**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет по лабораторной работе №4**

Авторы: Зюзько Роман, Иванов Дмитрий, Пак Руслан

Факультет: ФИТиП

Группы: M32341, М32351

Преподаватель:



Санкт-Петербург 2021

**Цель лабораторной работы**

Реализовать:

* алгоритм Ньютона (классически, с одномерным поиском, с направлением спуска)
* квазиньютоновский метод Давидона-Флетчера-Пауэлла
* квазиньютоновский метод Пауэлла
* метод Марквардта (двумя вариантами)

Исследовать:

* работу методов на двух функциях с заданным начальным приближением
* работу квазиньютоновских методов в сравнении с наилучшим методом Ньютона
* работу метода Марквардта на минимизации многомерной функции Розенброка (𝑛 = 100) в сравнении с наилучшим методом Ньютона

**Функции**

**1.**

2.

3.

**Вычислительные схемы методов**

***Методы Ньютона***

Метод Ньютона относится к классу итерационных методов. Если целевая функция

дважды дифференцируема в пространстве , то в процессе минимизации можно использовать Гессиан функции.

За счет использования Гессиана повышается скорость сходимости итерационной последовательности. На каждой итерации происходит квадратичная аппроксимация целевой функции в некоторой окрестности точки.

Для текущей точки в итерационном процессе происходит аппроксимация квадратичной функции, однако следующей точку релаксационной последовательности определяется как точка минимума аппроксимирующей квадратичной функции, описанной выше. Аппроксимация основана на формуле Тейлора. Пренебрегая остаточным членом в форме Пеано, запишем формулу Тейлора и обозначим за

Функция имеет одну точку в которой достигается минимум.

Условие минимума:

Тогда точка релаксационной последовательности определяется как:

- направление спуска.

Если положительно определенная, то принято называть ньютоновским направлением спуска.

В классическом варианте

***Классический***

Алгоритм минимизации:

1. Рассчитать градиент функции в точке и рассчитать Гессиан функции в точке

2. Решить систему линейных алгебраических уравнений:

3. Произвести пересчет

4. Минимум найден, если . Это эквивалентно . Иначе, повторить алгоритм с пункта 1.

Классический метод Ньютона не обладает свойством глобальной сходимости, так как если начальная точка достаточно далека от , то метод не сходится.

Имеет квадратичную локальную сходимость при условии того, что Гессиан функции удовлетворяет в окрестности условию Липшица. Таким образом, в малой окрестности метод будет сходится значительно быстрее.

***С одномерным спуском***

Точку минимума можно считать вспомогательным приближением.

- вспомогательная точка для построения релаксационной последовательности Тогда точку релаксационной последовательности можно определить как:

, где и это направление спуска. Алгоритм минимизации аналогичен классическому, однако определяется как:

Эффективность методы существенно зависит от того, является ли направлением спуска. Метод обладает свойством глобальной сходимости.

***С направлением спуска***

Данный метод позволяет избавится от ошибочного выбора направления возрастания функции. Ошибочный выбор связан с точками максимума функции и седловыми точками. В этих случаях необходимо проверять является ли направлением спуска. - направление спуска, то

Если , то не является направлением спуска. В этом случае следует использовать антиградиент. Тогда зададим следующим образом:

Для поиска оптимального можно использовать одномерный поиск. Остальные шаги аналогичны предыдущим методам. Метод обладает свойством глобальной сходимости.

***Квазиньютоновские методы***

Квазиньютоновские методы объединяют в себе достоинства метода Ньютона и метода наискорейшего спуска. Они не требуют обращений к Гессиану, однако при этом сохраняется высокая сходимость итерационной последовательности.

Общий вид релаксационной последовательности:

- направление спуска. Определяется как:

- антиградиент функции

- положительно определенная матрица порядка (n×n) специального вида для каждого квазиньютоновского метода

Вычисление матриц :

- матрица при достаточно больших k должна сходится к обратной матрице Гессе. x∗ точка минимума целевой функции. На первой итерации матрица является единичной, таким образом направление равняется антиградиенту. За счет того, что матрица на завершающей стадии поиска близка к мы можем ожидать высокую сходимость итерационной последовательности, присущую методу Ньютона.

чаще всего выбирается как исчерпывающий спуск в направлении , однако возможно задавать или применять дробление шага.

Основная идея квазиньютоновских методов заключается в том, что специальная матрица должна быть достаточно близка к и вычисление происходит за счет аппроксимации . Выбор удачной аппроксимации позволяет существенно сократить объем вычислений по сравнению с обращением к Гессиану, таким образом упрощается процедура построения направления спуска . Различные квазиньютоновские методы определяет выбор "поправочной"матрицы , используемой для вычисления .

Квазиньютоновские методы обладают глобальной сходимостью только в случае применения рестартов. Имеют сверхлинейную скорость сходимости, однако если Гессиан функции удовлетворяет условию Липшица, то в результате будет достигнута квадратичная скорость.

***Метод Давидона-Флэтчра-Пауэлла***

Данный метод обладает рядом свойств, опишем их подробнее:

1. сохраняет положительную определенность: если , то и , при условии

2. Если симметричная, то и симметричная

3. При минимизации квадратичной функции с положительно определенной матрицей A метод сводится к методу сопряженных направлений. Следовательно, метод дает точное решение не более чем за n итераций.

4. Матрицы вычисленные по методу связаны равенством:

При k = n, n векторов являются собственными для симметричной матрицы , а собственные значения равны единице. Таким образом, оказывается обратной к Гессиану квадратичной функции.

5. Если целевая функция не квадратичная, то метод не позволяет найти минимум за конечное число итераций. Для уменьшения ошибки и вероятности появления линейно зависимых направлений спуска, принято каждые n итераций принимать (единичной матрице)

6. Если целевая функция является квадратичной с положительно определенным Гессианом, то метод после n итераций дает:

***Метод Пауэлла***

Метод Пауэлла Аналогичен методу ДФП, однако имеет другое рекуррентное соотношение для вычисления матриц

Исследование методов Ньютона на различных функциях

Проведем исследование на двух функциях: *sin*(*x*) и 8 *x*2 + 4 *x y* + 5 *y*2, а также на различных начальных приближениях.

Точность: *eps* = 10−5

Для поиска ньютоновского направления используется метод Гаусса.

## Классический метод Ньютона

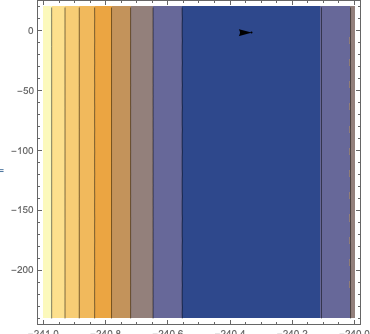
### Функция sin(x)

Начальное приближение: *x* = 1. Количество итераций: 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 1.0000000 | 1.6420840 | 0.6420840 | 0.9974601 |
| 1 | 1.0000000 | 1.5706701 | -0.0714138 | 1.0000000 |
| 2 | 1.0000000 | 1.5707913 | 0.0001212 | 1.0000000 |
| 3 | 1.0000000 | 1.5707913 | 0.0000000 | 1.0000000 |

Как можно заметить, метод нашел точку максимума, вместо минимума.

Это связано с тем, что для *sin*(*x*) Гессиан отрицательно определен.



Начальное приближение:

Количество итераций: 2

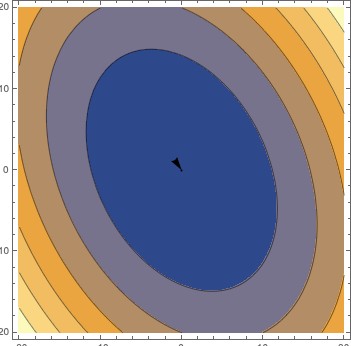
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 1.0000000 | -1.5708103 | -0.0300140 | -1.0000000 |
| 1 | 1.0000000 | -1.5708013 | 0.0000090 | -1.0000000 |

При данном начальном приближении метод находит истинный минимум, Гессиан определен положительно. Начальное приближение находится в ма- ленькой окрестности минимума.

Начальное приближение:

Количество итераций: 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 1.0000000 | 30.2006548 | 33.3122475 | -0.9374640 |
| 1 | 1.0000000 | 29.8293463 | -0.3713085 | -0.9998754 |
| 2 | 1.0000000 | 29.8451265 | 0.0157802 | -1.0000000 |
| 3 | 1.0000000 | 29.8451252 | -0.0000013 | -1.0000000 |



***Функция:* 8** · ***x*2 + 4** · ***x*** · ***y* + 5** · ***y*2**

Начальное приближение: (*x, y*) = (1*.*0*,* 2*.*0)

Количество итераций: 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0000000 | -0.0000110 | 0.0000067 | -1.0000110 | -1.9999933 | 0.0000000 |
| 1 | 1.0000000 | -0.0000042 | -0.0000033 | 0.0000068 | -0.0000100 | 0.0000000 |
| 2 | 1.0000000 | -0.0000042 | -0.0000033 | 0.0000000 | -0.0000000 | 0.0000000 |

Начальное приближение: (*x, y*) = (2*.*0*,* 1*.*0)

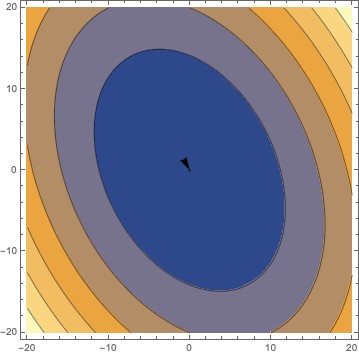
Количество итераций: 3

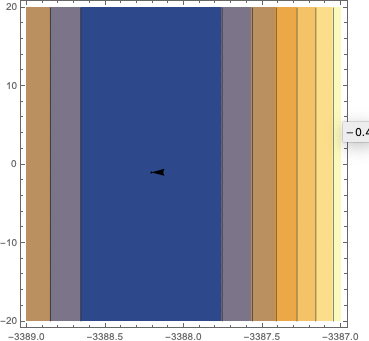
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0000000 | -0.0000060 | 0.0000224 | -2.0000060 | -0.9999776 | 0.0000000 |
| 1 | 1.0000000 | -0.0000042 | -0.0000033 | 0.0000018 | -0.0000257 | 0.0000000 |
| 2 | 1.0000000 | -0.0000042 | -0.0000033 | 0.0000000 | -0.0000000 | 0.0000000 |

Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Количество итераций: 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0000000 | -0.0000042 | -0.0000033 | -0.0000042 | -0.0000033 | 0.0000000 |





## Метод Ньютона с одномерной оптимизацией методом золотого сечения

## Функция sin(x)

Начальное приближение: *x* = 1. Количество итераций: 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | -346.5002993 | -221.4822838 | 0.6420840 | -1.0000000 |
| 1 | -0.5184478 | -221.4822821 | -0.0000033 | -1.0000000 |

Начальное приближение:

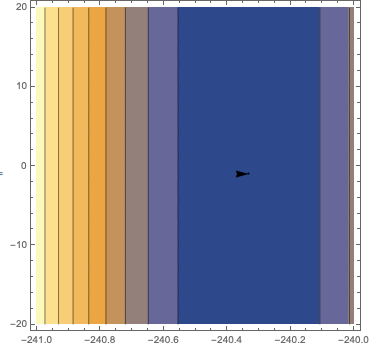
Количество итераций: 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 210.3414161 | -7.8539817 | -0.0300140 | -1.0000000 |
| 1 | -0.0091814 | -7.8539816 | -0.0000049 | -1.0000000 |

Начальное приближение:

Количество итераций: 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | -708.2025967 | -23594.9317404 | 33.3122475 | -1.0000000 |
| 1 | 1.0452909 | -23594.9316248 | 0.0001106 | -1.0000000 |
| 2 | 0.0041431 | -23594.9316248 | -0.0000050 | -1.0000000 |



***Функция:* 8** · ***x*2 + 4** · ***x*** · ***y* + 5** · ***y*2**

Начальное приближение: (*x, y*) = (1*.*0*,* 2*.*0)

Количество итераций: 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999990 | -0.0000100 | 0.0000088 | -1.0000110 | -1.9999933 | 0.0000000 |
| 1 | 0.9014830 | -0.0000047 | -0.0000021 | 0.0000058 | -0.0000121 | 0.0000000 |
| 2 | -0.0000087 | -0.0000047 | -0.0000021 | 0.0000006 | -0.0000012 | 0.0000000 |

Начальное приближение: (*x, y*) = (2*.*0*,* 1*.*0)

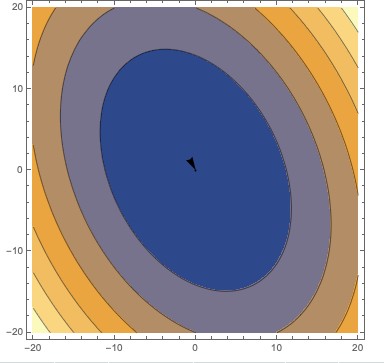
Количество итераций: 3

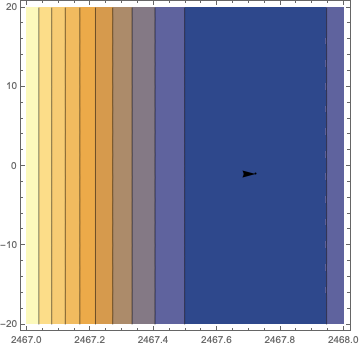
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999990 | -0.0000039 | 0.0000234 | -2.0000060 | -0.9999776 | 0.0000000 |
| 1 | 0.8118490 | -0.0000041 | 0.0000017 | -0.0000003 | -0.0000268 | 0.0000000 |
| 2 | -0.0000141 | -0.0000041 | 0.0000017 | -0.0000000 | -0.0000050 | 0.0000000 |

Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Количество итераций: 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | -0.0000000 | 0.0000000 | 0.0000000 | -0.0000042 | -0.0000033 | 0.0000000 |





## Метод ньютона с направлением спуска

В качестве одномерной оптимизации также выбран метод золотого сечения.

