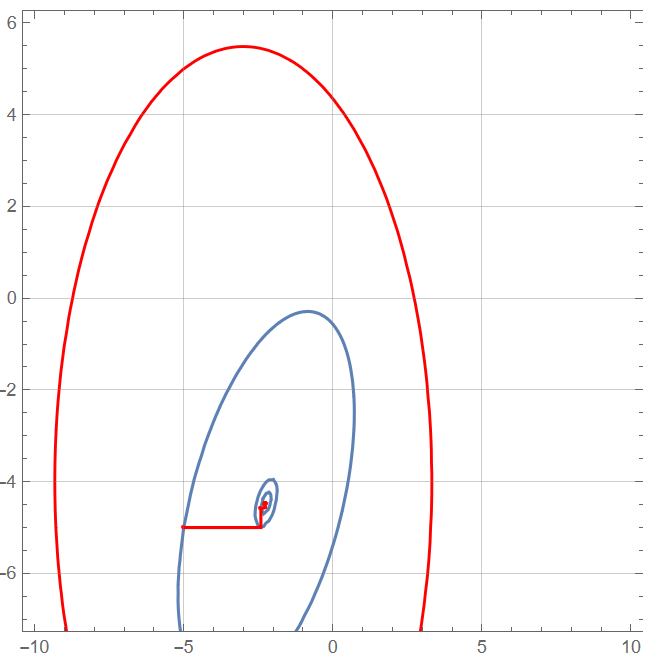


Рисунок 21. Метод внешних штрафных функции.

***Эллипс***

*Таблица 2. Функция Розенброка.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Методы штрафных функций** | **Внутренних** | | **Внешних** |
| *X0 = [-4, -4], Eps = 0.001, alpha = 30* | | | |
| Iter | 13 | | 10 |
| Value | 598 | | 522 |
| *X\* = [1.00, 1.00], F(X\*) = 0.00* | | *X\* = [0.997, 0.993], F(X\*) = 0.000* | |
| *X0 = [-4, -4], Eps = 0.000001, alpha = 30* | | | |
| Iter | 14 | | 169 |
| Value | 1036 | | 13890 |
| *X\* = [1.00, 1.00], F(X\*) = 0.00* | | *X\* = [1.00, 1.00], F(X\*) = 0.00* | |
| *X0 = [-4, -4], Eps = 0.001, alpha = 133* | | | |
| Iter | 17 | | 25 |
| Value | 782 | | 1302 |
| *X\* = [1.00, 1.00], F(X\*) = 0.00* | | *X\* = [1.00, 1.00], F(X\*) = 0.00* | |
| *X0 = [-4, -4], Eps = 0.000001, alpha = 133* | | | |
| Iter | 18 | | 48 |
| Value | 1332 | | 3939 |
| *X\* = [1.00, 1.00], F(X\*) = 0.00* | | *X\* = [1.00, 1.00], F(X\*) = 0.00* | |
| *X0 = [-3, 3], Eps = 0.001, alpha = 5* | | | |
| Iter | 16 | | 11 |
| Value | 736 | | 574 |
| *X\* = [1.00, 1.00], F(X\*) = 0.00* | | *X\* = [1.00, 1.00], F(X\*) = 0.00* | |
| *X0 = [-3, 3], Eps = 0.001, alpha = 10.* | | | |
| Iter | 21 | | 14 |
| Value | 966 | | 769 |

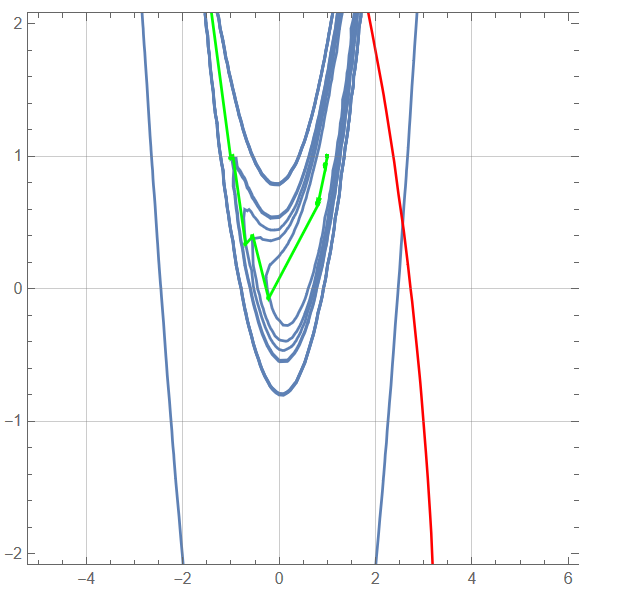


Рисунок 22. Метод внутренних штрафных функций. (-3, 3), eps = 0.001, alpha = 5.

