Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Лабораторная работа № 2

Выполнили студенты:
Иванов А.С. M32001
Круглов Г.Н. M32001
Писарева Ю.И. M32051

Проверил:
Москаленко М.А.

Санкт-Петербург
2022

Постановка задачи

Пусть дана функций $f(x)$, ограниченная снизу на множестве R^n и имеющая непрерывные частные производные во всех его точках.

Требуется найти локальный минимум функции на множестве допустимых решений $X = R^n$, т.е. найти такую точку $x^* \in R^n$, что $f(x^*) = \min\limits_{x \in R^n} f(x)$

Описание методов

Метод градиентного спуска

Стратегия метода заключается в построении последовательности точек $\{x^k\}, k = 0, 1...$ таких, что $f(x^{k+1}) < f(x^k), k = 0, 1...$ Точки последовательности $\{x^k\}$ вычисляются по правилу:

$$x^{k+1} = x^k - t_k \nabla f(x^k), k = 0, 1...$$

где точка x^0 задаётся пользователем, $\nabla f(x^k)-$ градиент функции $f(x)$, вычисленный в точке x^k. Выбор величины шага зависит от конкретной вариации метода и будет рассмотрен дальше. Построение последовательности x^k заканчивается в точке {x^k}, для которой $||\nabla f(x^k)|| < \varepsilon_1$, где ε_1 - заданное малое положительное число, или $k \geq M$, где M - предельное число итераций, или при одновременном выполнении двух неравенстр $||x^{k+1} - x^k|| < \varepsilon_2, |f(x^{k+1}) - f(x^k)| < \varepsilon_2$, где ε_2 - малое положительное число.

Метод градиентного спуска с постоянным шагом

Величина шага t_k задаётся пользователем и остаётся постоянной до тех пор, пока функция убывает в точках последовательности, что контролируется путём проверки выполнения условия $f(x^{k+1}) - f(x^k) < 0$. При невыполнении условия величина шага уменьшается на 2 и заново вычилсяется точка x^{k+1}

Метод градиентного спуска с дроблением шага

В этом варианте градиентного метода величина шага t_k на каждой итерации выбирается из условия выполнения неравенства

$$f(x^{k+1}) = f(x^k - t_k f'(x^k)) \leq f(x^k) - \epsilon t_k ||f'(x^k)||^2$$

Процедуру нахождения такого t_k оформляют так: выбирается число $\delta \in (0; 1)$ и некоторый начальный шаг t_0, затем для каждого k полагаются $t_k = t_0$ и делают шаг градиентного метода. Если с таким t_k условие выполняется, то переходят к следующему k. Если же не выполняется, то умножают t_k на δ ("дробят шаг") и повторяют эту процедуру до тех пор, пока неравенство не будет выполнено.

Метод наискорейшего градиентного спука

Величина шага t_k в этом методе определяется для каждого значения k из условия

$$\varphi(t_k) = f(x^l - t_k \nabla f(x^k)) \to \min_{t_k}$$

Решение задачи связан с использованием численных методов, когда ищется

$$\min_{t_k \in [a,b]} \varphi(t_k) = \min_{t_k \in [a,b]} f(x^k - t_k \nabla f(x^k))$$

Границы интервала $[a, b]$ задаются пользователем. При этом степень близости найденного значения t_k к оптимальному значению t_k^*, удовлетворяющему условиям $\frac{d\varphi}{dt_k} = 0$, $\frac{d^2\varphi}{dt_k^2} > 0$, зависит от задания интервала $[a, b]$ и точности методов одномерной оптимизации

Метод Флетчера-Ривза

Стратегия метода Флетчера-Ривса состоит в построении последовательности точек $\{x^k\}, k = 0, 1...$ таких, что $f(x^{k+1}) < f(x^k), k = 0, 1...$ Точки последовательности $\{x^k\}$ вычисляются по правилу:

$$x^{k+1} = x^k + t_k\alpha^k, k = 0, 1...$$

$$\alpha^k = -\nabla f(x^0) + \beta_{k-1}\alpha^{k-1}$$

$$\alpha^0 = -\nabla f(x^0)$$

$$\beta_{k-1} = \frac{||\nabla f(x^k)||^2}{||\nabla f(x^{k-1})||^2}$$

Точка x^0 задаётся пользователем, величина шага t_k определяется для каждого значения k из условия

$$\varphi(t_k) = f(x^l - t_k\nabla f(x^k)) \to \min_{t_k}$$

Результаты работы различных методов

Метод градиентного спуска с постоянным шагом

Этот метод не использует методы одномерной оптимизации, так что были проведены запуски с различным начальным шагом

Функция 1: $(x + 3y - 12)^2 + (2x + y + 4)^2$

Шаг	X	Y	Результат	Кол-во итераций
1	1	1	(1,0047895745138775;2,9952098699303784)	40
15	1	1	(1,0047023908229828;2,995297053620389)	43
125	1	1	(1,0048064457018508;2,995192998742741)	41
1	7	5	(1,0049132382961583;2,995086206147026)	45
125	7	5	(1,0050430447129908;2,994956399732048)	48
125	7	5	(1,0050136469481432;2,9949857974955605)	46

Функция 2: $(x^2 + xy + y^2)/500$

Шаг	X	Y	Результат	Кол-во итераций
1	273	25	(0,35289885235045176;-0,3528928884624158)	2928
15	273	25	(0,3470310247596055;-0,34702798287041486)	193
125	273	25	(0,2949223774514892;-0,29492304405577247)	21
1	-1	0,5	(-0,36995429673957003;0,32125685658737896)	387
15	-1	0,5	(-0,3612530235396421;0,3181951726077379)	26

Шаг	X	Y	Результат	Кол-во итераций
125	-1	0,5	(-0,32031282811885053;0,31249967187681277)	3

Метод градиентного спуска с дроблением шага

Этот метод не использует методы одномерной оптимизации, так что были проведены запуски с различными значениями коэфициента дробления

Функция 1: $(x+3y-12)^2+(2x+y+4)^2$

Шаг	Коэф. дробления	X	Y	Результат	Кол-во итераций
1	0.1	-120	115	(0,9973013320626228;3,0024974057002423)	48
1	0.5	-120	115	(0,9988608416852122;3,0010538926468673)	28
1	0.95	-120	115	(1,000118925699312;3,0003807563587355)	51
1	0.1	-120	115	(0,9973013320626228;3,0024974057002423)	48
1	0.5	1	1	(0,9999615753780839;2,999961575142375)	8
1	0.95	1	1	(1,0006860321714897;2,9998340405277846)	35
1	0.1	7	5	(0,9998636092520382;2,9995913833586245)	43
1	0.5	7	5	(1,0009356077135798;2,9991372395207043)	20
1	0.95	7	5	(0,9999719769043325;2,9995215242972)	41

Метотод наискорейшего градиентного спуска

Этот метод был протестирован с использованием метода золотого сечения и метода фибоначи

Функция 1: $(x+3y-12)^2+(2x+y+4)^2$

Метод	X	Y	Результат	Кол-во итераций
Фибоначи	-120	115	(0,9985158419025808;3,0013786322323948)	24
Золотое сечение	-120	115	(0,999017817613472;3,0009095853178493)	26
Фибоначи	1	1	(1,000396978027575;2,9996121926786548)	7
Золотое сечение	1	1	(1,000445531718261;2,999565388887183)	7
Фибоначи	7	5	(1,0000621472224815;2,9999364775873687)	9
Золотое сечение	7	5	(1,0000536178643151;3,000019258652365)	6