Функция sin*(*x*)*

Начальное приближение: *x* = 1. Количество итераций: 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 528.0679989 | -284.3141357 | -0.5402981 | -1.0000000 |
| 1 | -0.1219434 | -284.3141352 | -0.0000044 | -1.0000000 |

Начальное приближение:

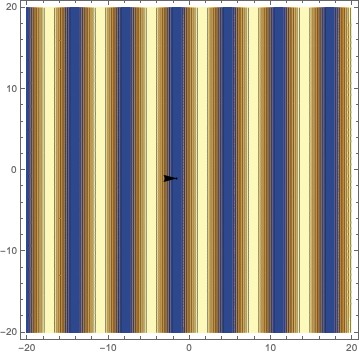
Количество итераций: 2.

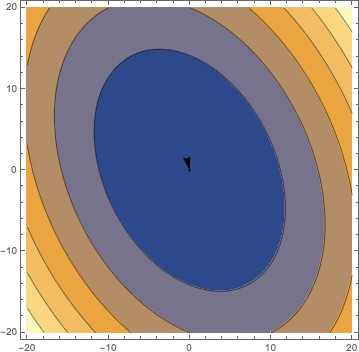
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | 210.3414161 | -7.8539817 | -0.0300140 | -1.0000000 |
| 1 | -0.0091814 | -7.8539816 | -0.0000049 | -1.0000000 |

Начальное приближение:

Количество итераций: 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk* | *pk* | *f* (*xk*) |
| 0 | -708.2025967 | -23594.9317404 | 33.3122475 | -1.0000000 |
| 1 | 1.0452909 | -23594.9316248 | 0.0001106 | -1.0000000 |
| 2 | 0.0041431 | -23594.9316248 | -0.0000050 | -1.0000000 |





***Функция:* 8** · ***x*2 + 4** · ***x*** · ***y* + 5** · ***y*2**

Начальное приближение: (*x, y*) = (1*.*0*,* 2*.*0)

Количество итераций: 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999990 | -0.0000100 | 0.0000088 | -1.0000110 | -1.9999933 | 0.0000000 |
| 1 | 0.9014830 | -0.0000047 | -0.0000021 | 0.0000058 | -0.0000121 | 0.0000000 |
| 2 | -0.0000087 | -0.0000047 | -0.0000021 | 0.0000006 | -0.0000012 | 0.0000000 |

Начальное приближение: (*x, y*) = (2*.*0*,* 1*.*0)

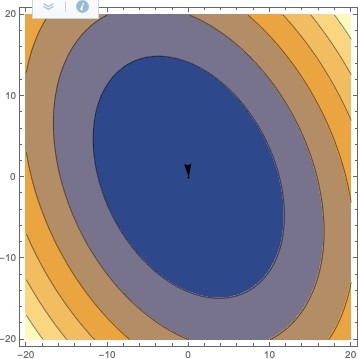
Количество итераций: 3

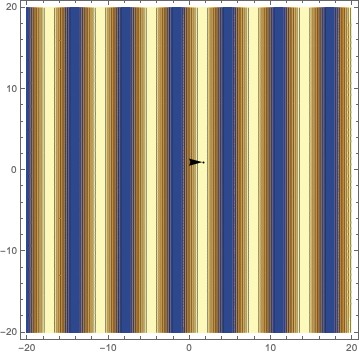
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999990 | -0.0000039 | 0.0000234 | -2.0000060 | -0.9999776 | 0.0000000 |
| 1 | 0.8118490 | -0.0000041 | 0.0000017 | -0.0000003 | -0.0000268 | 0.0000000 |
| 2 | -0.0000141 | -0.0000041 | 0.0000017 | -0.0000000 | -0.0000050 | 0.0000000 |

Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Количество итераций: 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | -0.0000000 | 0.0000000 | 0.0000000 | -0.0000042 | -0.0000033 | 0.0000000 |





# Исследование работы методов на функциях

Функции:

*f* (*x*) = *x*12 + *x*22 − 1*.*2 · *x*1 · *x*2 где *x*0 = (4*,* 1)*T*

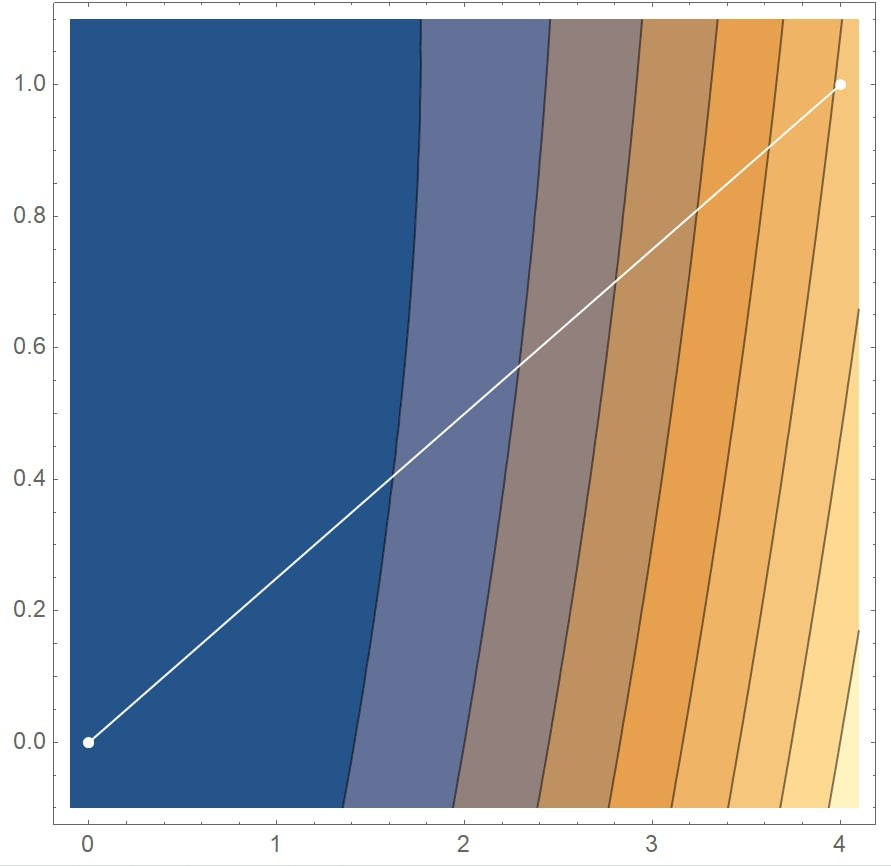
*f* (*x*) = 100 · (*x*2 − *x*12)2 + (1 − *x*1)2 где *x*0 = (−1*.*2*,* 1)*T*

Для поиска ньютоновского направления используется метод Гаусса.

## Классический метод Ньютона

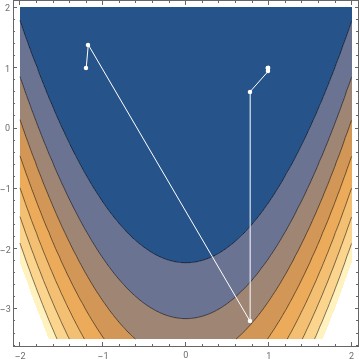
Функция: *f* (*x*) = *x*12 + *x*22 − 1*.*2 · *x*1 · *x*2 где *x*0 = (4*,* 1)*T*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0000000 | -0.0000205 | -0.0000707 | -4.0000205 | -1.0000707 | 0.0000000 |
| 1 | 1.0000000 | -0.0000125 | -0.0000125 | 0.0000080 | 0.0000582 | 0.0000000 |
| 2 | 1.0000000 | -0.0000125 | -0.0000125 | -0.0000000 | -0.0000000 | 0.0000000 |



Функция: *f* (*x*) = 100 · (*x*2 − *x*12)2 + (1 − *x*1)2 где *x*0 = (−1*.*2*,* 1)*T*

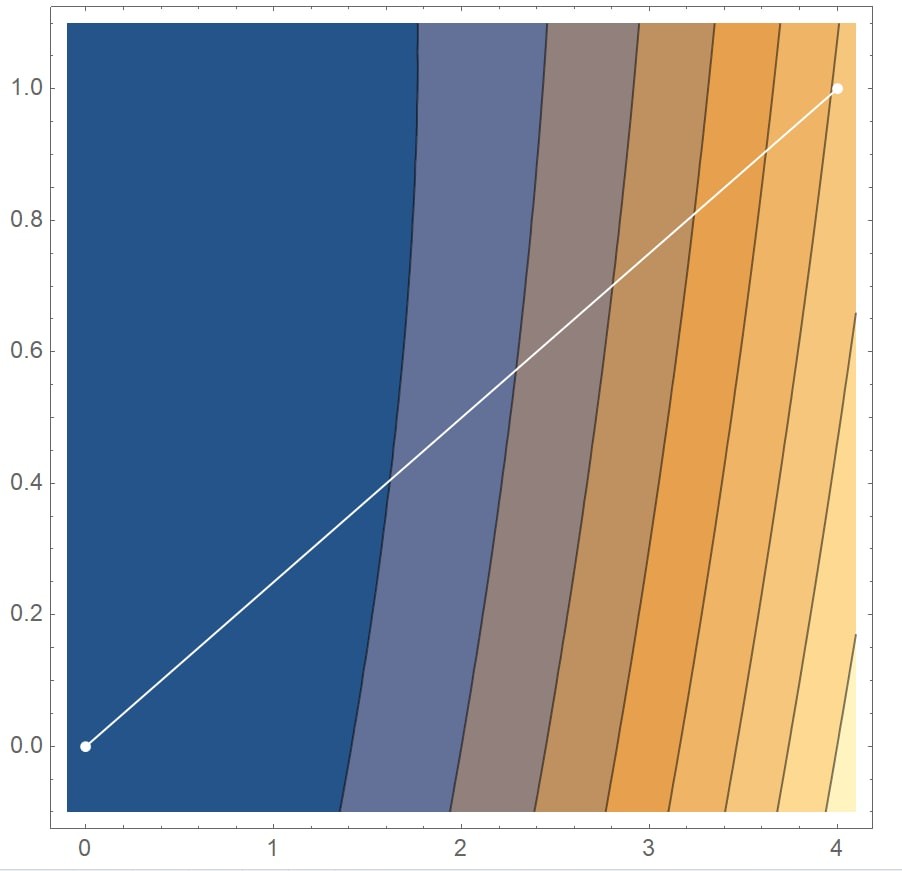
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0000000 | -1.1752973 | 1.3807088 | 0.0247027 | 0.3807088 | 4.7319562 |
| 1 | 1.0000000 | 0.7766295 | -3.2068503 | 1.9519268 | -4.5875591 | 1451.6626929 |
| 2 | 1.0000000 | 0.7769160 | 0.6035470 | 0.0002865 | 3.8103973 | 0.0497667 |
| 3 | 1.0000000 | 0.9944222 | 0.9415638 | 0.2175063 | 0.3380168 | 0.2238718 |
| 4 | 1.0000000 | 0.9946706 | 0.9893646 | 0.0002484 | 0.0478008 | 0.0000284 |
| 5 | 1.0000000 | 0.9970007 | 0.9939999 | 0.0023301 | 0.0046354 | 0.0000090 |
| 6 | 1.0000000 | 0.9970099 | 0.9940237 | 0.0000092 | 0.0000237 | 0.0000089 |
| 7 | 1.0000000 | 0.9970099 | 0.9940238 | 0.0000000 | 0.0000001 | 0.0000089 |