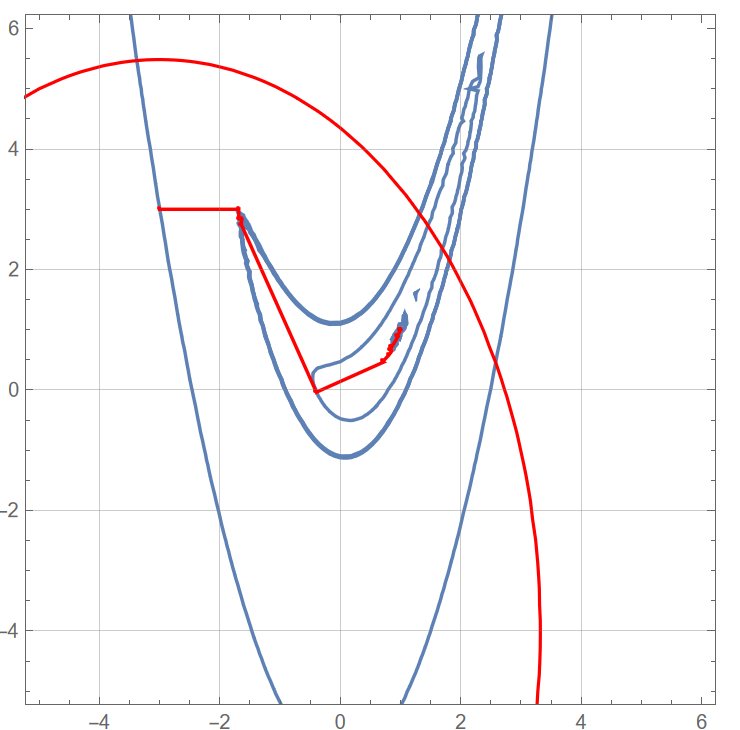


Рисунок 23. Метод внешних штрафных функций. (-3, 3), eps = 0.001, alpha = 5.

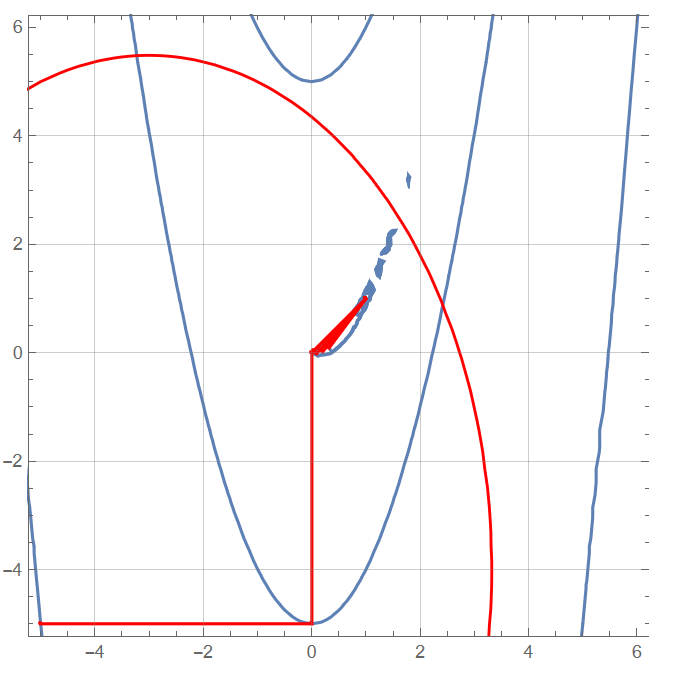
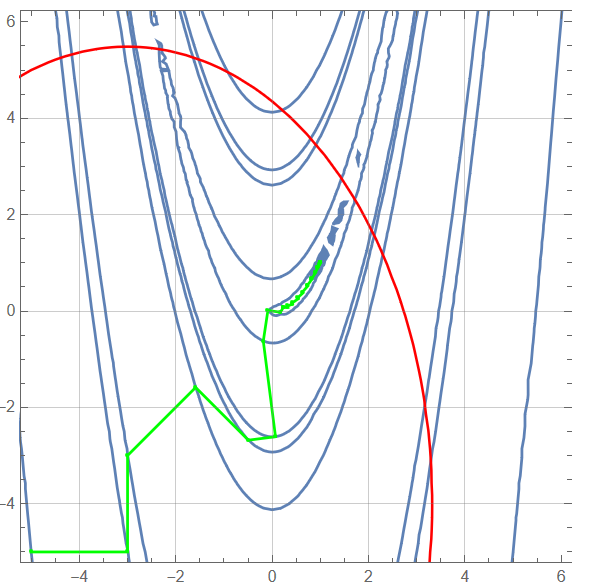


Рисунок 24. Метод внешних штрафных функций. (-4, -4), eps = 0.001.



1. Рисунок 25. Метод внутренних штрафных функций. (-4, -4), eps = 0.001.
   1. Вывод

Таким образом в лабораторной работе №8 мы рассмотрели методы минимизации функции, заданной на допустимом множестве. К этим методам относятся метод внутренних штрафных функций и внешних штрафных функций. Оба метода относятся к более общему методу – методу барьерных функций. Данные алгоритмы основаны на добавлении к основной функции функций штрафа. Принципиальная разница заключается в составлении этих самых штрафных функций. Для метода внешних функций они составлены таким образом, что сходимость решения не чувствительна к начальной точке, в отличии от метода внутренних штрафов, для которой принципиально, чтобы начальная точка была внутри допустимой области.

Зафиксировав коэффициент овражности, увеличивая точность, количество вычислений у метода внутренних штрафных функций увеличиваются в 2 раза.

Метод внутренних штрафных функций чувствителен к овражности функции Розенброка.

# Заключение

По итогу выполнения лабораторных работ по курсу «Методы оптимизации» мною были изучены и реализованы в среде компьютерной алгебры следующие методы оптимизации: методы одномерной минимизации, методы градиентного спуска, метод циклического покоординатного спуска, симплекс-методы и методы безусловной оптимизации.

Оптимизация проводилась для функций двух видов: квадратичная функция и функция Розенброка, которая имеет «овражную» структуру. Были выявлены плюсы и минусы различных методов. Например, для поиска минимума квадратичной функции метод Ньютона имеет наименьшее количество итераций, а именно всего одну итерацию. А для функции Розенброка эффективен метод Розенброка, который на каждой итерации строит новый ортогональный базис, благодаря чему находит оптимальное направление спуска.