Функция 2: $(x^2+xy+y^2)/500$

Метод	X	Y	Результат	Кол-во итераций
Фибоначи	273	25	(0,2998207278607703;-0,2998213891610363)	27
Золотое сечение	273	25	(0,2998132732165105;-0,2998139345220134)	27
Фибоначи	-1	0,5	(-0,3136015979968651;0,3008006654575026)	4
Золотое сечение	-1	0,5	(-0,3136003247887139;0,3007996751992102)	4
Фибоначи	2	7	(-0,3343677277113275;0,3367264745669717)	9
Золотое сечение	2	7	(-0,3343650050239134;0,33672363453667353)	9

Функция 3: $(x^2+y-11)^2+(x+y^2-7)^2$ (Функция Химмельблау)

Метод	X	Y	Результат	Кол-во итераций
Фибоначи	4	3,5	(2,9997995148656207;2,000321376896734)	9
Золотое сечение	4	3,5	(-3,7792617189802784;-3,28312766571972)	14
Фибоначи	-2	4	(-2,8050575513465;3,1313618992402494)	12
Золотое сечение	-2	4	(-2,805114875034324;3,1313115342708877)	113
Фибоначи	-2	-4,2	(-3,7792800220788245;-3,283216788591051)	17
Золотое сечение	-2	-4,2	(3,2177307042591488;1,3863001585023)	100000

Метод Флетчера-Ривза

Дли минимизации одномерной функции был использован метод золотого сечения. Так же было протестировано поведение метода на не квадратичной функции.

Функция 1: $(x+3y-12)^2+(2x+y+4)^2$

X	Y	Результат	Количество итераций
-120	115	(1,0000000766028805;2,9999999628017093)	2
1	1	(0,9999978445640791;2,9999973055625646)	2
7	5	(0,9999974148213207;2,9999976869358282)	2

Функция 2: $(x^2+xy+y^2)*1000$

X	Y	Результат	Количество итераций

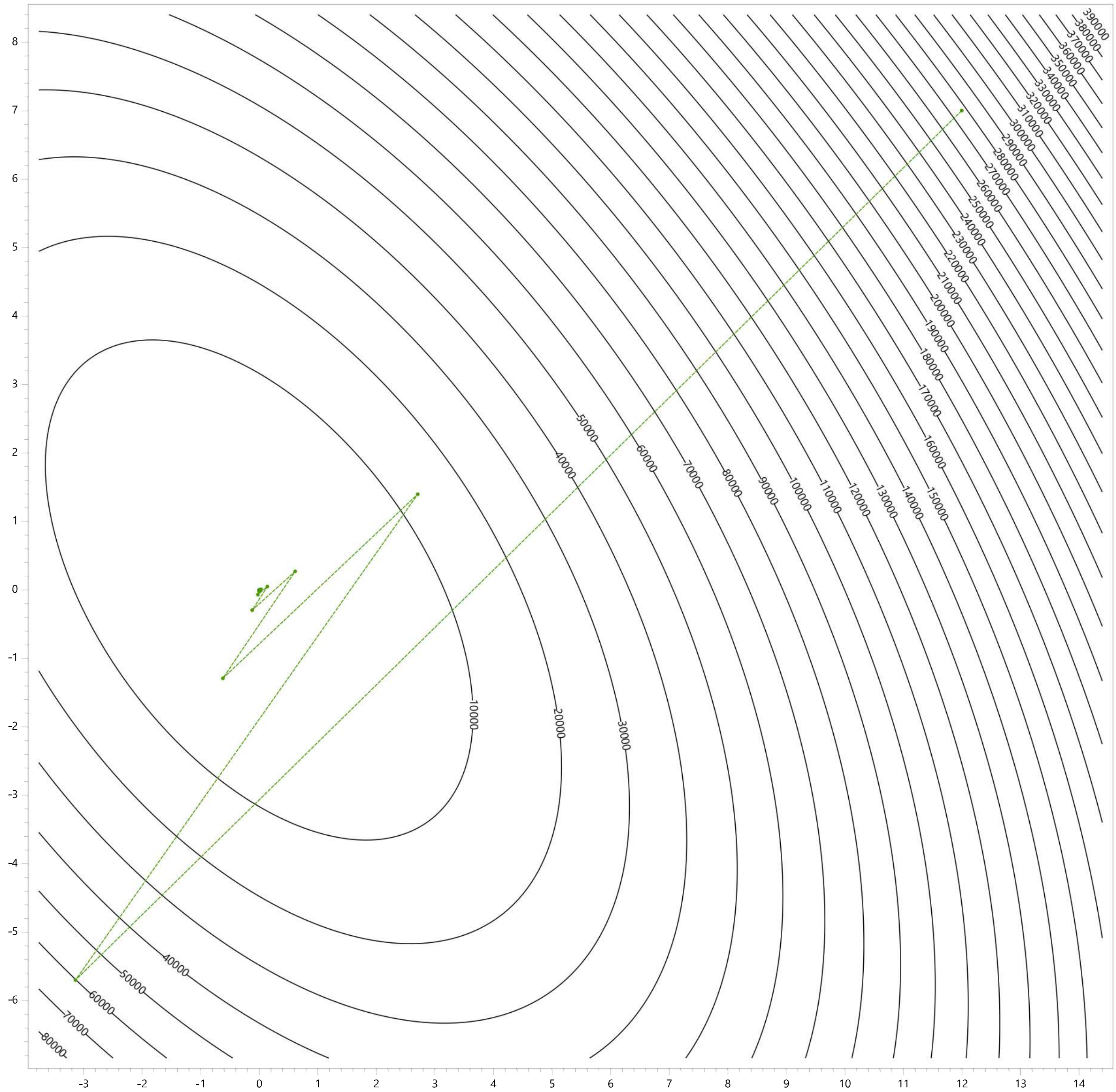
X	Y	Результат	Количество итераций
273	283	(3,733272163884204E-09;-3,726700884122549E-09)	3
-1	2	(-3,1590929872038487E-10;-4,998025087132874E-07)	2
12	7	(-4,438395862139289E-08;4,569474561841615E-08)	3