## Метод Ньютона с одномерной оптимизацией методом золотого сечения

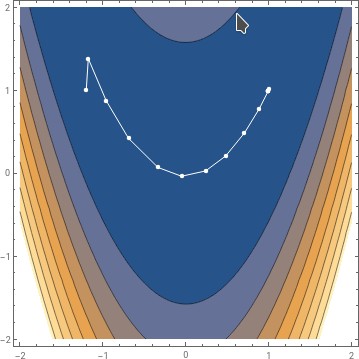
Функция: *f* (*x*) = *x*12 + *x*22 − 1*.*2 · *x*1 · *x*2 где *x*0 = (4*,* 1)*T*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999985 | -0.0000144 | -0.0000692 | -4.0000205 | -1.0000707 | 0.0000000 |
| 1 | 1.0949930 | -0.0000123 | -0.0000071 | 0.0000019 | 0.0000567 | 0.0000000 |
| 2 | 0.0000422 | -0.0000123 | -0.0000071 | -0.0000002 | -0.0000054 | 0.0000000 |



Функция: *f* (*x*) = 100 · (*x*2 − *x*12)2 + (1 − *x*1)2 где *x*0 = (−1*.*2*,* 1)*T*

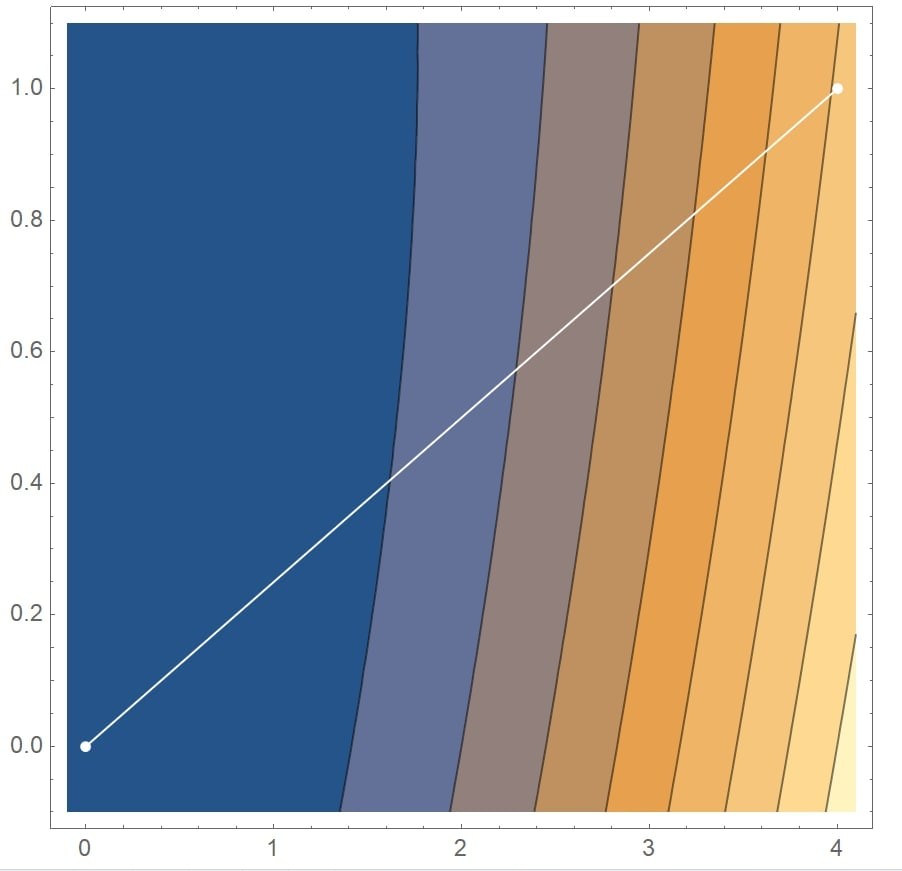
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *k* | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0041967 | -1.1751936 | 1.3823065 | 0.0247027 | 0.3807088 | 4.7316178 |
| 1 | 0.0739520 | -0.9595228 | 0.8753076 | 2.9163608 | -6.8557792 | 4.0456309 |
| 2 | 1.3861153 | -0.6898562 | 0.4206978 | 0.1945484 | -0.3279740 | 3.1603598 |
| 3 | 2.5344459 | -0.3340534 | 0.0696941 | 0.1403868 | -0.1384933 | 1.9552388 |
| 4 | 1.9895399 | -0.0509768 | -0.0360816 | 0.1422824 | -0.0531659 | 1.2541679 |
| 5 | 2.4084221 | 0.2387900 | 0.0275248 | 0.1203139 | 0.0264100 | 0.6664414 |
| 6 | 2.2073311 | 0.4821507 | 0.2088475 | 0.1102511 | 0.0821457 | 0.3239667 |
| 7 | 2.4988044 | 0.7076356 | 0.4852989 | 0.0902371 | 0.1106335 | 0.1093450 |
| 8 | 2.5026798 | 0.8852453 | 0.7753187 | 0.0709678 | 0.1158837 | 0.0201252 |
| 9 | 2.8402177 | 1.0044694 | 1.0100798 | 0.0419771 | 0.0826560 | 0.0001457 |
| 10 | 0.7603725 | 0.9971968 | 0.9946133 | -0.0095645 | -0.0203407 | 0.0000123 |
| 11 | 0.8540904 | 0.9970307 | 0.9940968 | -0.0001945 | -0.0006048 | 0.0000089 |
| 12 | 0.2229746 | 0.9970260 | 0.9940805 | -0.0000208 | -0.0000731 | 0.0000089 |
| 13 | 0.0016309 | 0.9970260 | 0.9940804 | -0.0000161 | -0.0000567 | 0.0000089 |



## Метод Ньютона с направлением спуска

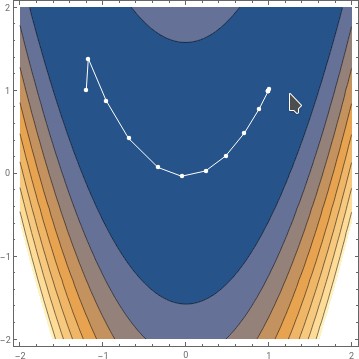
В качестве одномерной оптимизации также выбран метод золотого сечения. Функция: *f* (*x*) = *x*12 + *x*22 − 1*.*2 · *x*1 · *x*2 где *x*0 = (4*,* 1)*T*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итерация | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 0.9999985 | -0.0000144 | -0.0000692 | -4.0000205 | -1.0000707 | 0.0000000 |
| 1 | 1.0949930 | -0.0000123 | -0.0000071 | 0.0000019 | 0.0000567 | 0.0000000 |
| 2 | 0.0000422 | -0.0000123 | -0.0000071 | -0.0000002 | -0.0000054 | 0.0000000 |



Функция: *f* (*x*) = 100 · (*x*2 − *x*12)2 + (1 − *x*1)2 где *x*0 = (−1*.*2*,* 1)*T*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *k* | *αk* | *xk*1 | *xk*2 | *pk*1 | *pk*2 | *f* (*xk*) |
| 0 | 1.0041967 | -1.1751936 | 1.3823065 | 0.0247027 | 0.3807088 | 4.7316178 |
| 1 | 0.0739520 | -0.9595228 | 0.8753076 | 2.9163608 | -6.8557792 | 4.0456309 |
| 2 | 1.3861153 | -0.6898562 | 0.4206978 | 0.1945484 | -0.3279740 | 3.1603598 |
| 3 | 2.5344459 | -0.3340534 | 0.0696941 | 0.1403868 | -0.1384933 | 1.9552388 |
| 4 | 1.9895399 | -0.0509768 | -0.0360816 | 0.1422824 | -0.0531659 | 1.2541679 |
| 5 | 2.4084221 | 0.2387900 | 0.0275248 | 0.1203139 | 0.0264100 | 0.6664414 |
| 6 | 2.2073311 | 0.4821507 | 0.2088475 | 0.1102511 | 0.0821457 | 0.3239667 |
| 7 | 2.4988044 | 0.7076356 | 0.4852989 | 0.0902371 | 0.1106335 | 0.1093450 |
| 8 | 2.5026798 | 0.8852453 | 0.7753187 | 0.0709678 | 0.1158837 | 0.0201252 |
| 9 | 2.8402177 | 1.0044694 | 1.0100798 | 0.0419771 | 0.0826560 | 0.0001457 |
| 10 | 0.7603725 | 0.9971968 | 0.9946133 | -0.0095645 | -0.0203407 | 0.0000123 |
| 11 | 0.8540904 | 0.9970307 | 0.9940968 | -0.0001945 | -0.0006048 | 0.0000089 |
| 12 | 0.2229746 | 0.9970260 | 0.9940805 | -0.0000208 | -0.0000731 | 0.0000089 |
| 13 | 0.0016309 | 0.9970260 | 0.9940804 | -0.0000161 | -0.0000567 | 0.0000089 |



## Сравнение количества итераций

Функция: *f* (*x*) = *x*12 + *x*22 − 1*.*2 · *x*1 · *x*2 где *x*0 = (4*,* 1)*T*

|  |  |
| --- | --- |
| Классический | 3 |
| С одномерной оптимизацией | 3 |
| С направлением спуска | 3 |
| Наискорейший спуск | 23 |

Функция: *f* (*x*) = 100 · (*x*2 − *x*12)2 + (1 − *x*1)2 где *x*0 = (−1*.*2*,* 1)*T*

|  |  |
| --- | --- |
| Классический | 8 |
| С одномерной оптимизацией | 13 |
| С направлением спуска | 13 |
| Наискорейший спуск | 89 |

## Выводы по исследованиям на заданных функциях

является квадратичной, поэтому все методы сходятся с достаточно высокой скоростью.

В ходе исследования классический метод Ньютона показал наилучшие результаты.

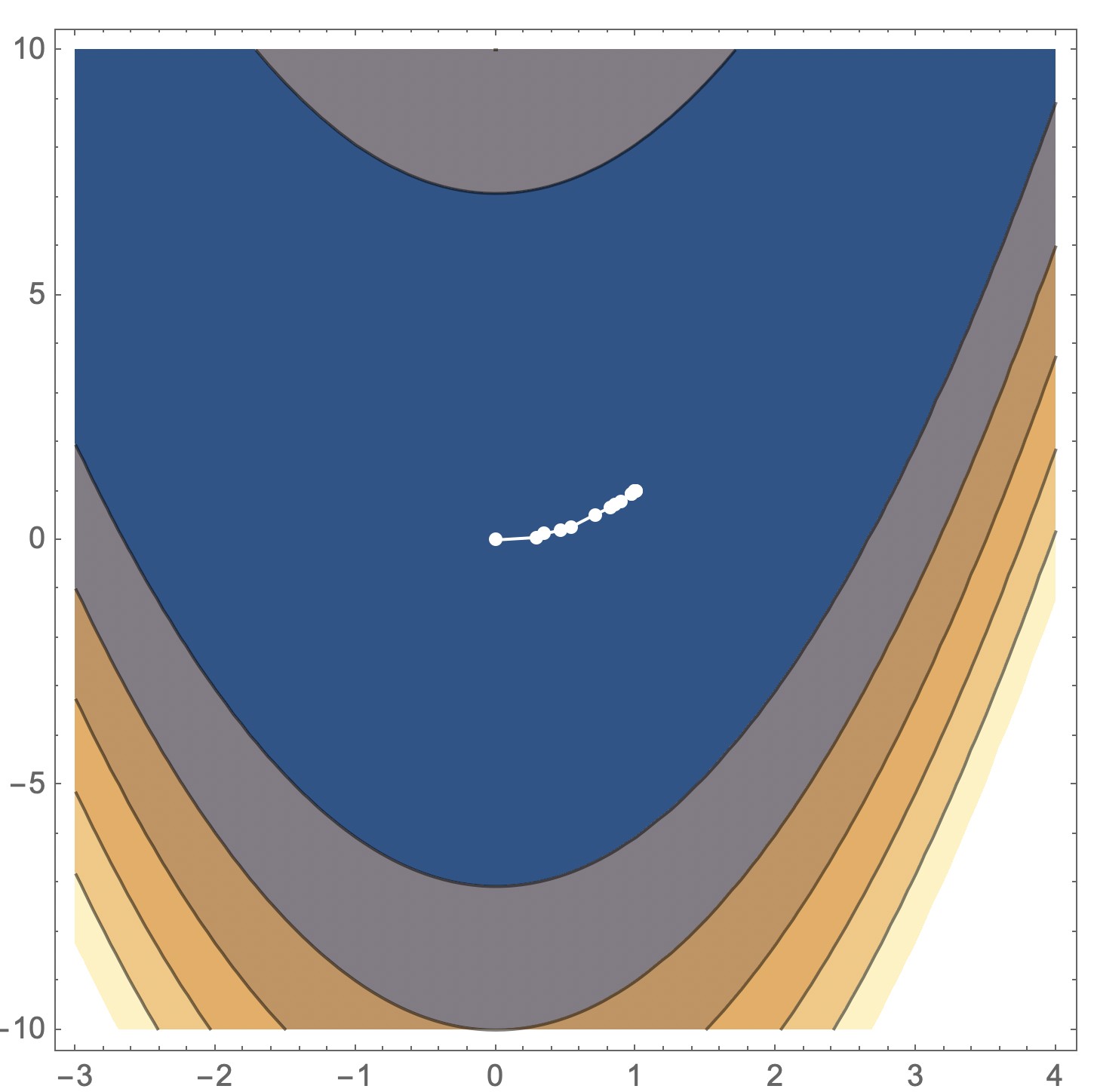
# Квазиньютоновские методы

Проведем сравнение методов Давидона-Флетчера-Пауэлла, Пауэла и клас- сического метода Ньютона на четырех функциях с различным начальным приближением:

* 1. *f* (*x*) = 100 Метод Давидона-Флетчера-Пауэлла

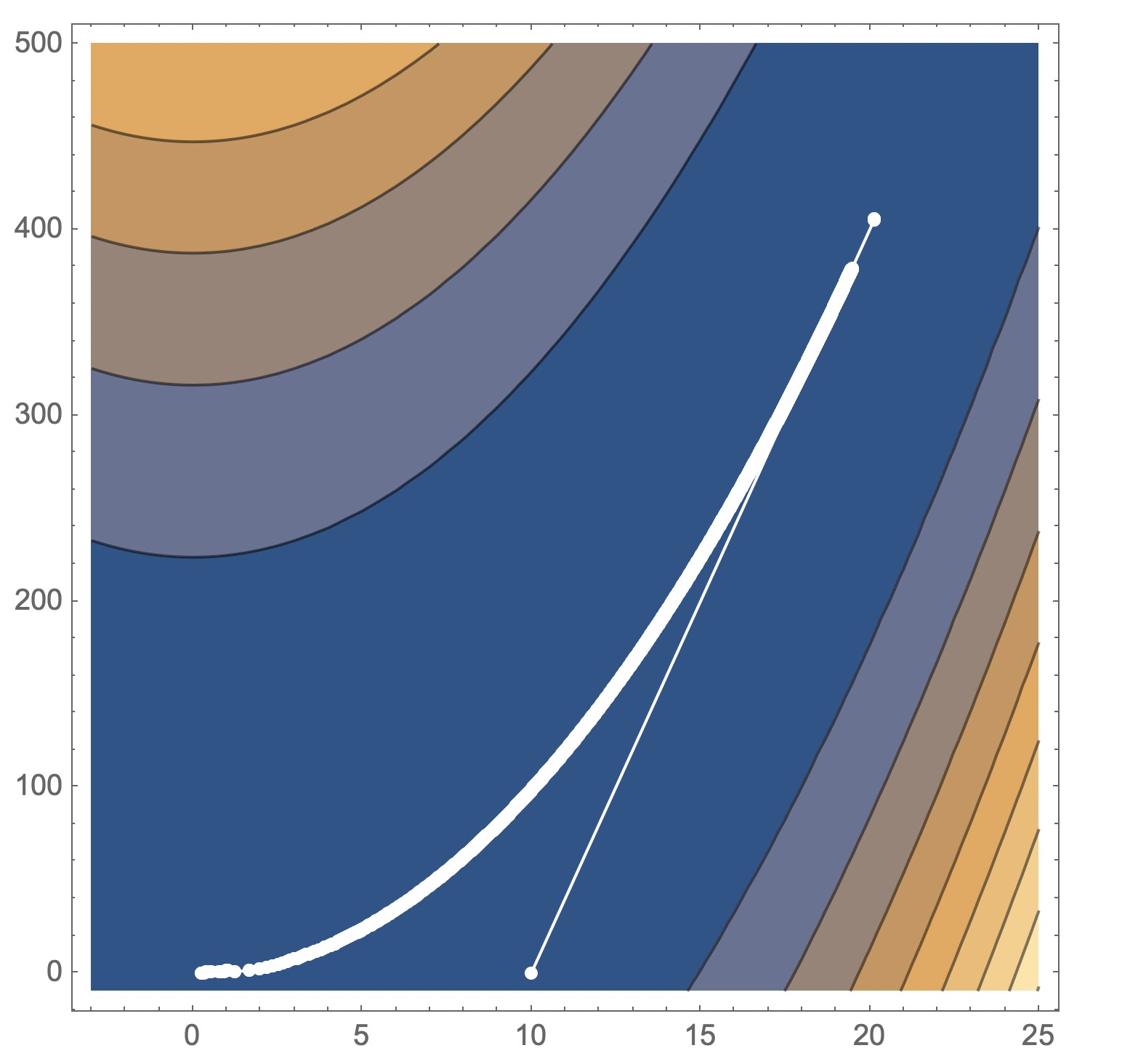
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Границы одномерного поиска: -10 до 15.



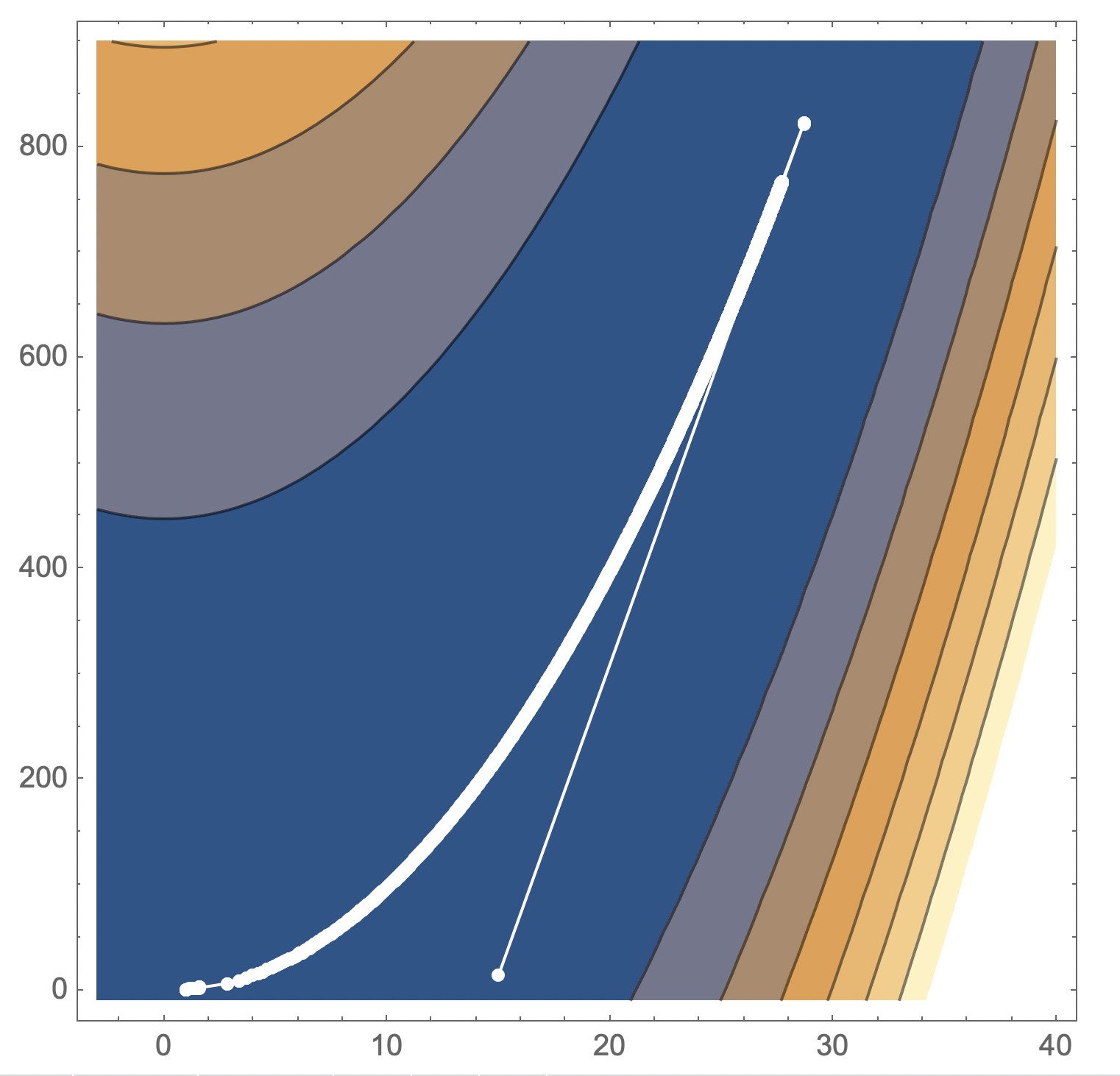
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

Границы одномерного поиска: -10 до 15.

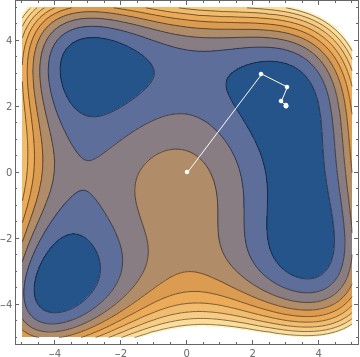
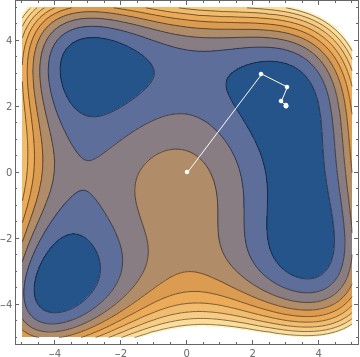


Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

Границы одномерного поиска: -10 до 15.



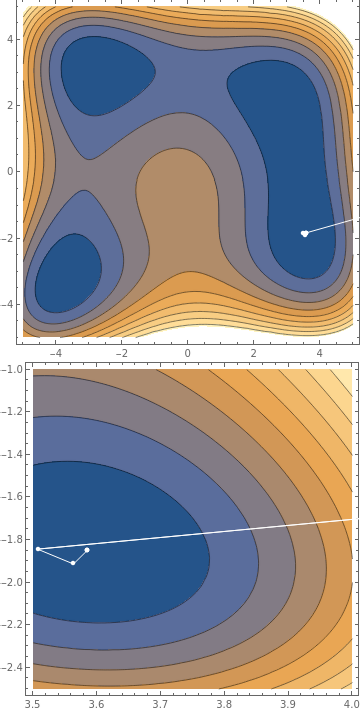
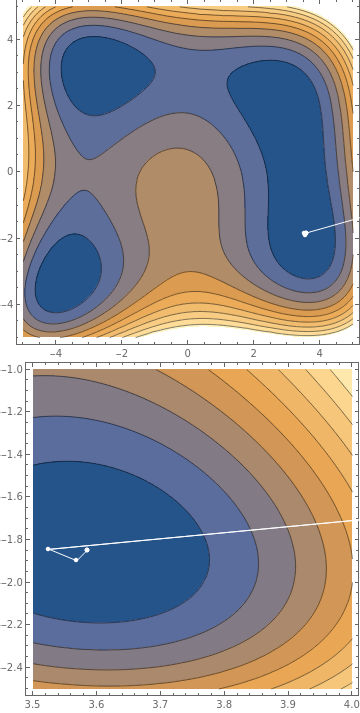
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)



На левой картинке используются границы одномерного поиска от −10 до

10, а на правой от −1000 до 1000.

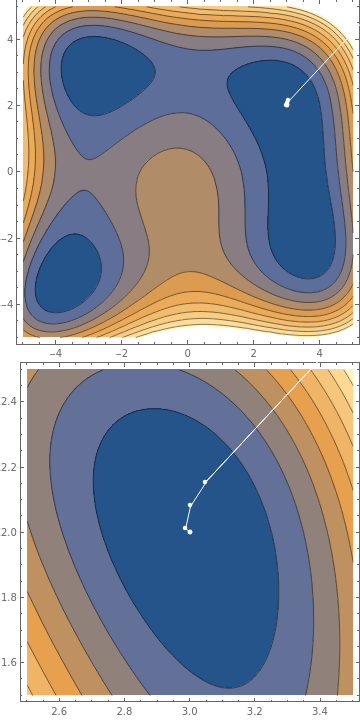
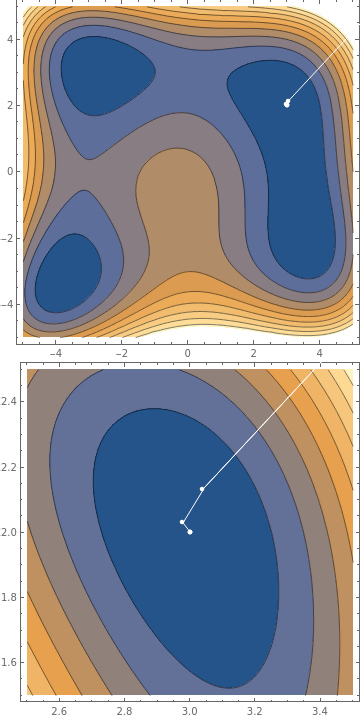
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

На левой картинке используются границы одномерного поиска от −10 до

10, а на правой от −1000 до 1000.

Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

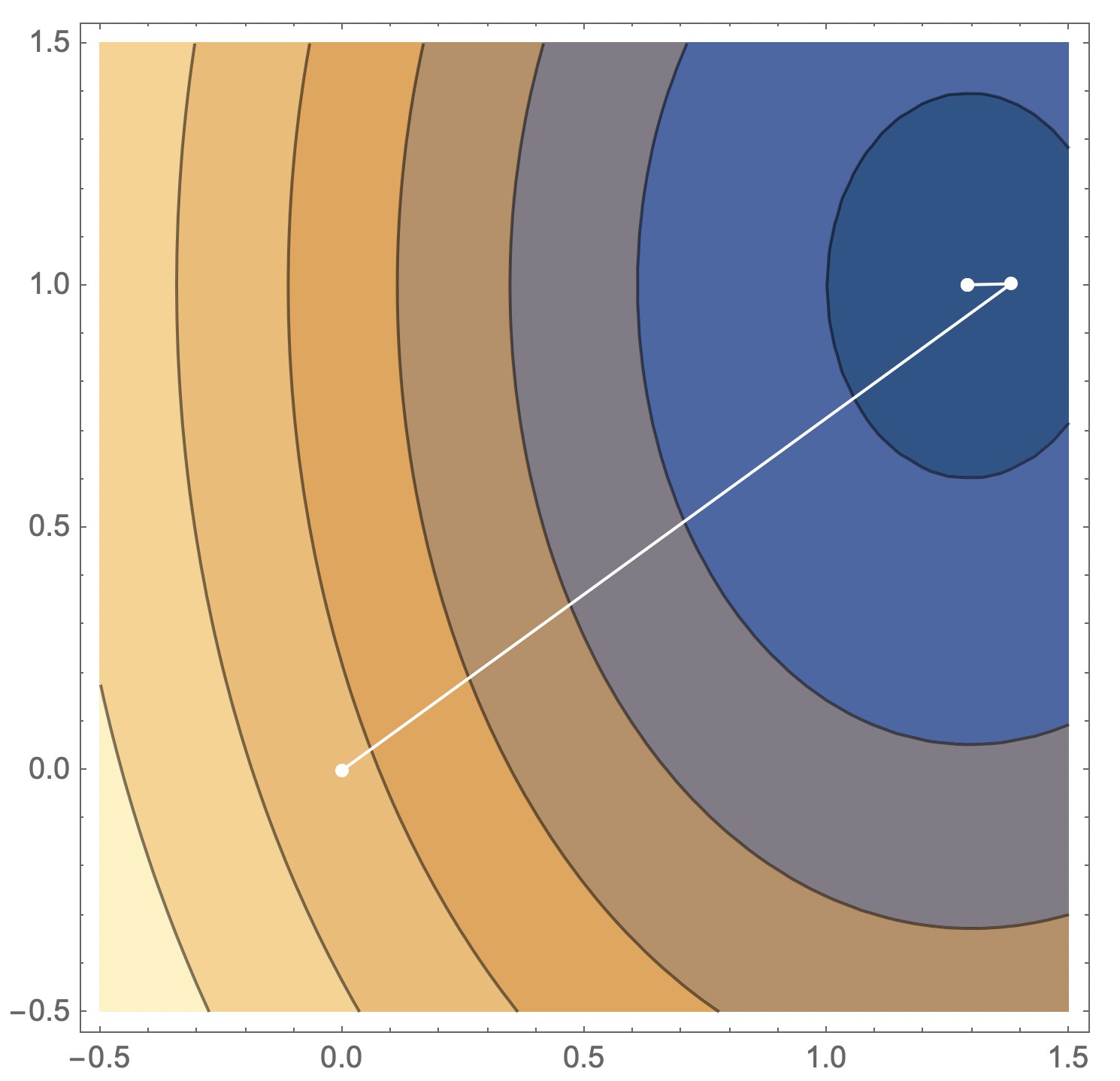
 

На левой картинке используются границы одномерного поиска от −10 до

10, а на правой от −1000 до 1000.

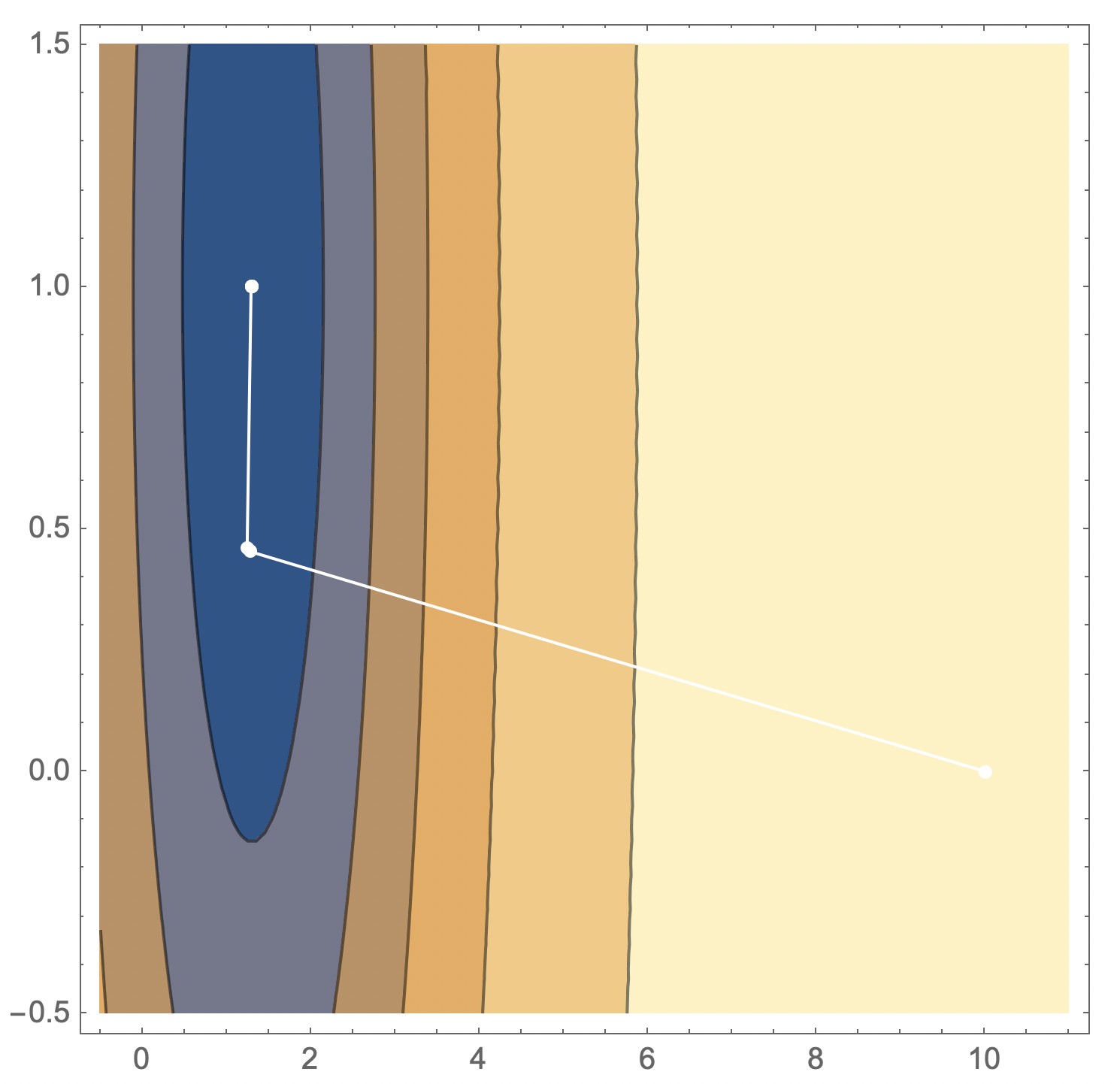
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Границы одномерного поиска: -10 до 15.



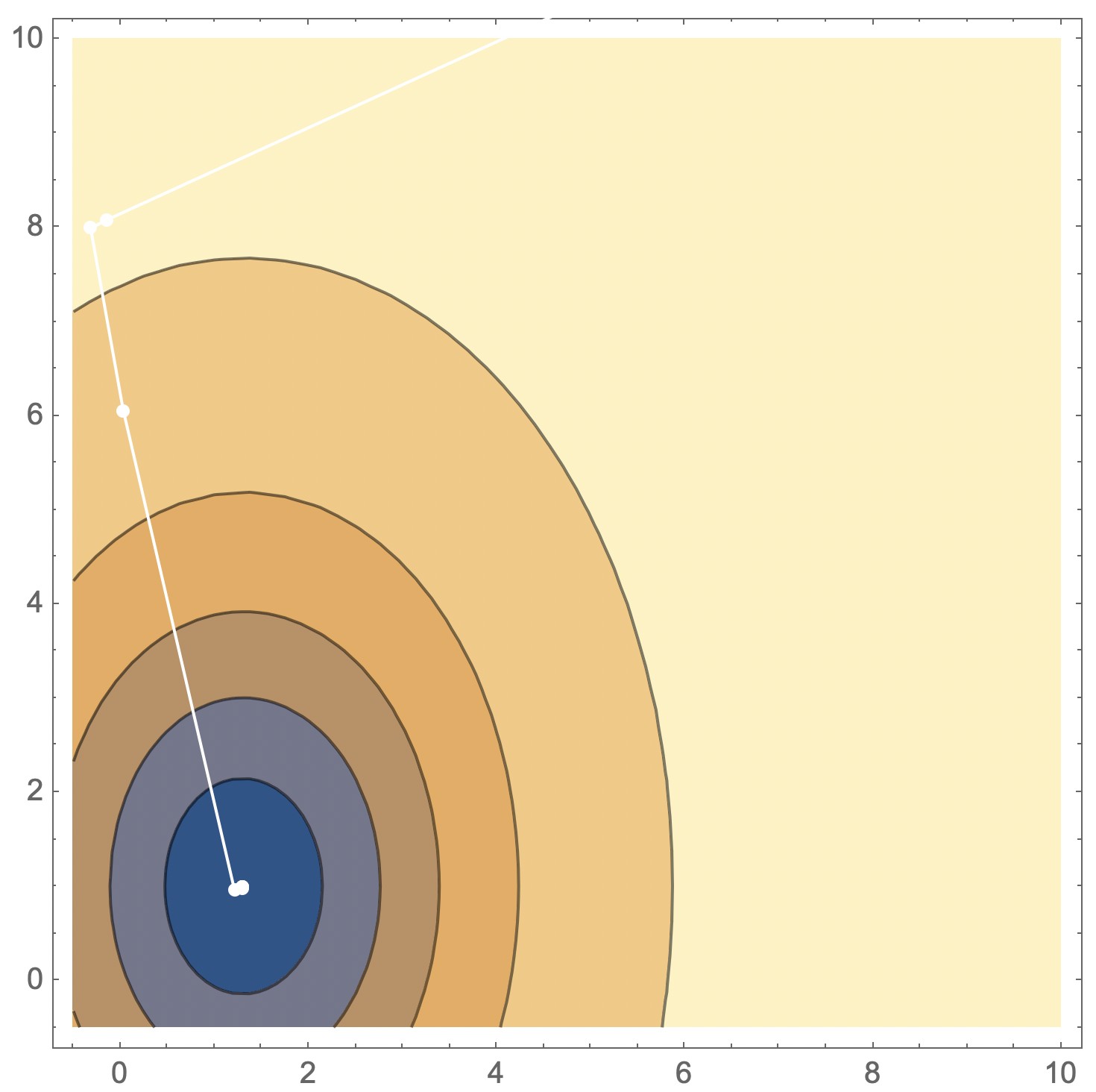
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

Границы одномерного поиска: -10 до 15.