Функция 3: $\log(x^2 - xy + 3y^2 + 3) + 5$

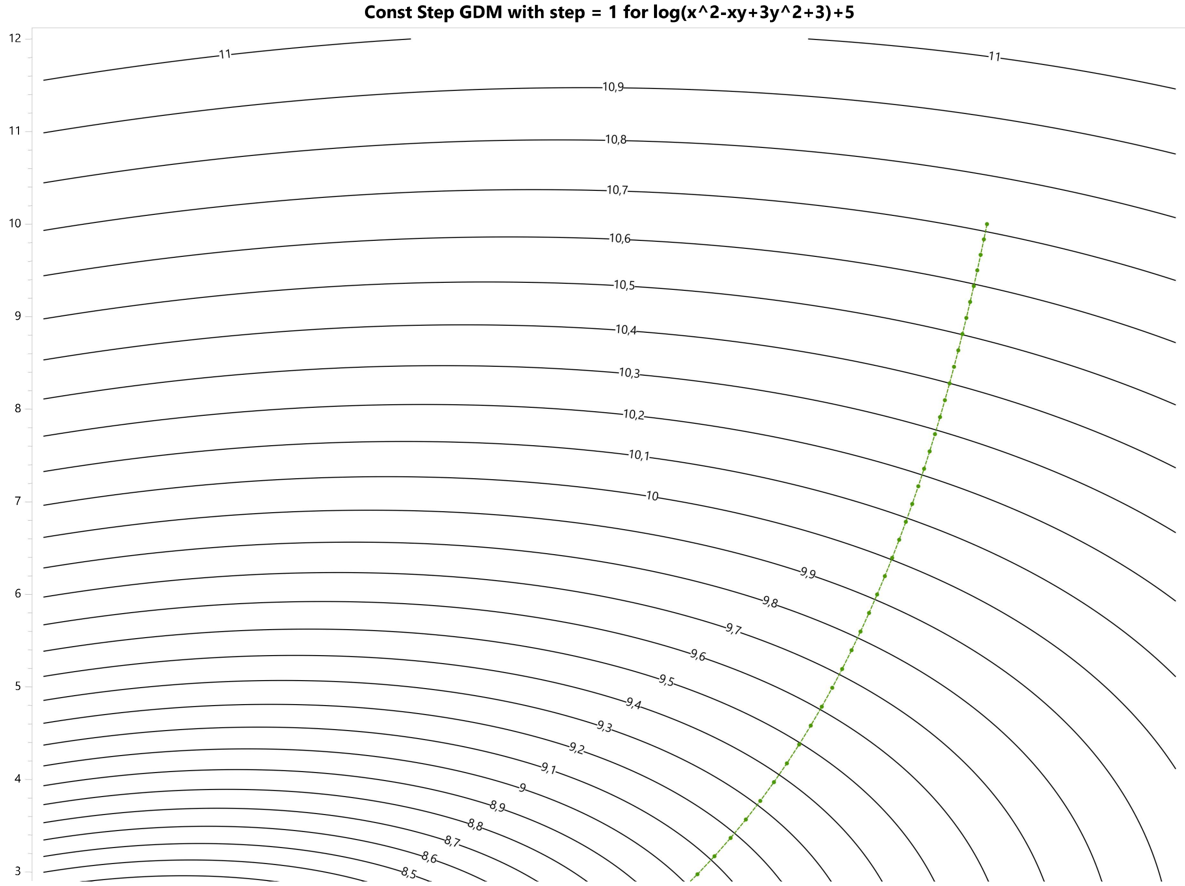
X	Y	Результат	Количество итераций
10	10	(-0,09224838099444235;-0,023870664715973867)	6236
35	72	(4,72343586167181;-0,3503747609778718)	100000
1	2	(0,0016217405735211785;0,0003176462913834176)	16

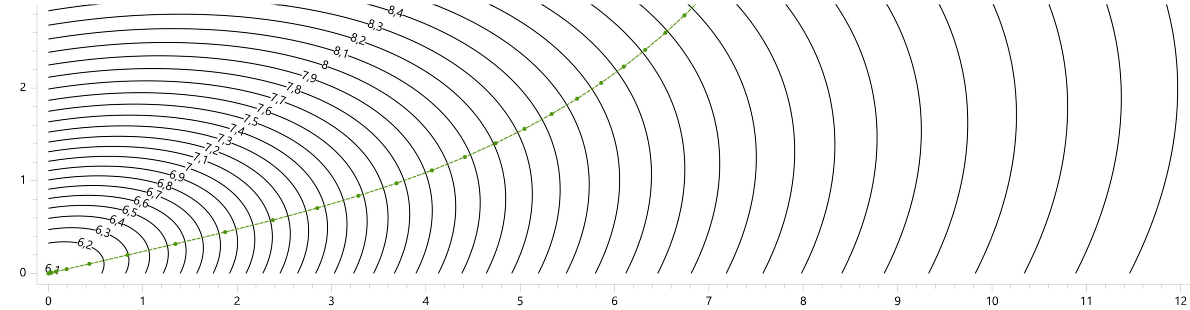
Графики

Const Step GDM with step = 1 for 1000∗(x^2+xy+y^2)