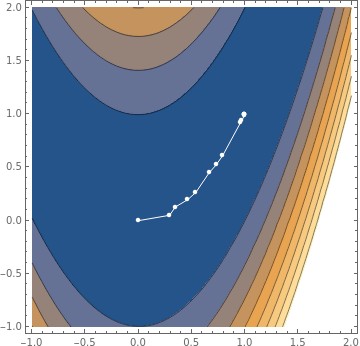
Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

Границы одномерного поиска: -10 до 15.

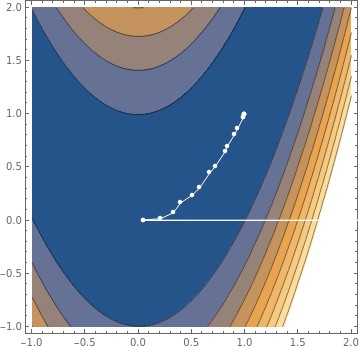


## Метод Пауэлла

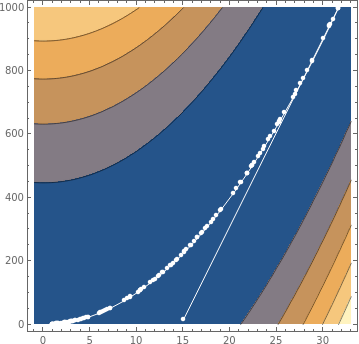
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)



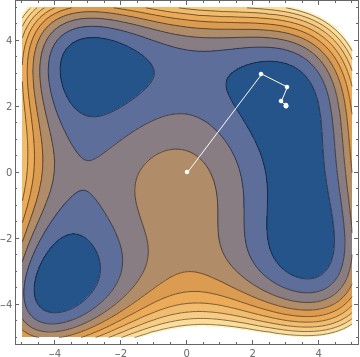
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)



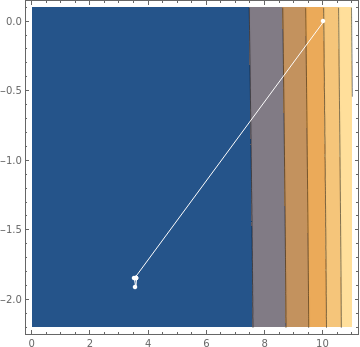
Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)



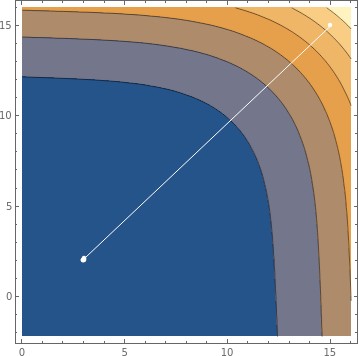
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

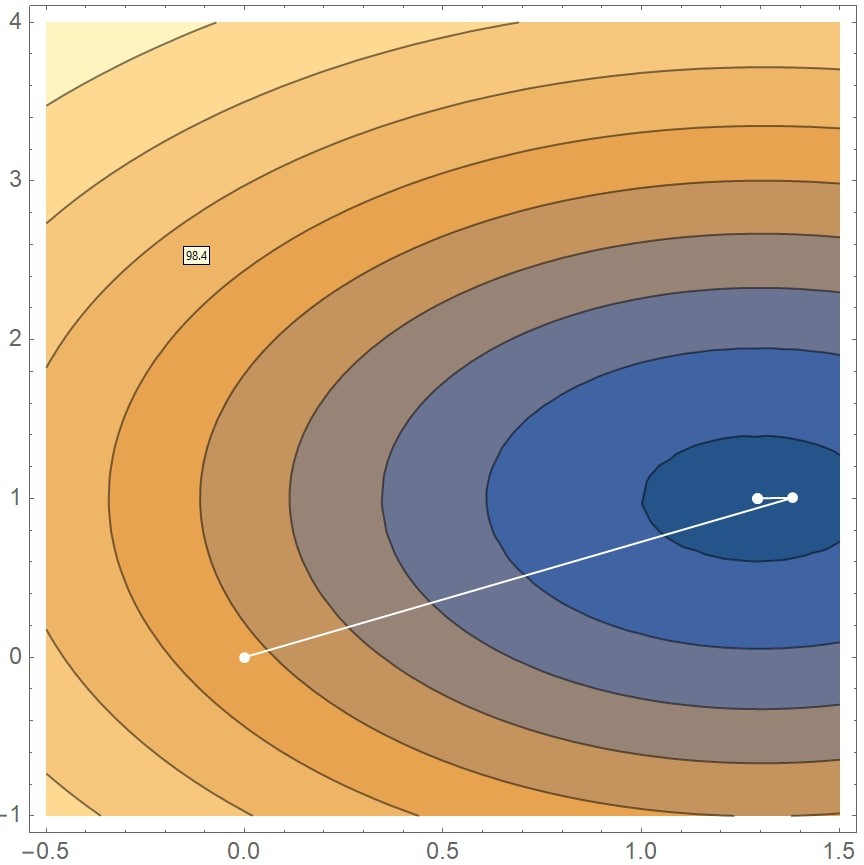


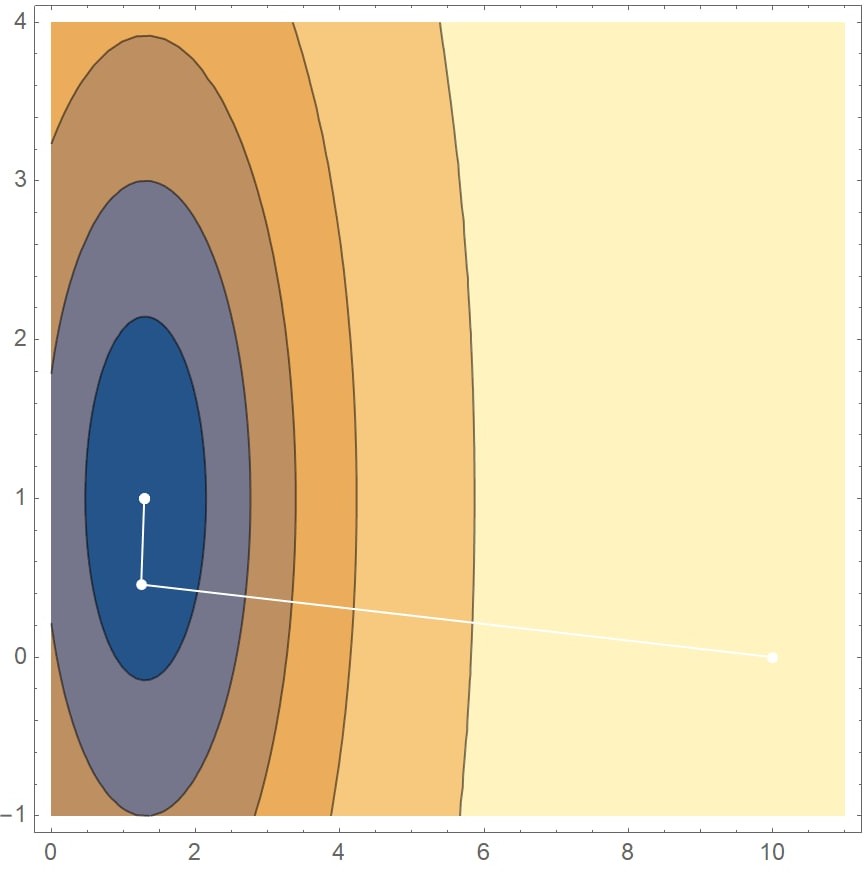
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

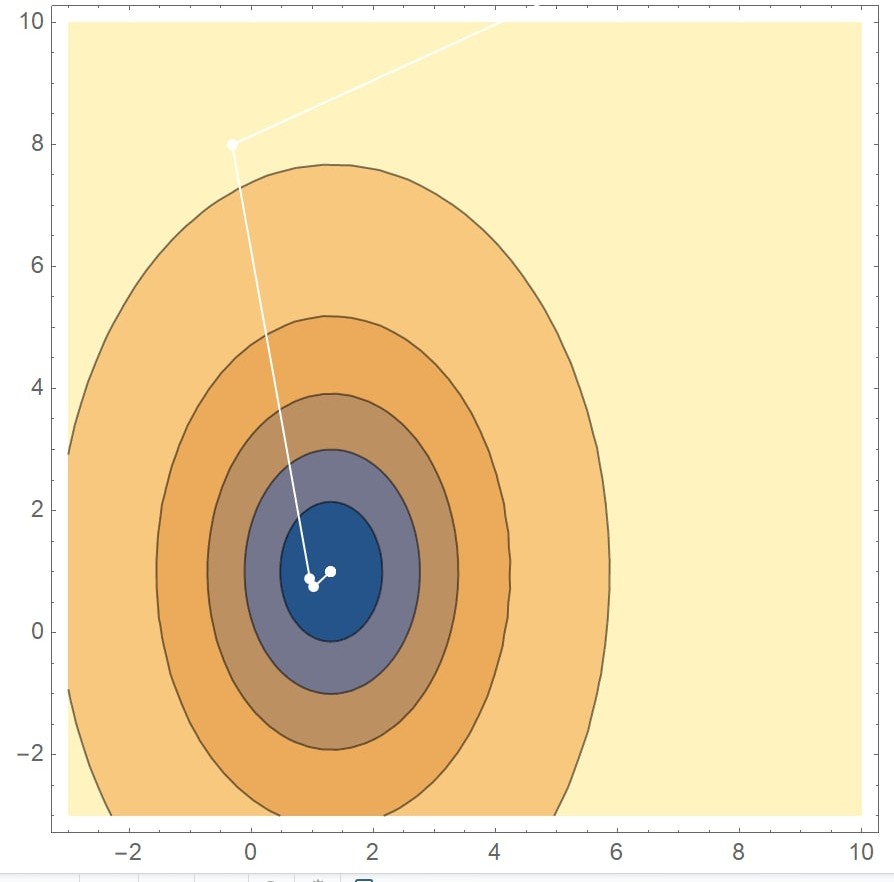


Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)



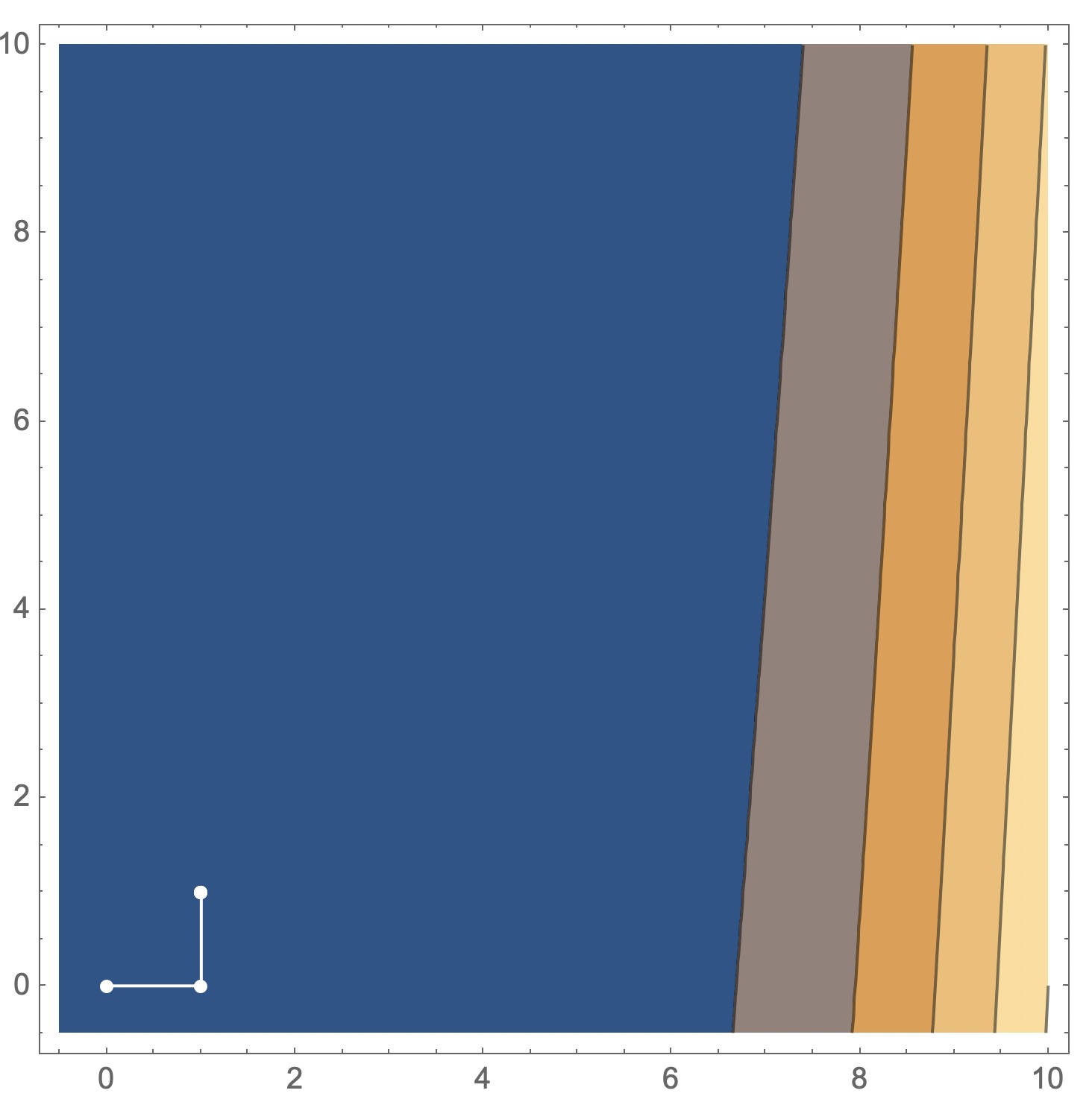
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