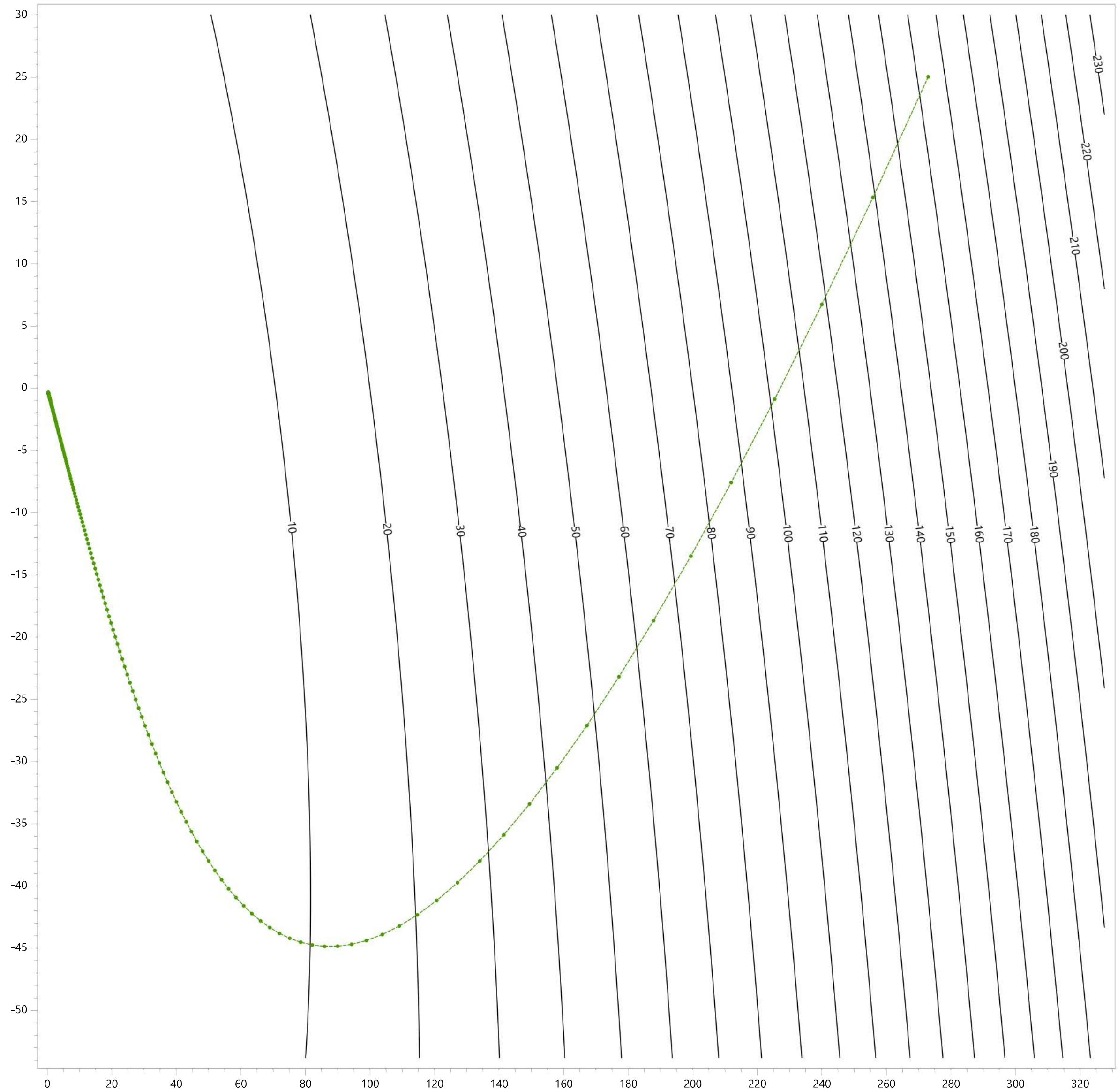


Const Step GDM with step = 1 for log(x^2-xy+3y^2+3)+5

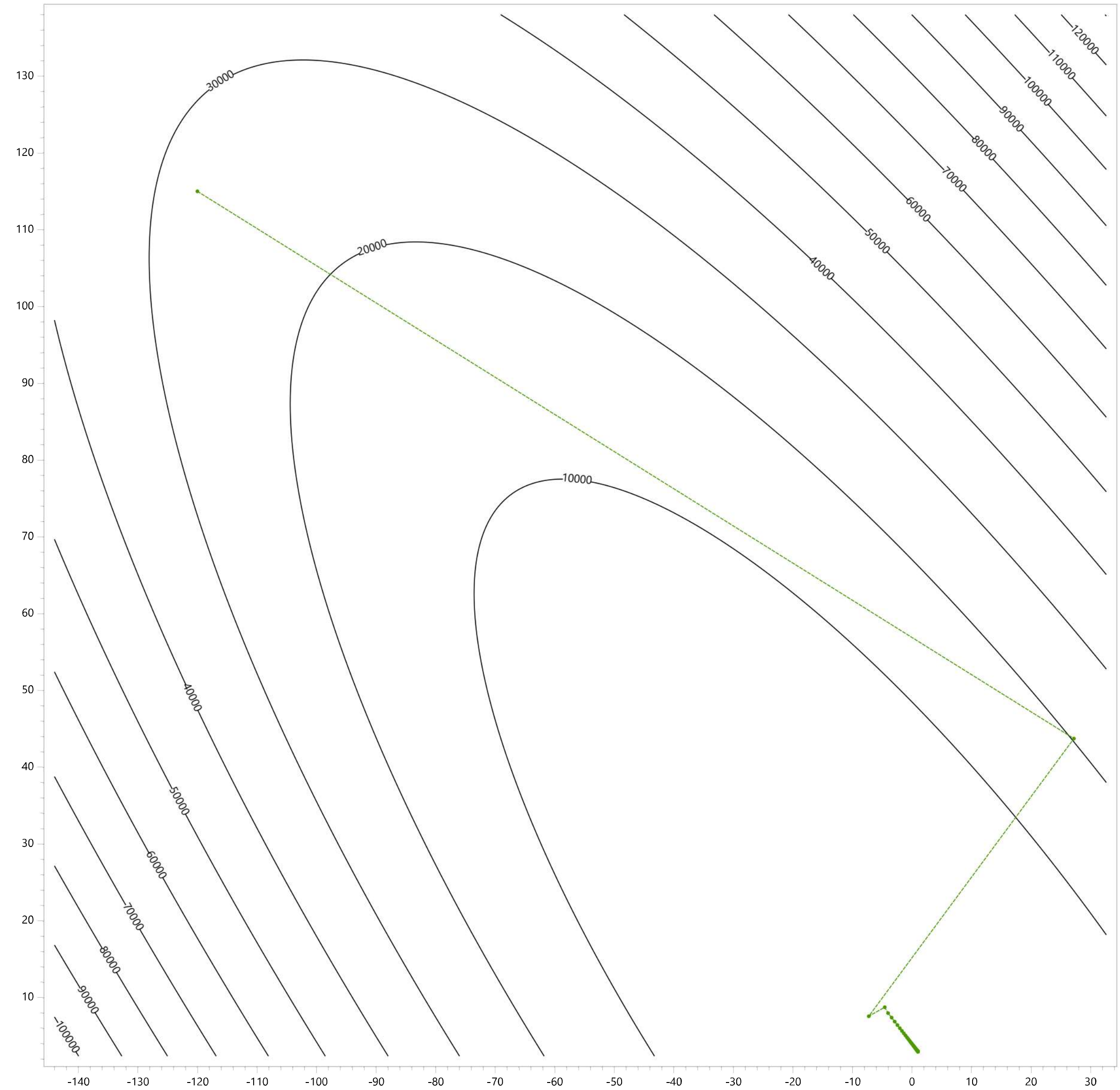


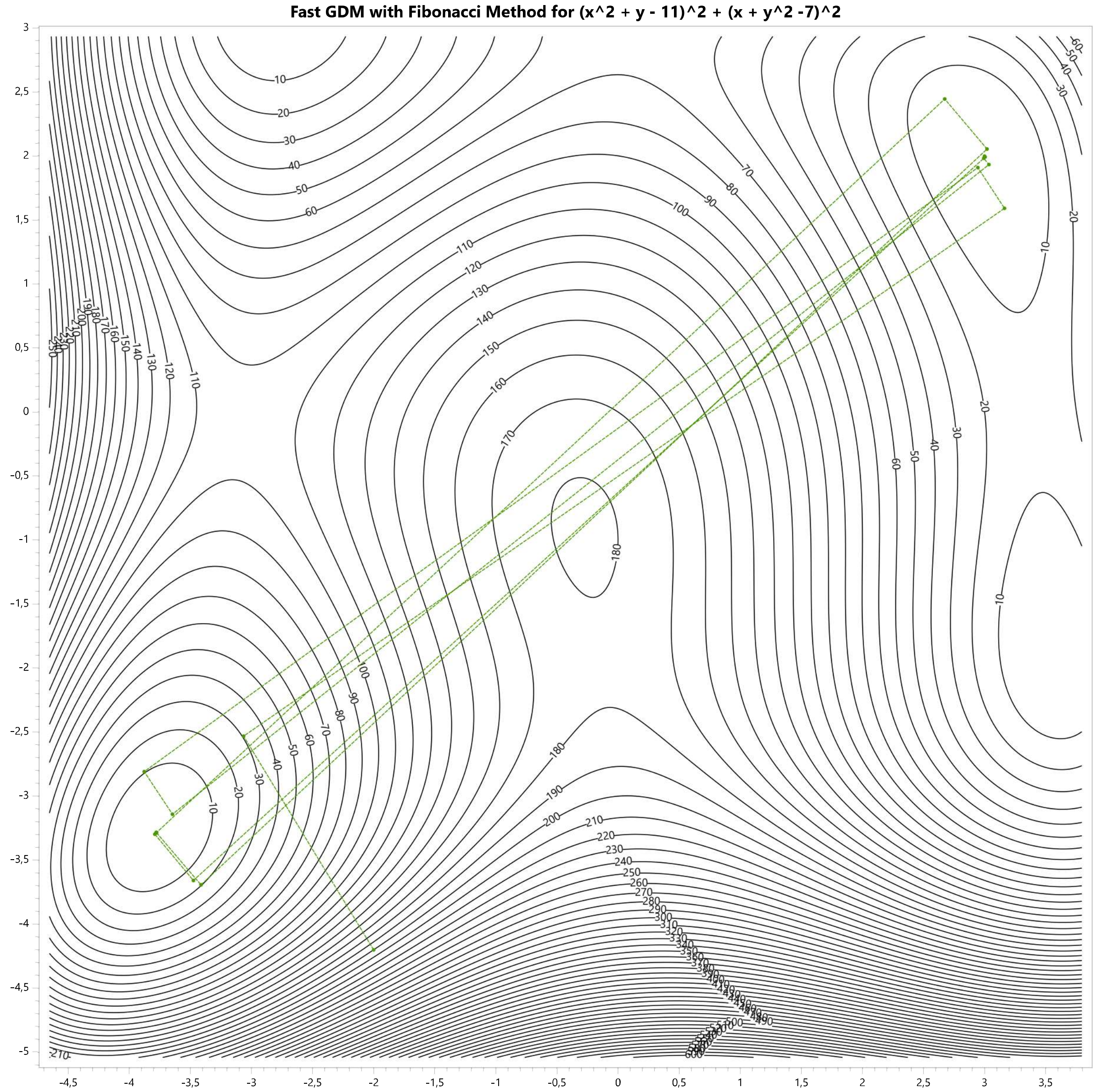


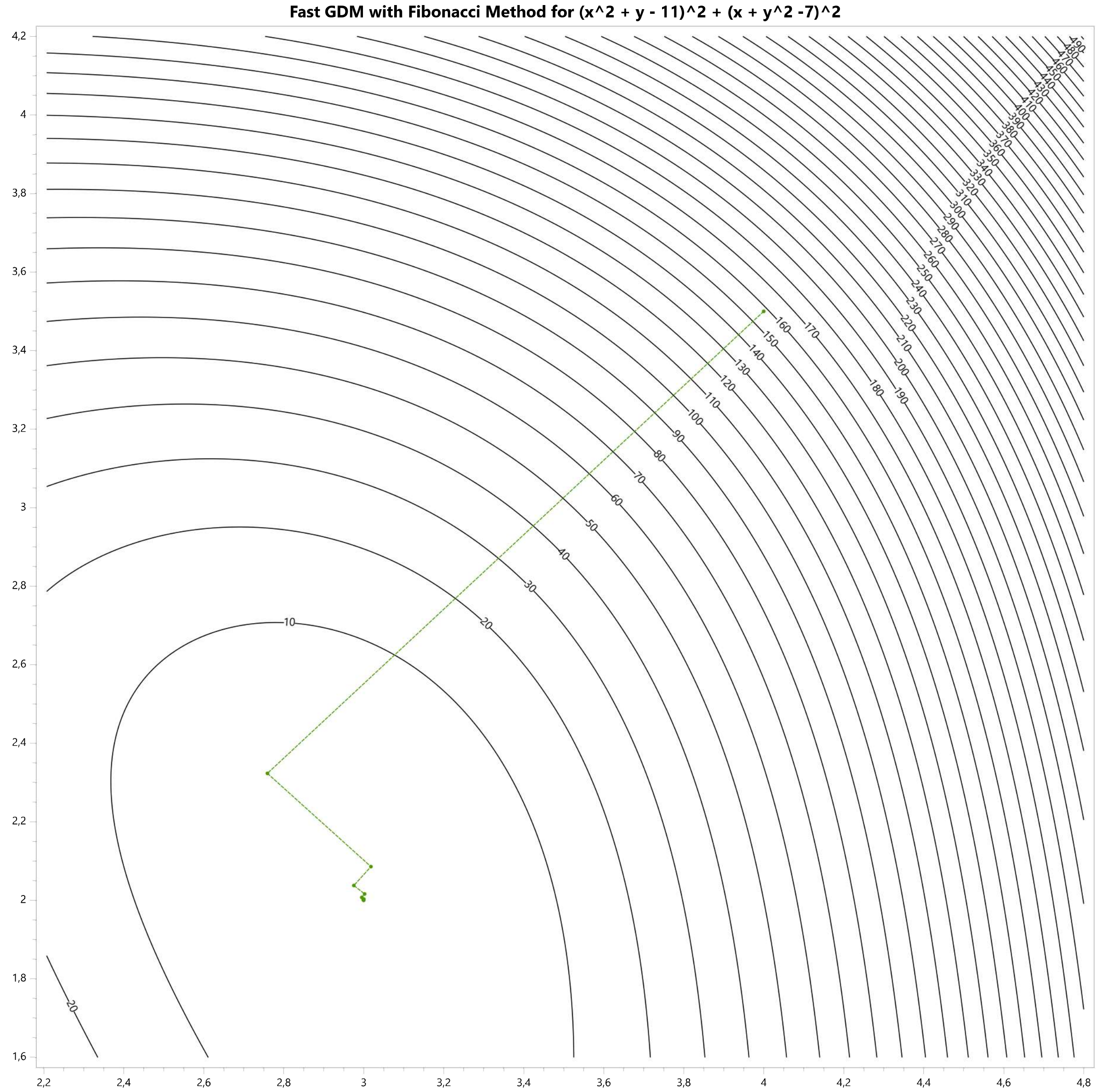
Const Step GDM with step = 15 