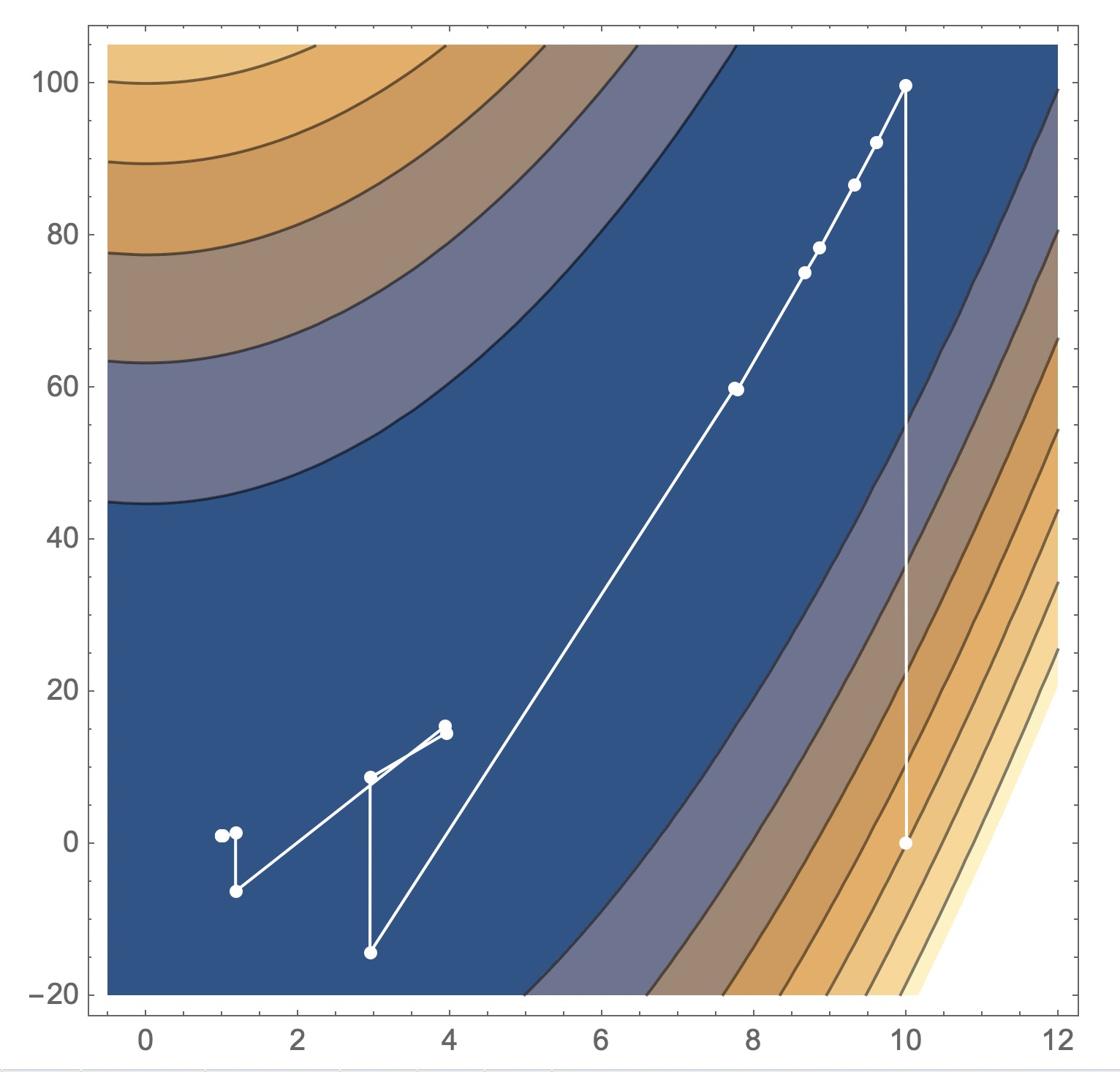
Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

## Классический Ньютон

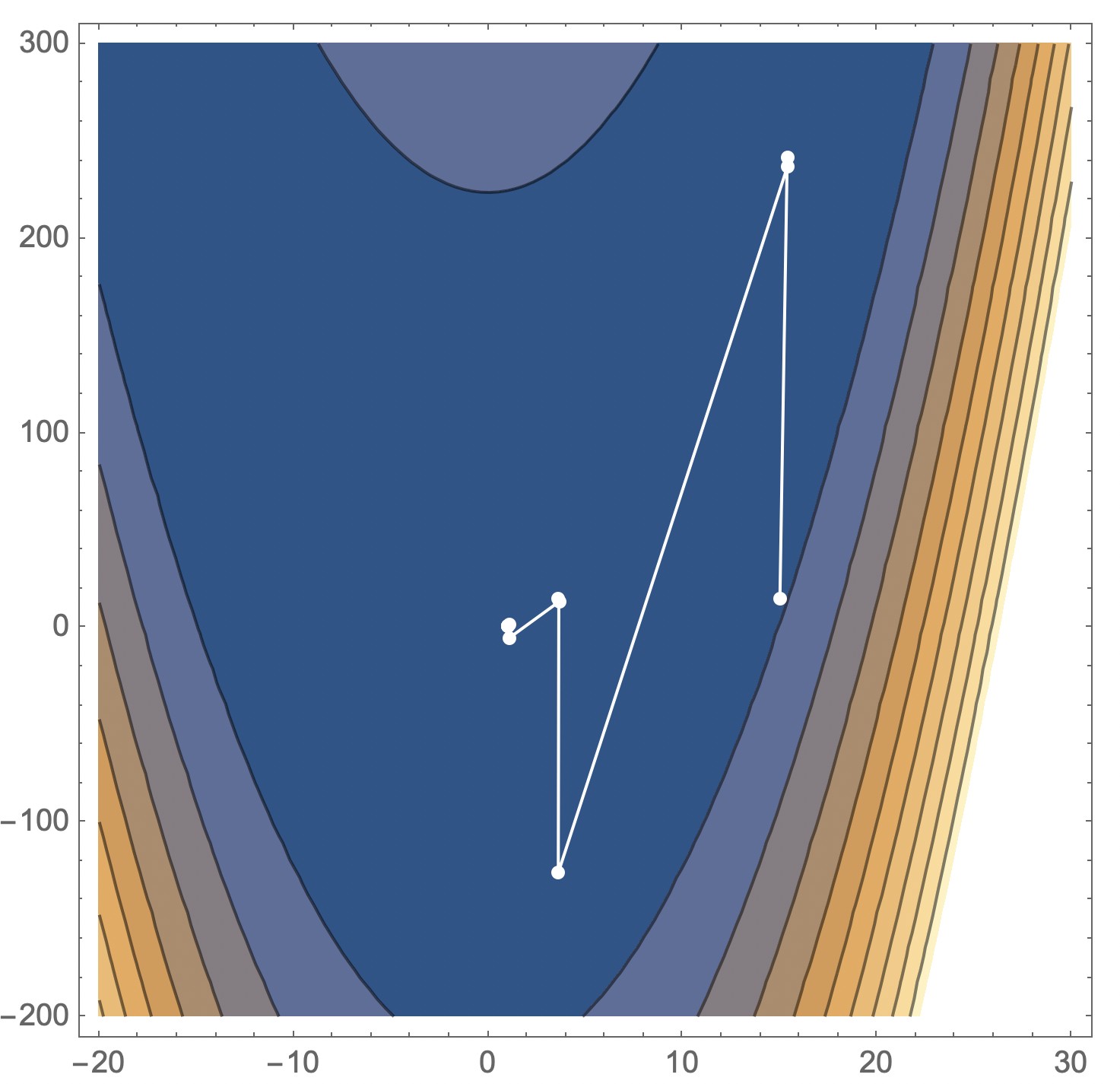
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)



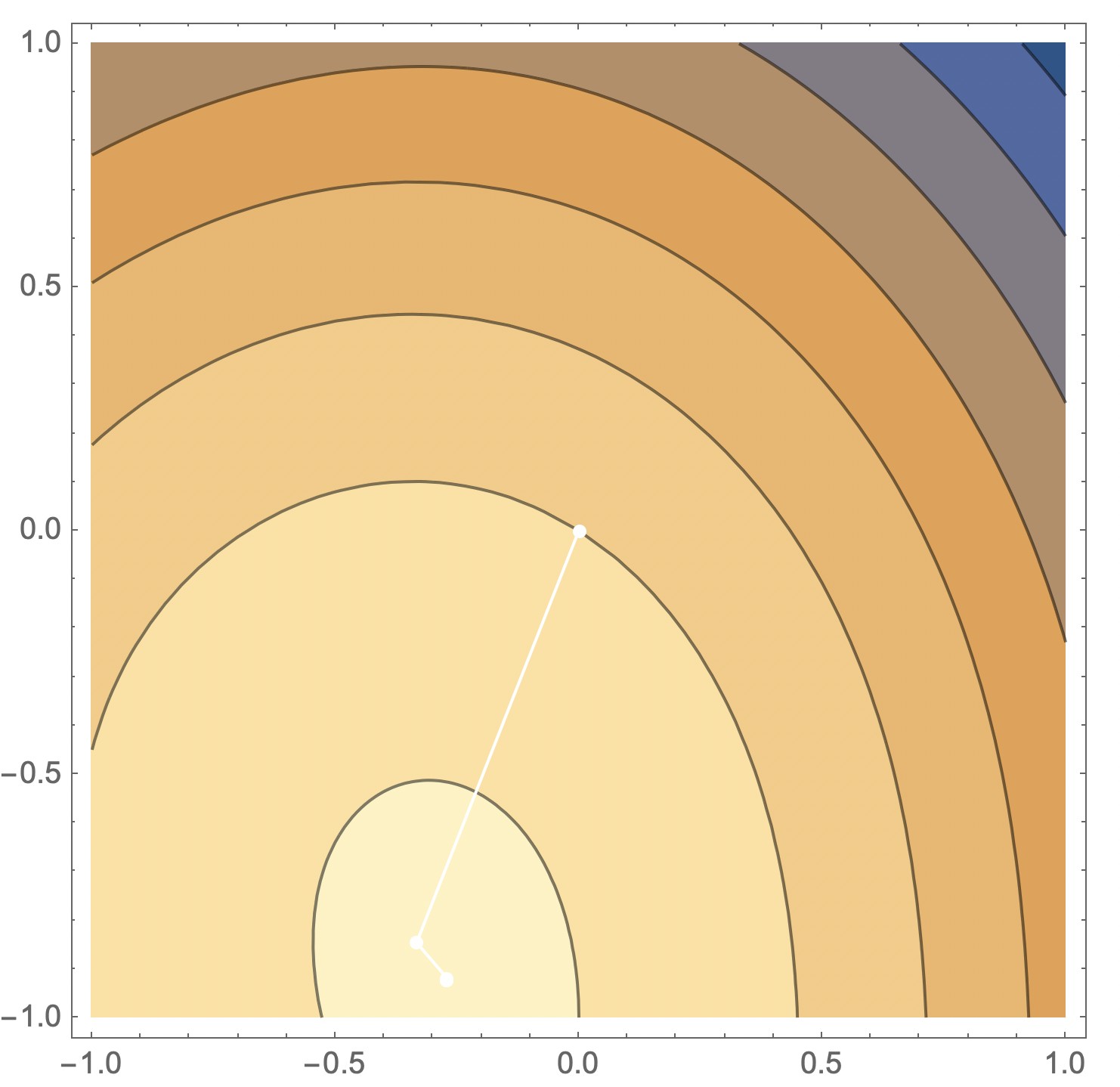
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)