for (x^2+xy+y^2) div 500



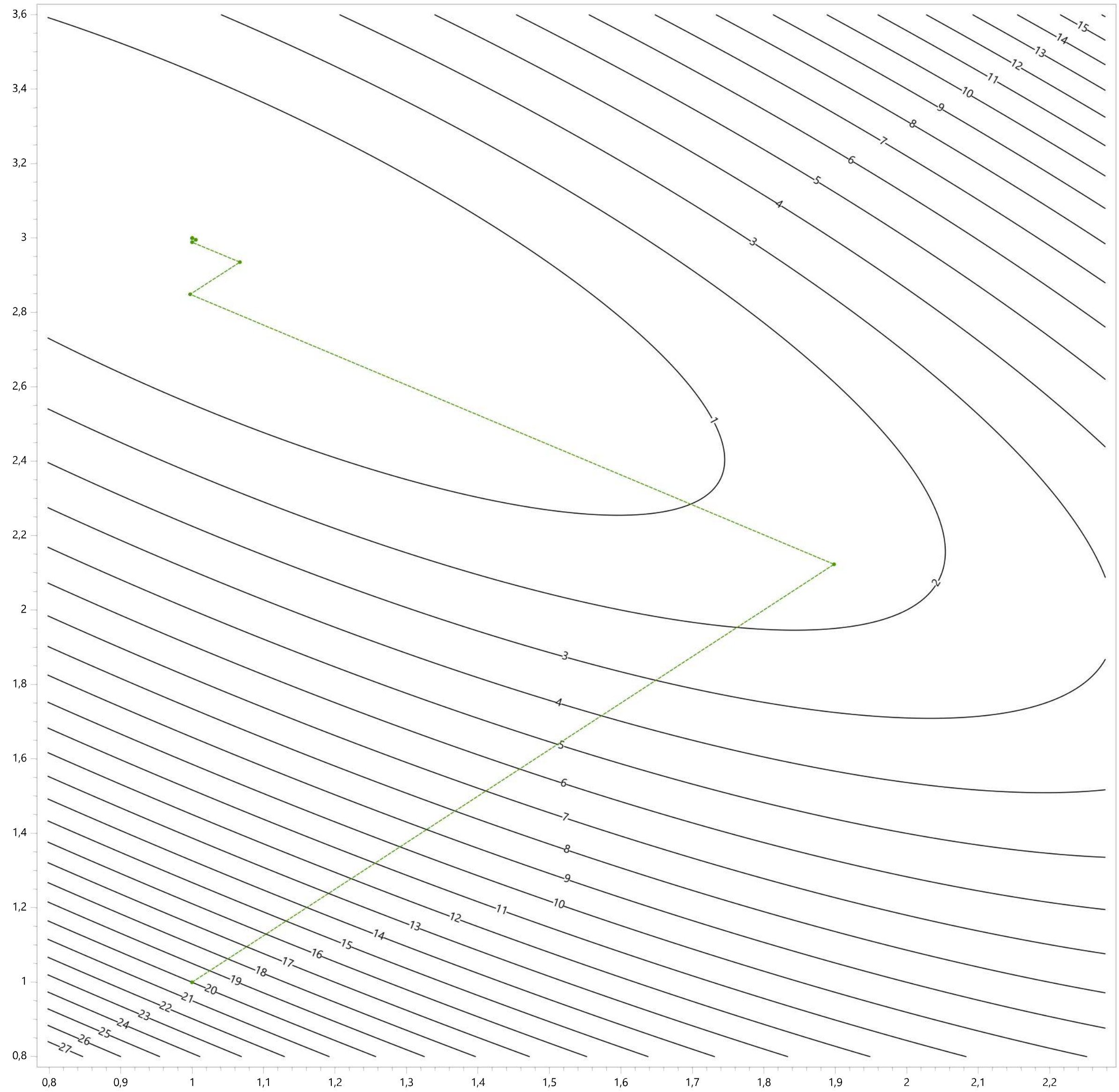
Const Step GDM with step = 15 for (x+3∗y-12)^2+(2∗x+y+4)^2

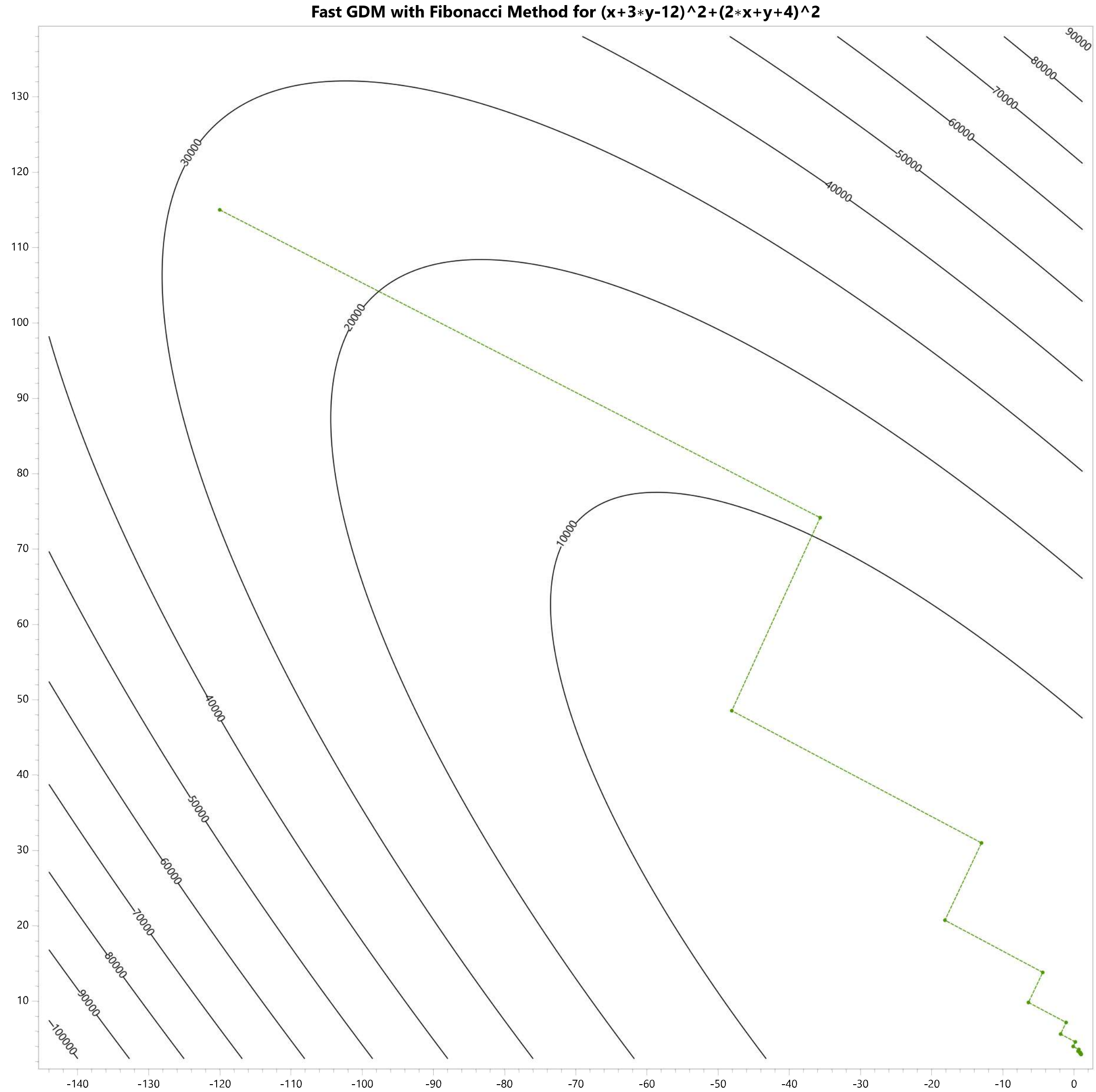


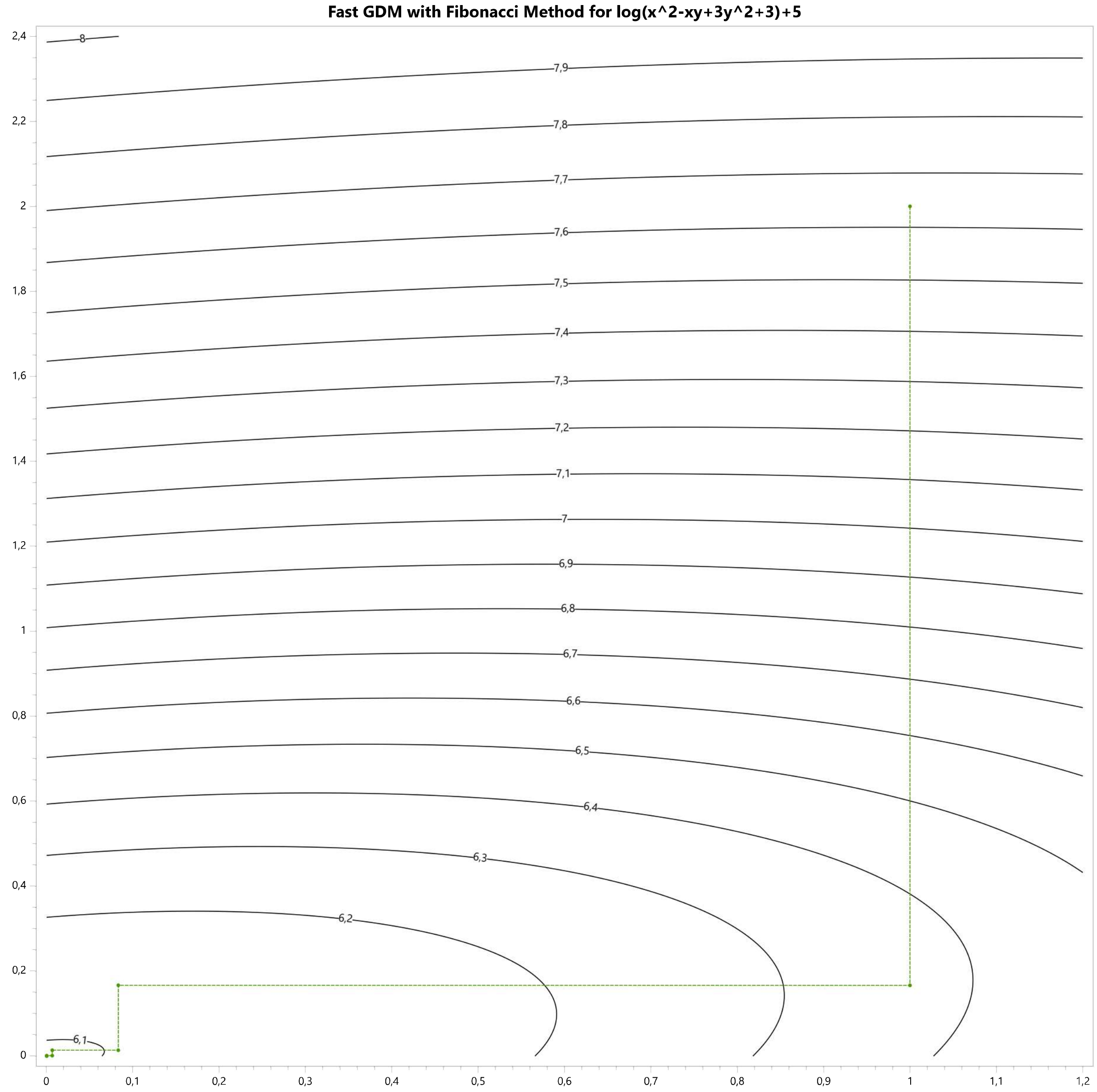

Fast GDM with Fibonacci Method for (x^2 + y - 11)^2 + (x + y^2 -7)^2


Fast GDM with Fibonacci Method for (x^2 + y - 11)^2 + (x + y^2 -7)^2
4,2
4
3,8
3,6
3,4
3,2
3
2,8
2,6
2,4
2,2
2
1,8
1,6
2,2
2,4
2,6
2,8
3
3,2
3,4
3,6
3,8
4
4,2
4,4
4,6
4,8
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490

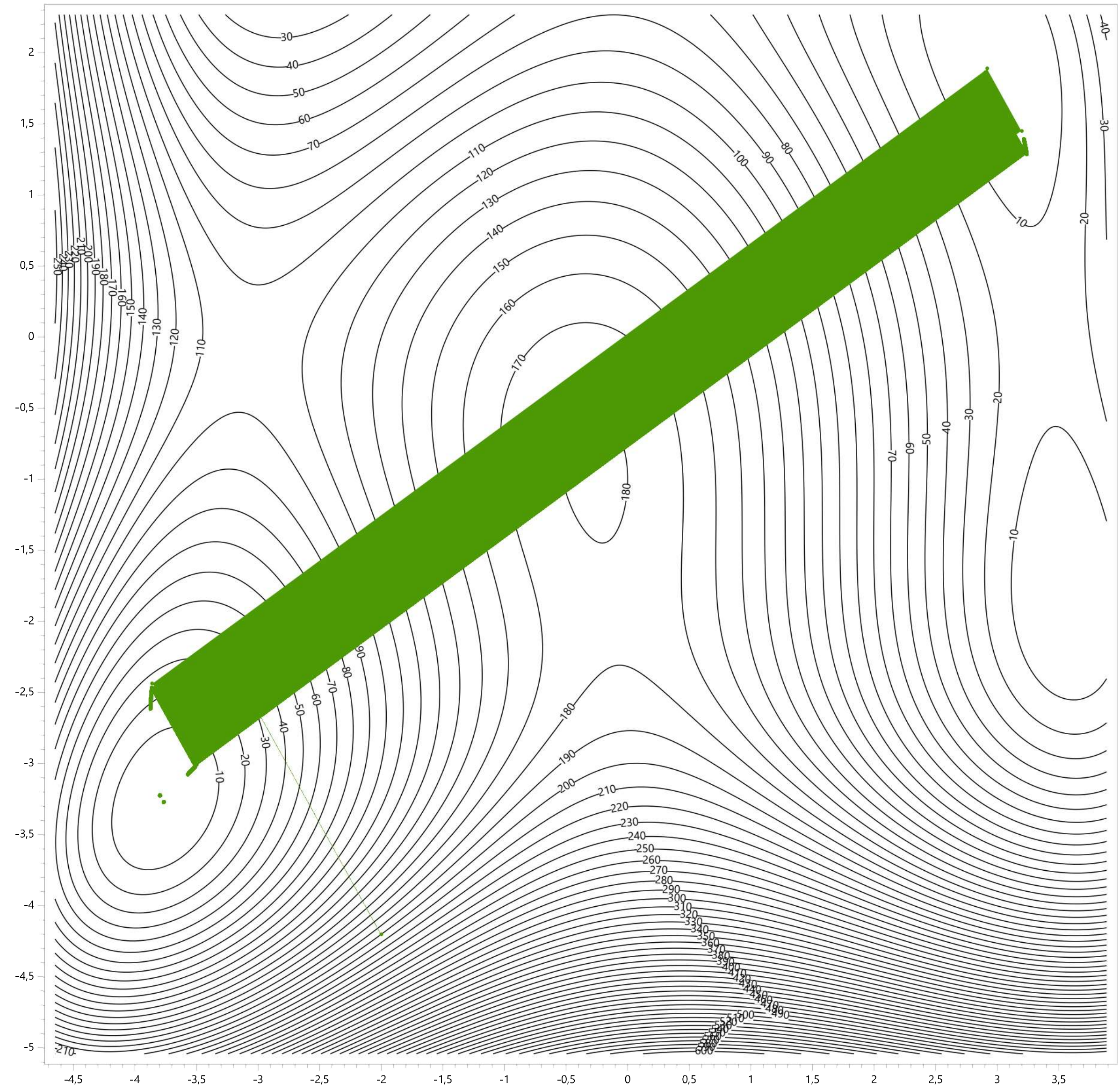
Fast GDM with Fibonacci Method for (x+3*y-12)^2+(2*x+y+4)^2