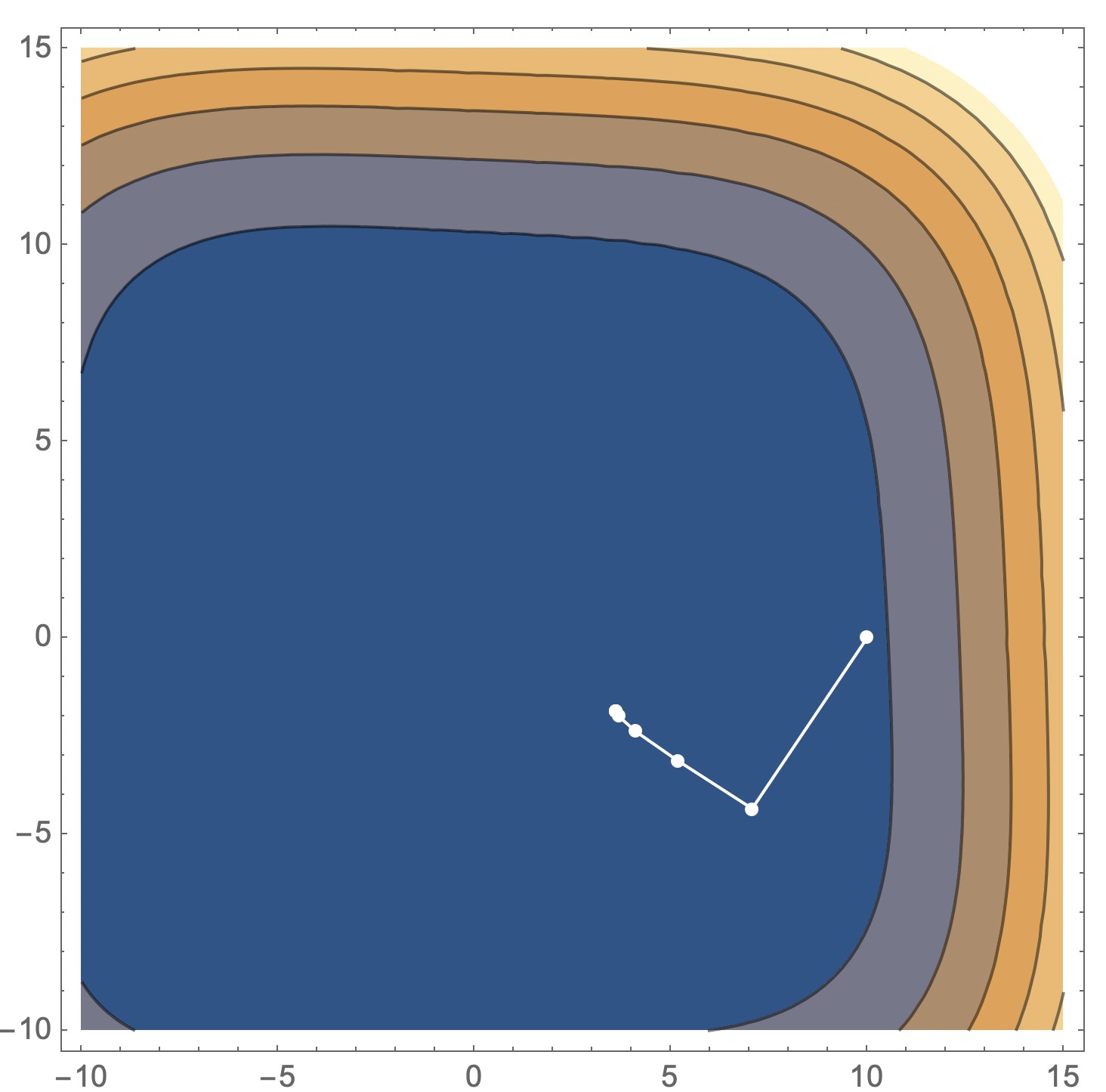
Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

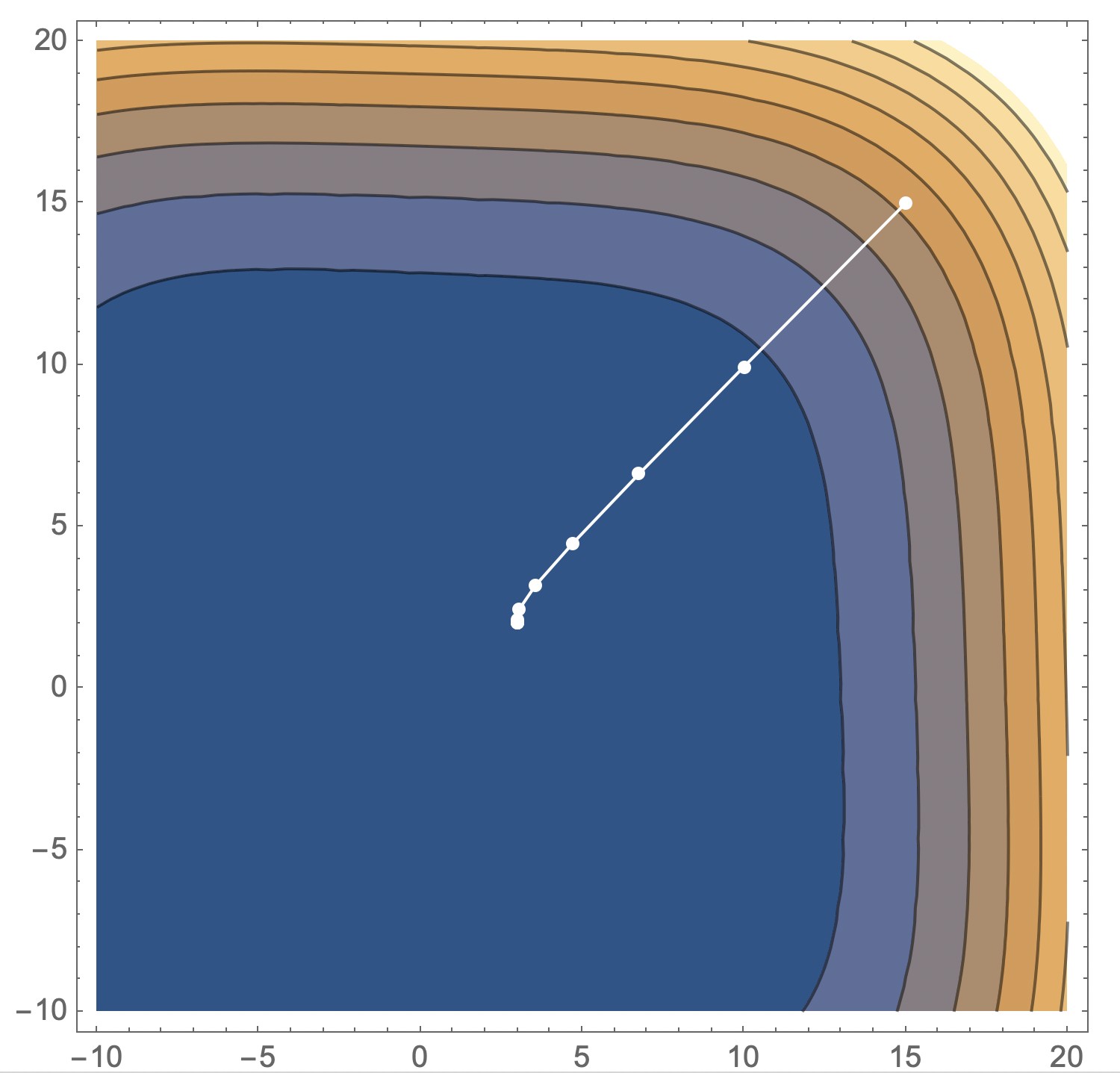


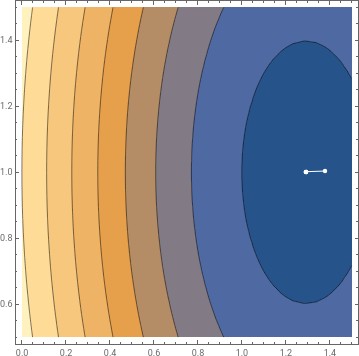
Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0). В данном случае метод нашел максимум

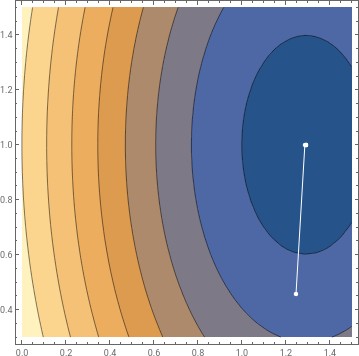


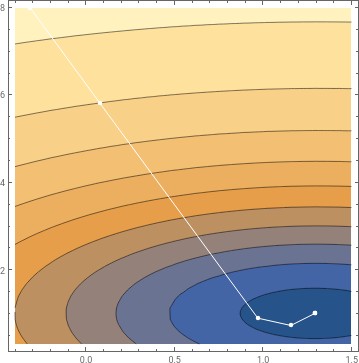
Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)



Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

Начальное приближение: (*x, y*) = (0*.*0*,* 0*.*0)

Начальное приближение: (*x, y*) = (10*.*0*,* 0*.*0)

Начальное приближение: (*x, y*) = (15*.*0*,* 15*.*0)

Ниже приведены таблицы итераций для каждого из исследуемых выше ме- тодов. Возможности построения траектории для функции в четырехмерном пространстве в ходе исследования не было.

ДФП

|  |  |
| --- | --- |
| Начальное приближение | Количество итераций |
| (0, 0, 0, 0) | 0 |
| (0, 10, 0, 10) | 83 |
| (15, 15, 15, 15) | 21 |

Пауэлл

|  |  |
| --- | --- |
| Начальное приближение | Количество итераций |
| (0, 0, 0, 0) | 0 |
| (0, 10, 0, 10) | 25 |
| (15, 15, 15, 15) | 15 |

Классический Ньютон

|  |  |
| --- | --- |
| Начальное приближение | Количество итераций |
| (0, 0, 0, 0) | 13 |
| (0, 10, 0, 10) | 22 |
| (15, 15, 15, 15) | 29 |

## Сравнение методов

В таблице приведены результаты запусков методов на начальных точках

(0*,* 0), (10*,* 0) и (15*,* 15). Знак вопроса означает, что метод не сошелся.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | ДФП | Пауэлл | Ньютон |
| *f*1 | 14 / 3005 / 2596 | 15 / 76 / 91 | 5 / 19 / 11 |
| *f*2 | 8 / 6 / 5 | 8 / 6 / 5 | 5 / 8 / 10 |
| *f*3 | 0 / 83 / 21 | 0 / 25 / 15 | ? / 171194 / 67550 |
| *f*4 | 4 / 6 / 8 | 4 / 6 / 7 | 13 / 22 / 29 |

Методы ДФП и Пауэлла имеют одинаковую скорость сходимости. Результаты сильно зависят от выбора начального приближения, даже если точки приближения находятся на небольшом расстоянии.

Метод Ньютона выигрывает по скорости сходимости у ДФП и Пауэлла, однако во второй функции метод Ньютона сошелся к точке максимума, а не точке минимума.

Метод Ньютона показал плохую сходимость на функции *f*4.

***Выводы***

В ходе лабораторной работы были реализованы методы Ньютона и его различные модификации, а также квазиньютоновские методы. Исследование было проведено как на квадратичных функциях, так и на не квадратичных. Было исследовано влияние выбора начального приближения на точность и скорость сходимости функции.

Методы Ньютона различаются по выбору направления и построению релаксационной последовательности, а также показывают различные и неоднозначные результаты на исследуемых функциях.

Классический метод Ньютона показывает себя лучше остальных методов, если начальное приближение находится в маленькой окрестности искомой точки минимума. Однако, если функция не является квадратичной, то результат непредсказуем.

Метод Ньютона с направлением спуска позволяет находить минимум и для не квадратичных функций, однако, может показывать медленную скорость сходимости.

Квазиньтоновские методы показали значительно лучшие результаты при наличии рестартов. Это обусловлено тем, что при наличии рестарстов они обладают глобальность сходимостью. Однако, Гессиан некоторых функций удовлетворял условию Липшица и на таких функциях скорость сходимости снижалась до квадратичной.

**[Ссылка на Github проекта](https://github.com/RuslanPark/ITMO-optimization-methods-course/tree/main/lab4)**