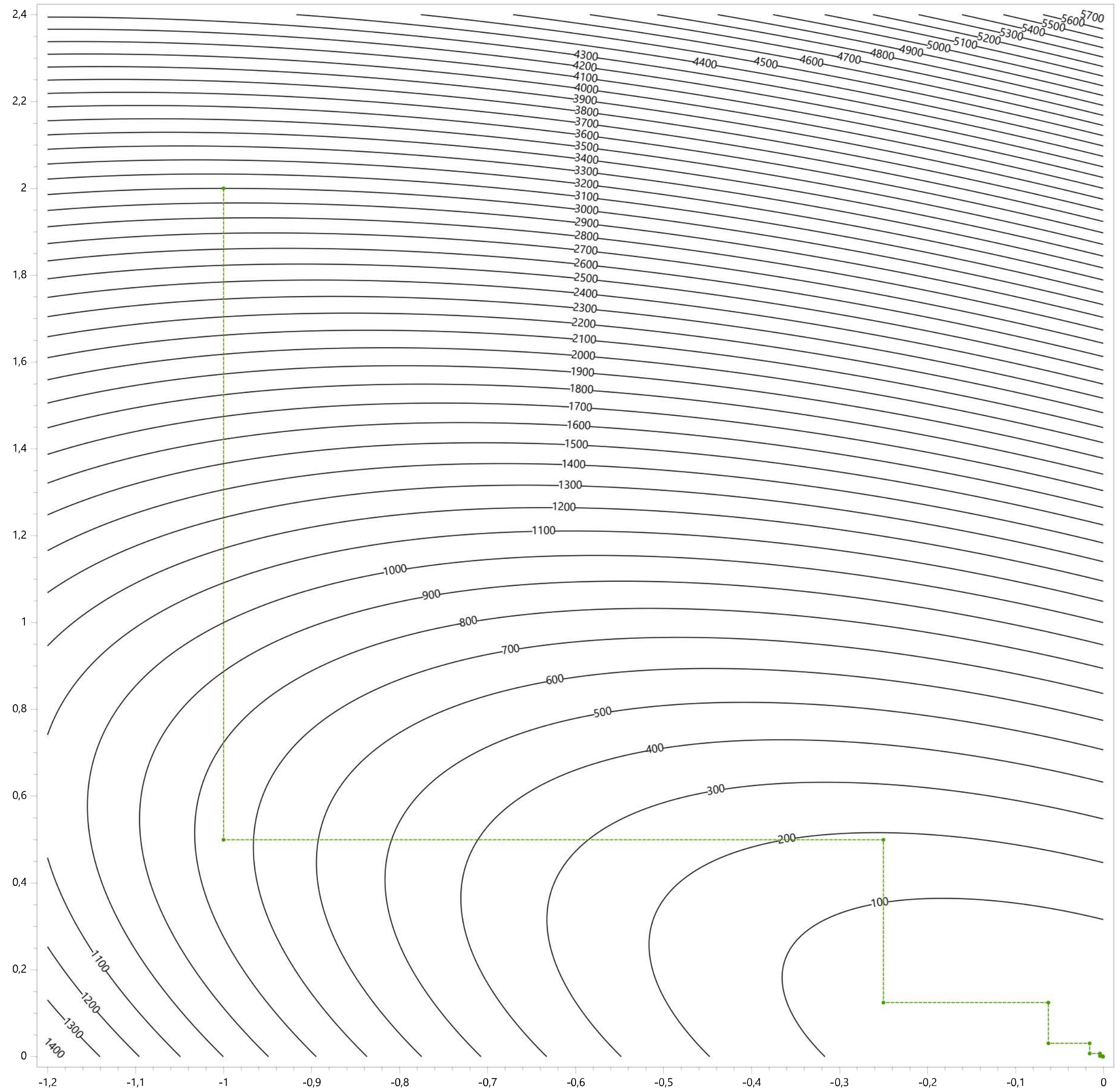

Fast GDM with Fibonacci Method for (x+3*y-12)^2+(2*x+y+4)^2
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
-140
-130
-120
-110
-100
-90
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
90
100
110
120
130

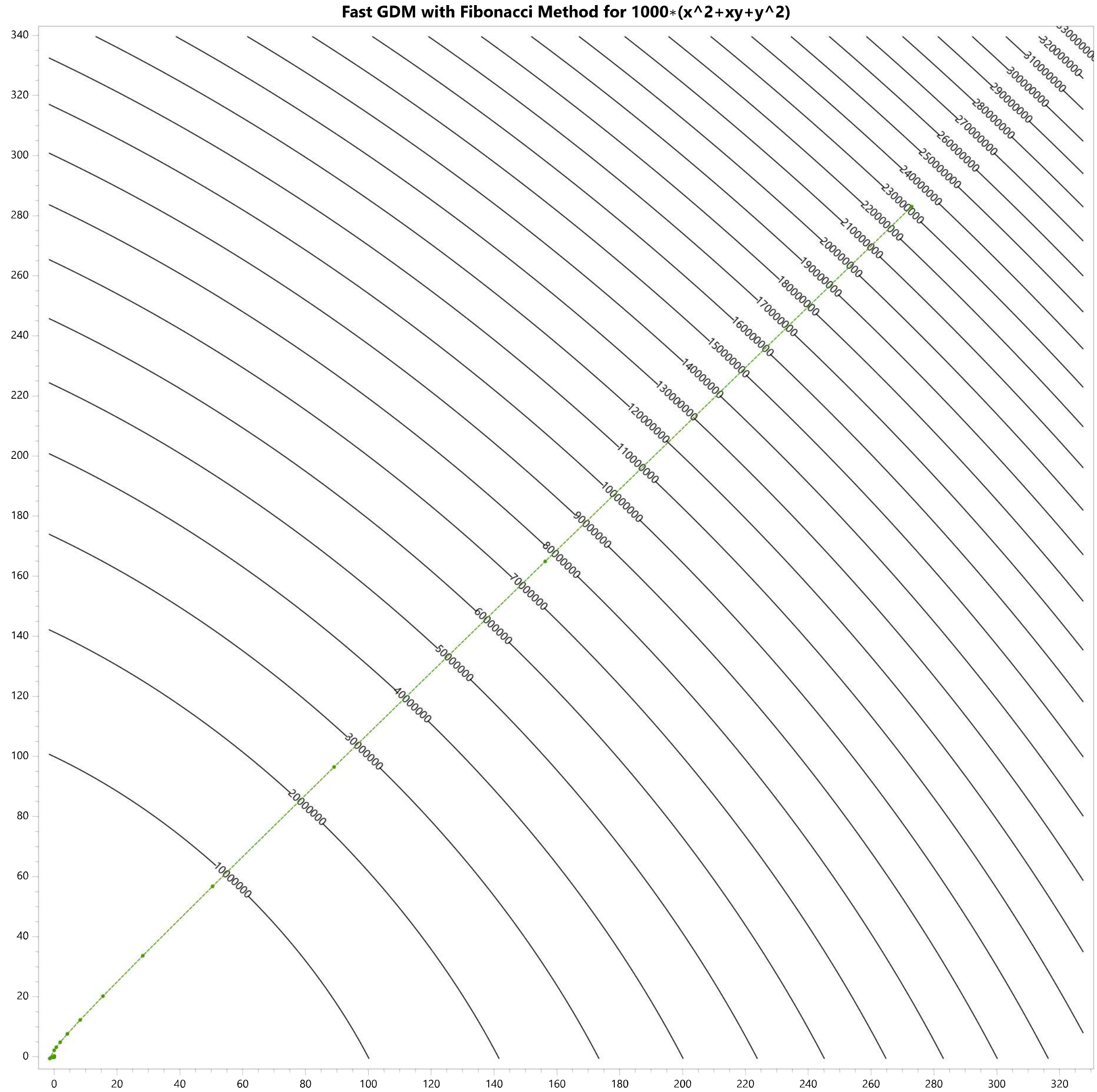

Fast GDM with Fibonacci Method for log(x^2-xy+3y^2+3)+5
8
7,9
7,8
7,7
7,6
7,5
7,4
7,3
7,2
7,1
7
6,9
6,8
6,7
6,6
6,5
6,4
6,3
6,2
6,1
0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1
1,1
1,2
0,2
0,4
0,6
0,8
1,2
1,4
1,6
1,8
2
2,2
2,4

Fast GDM with Golden Ratio Method for (x^2 + y - 11)^2 + (x + y^2 -7)^2

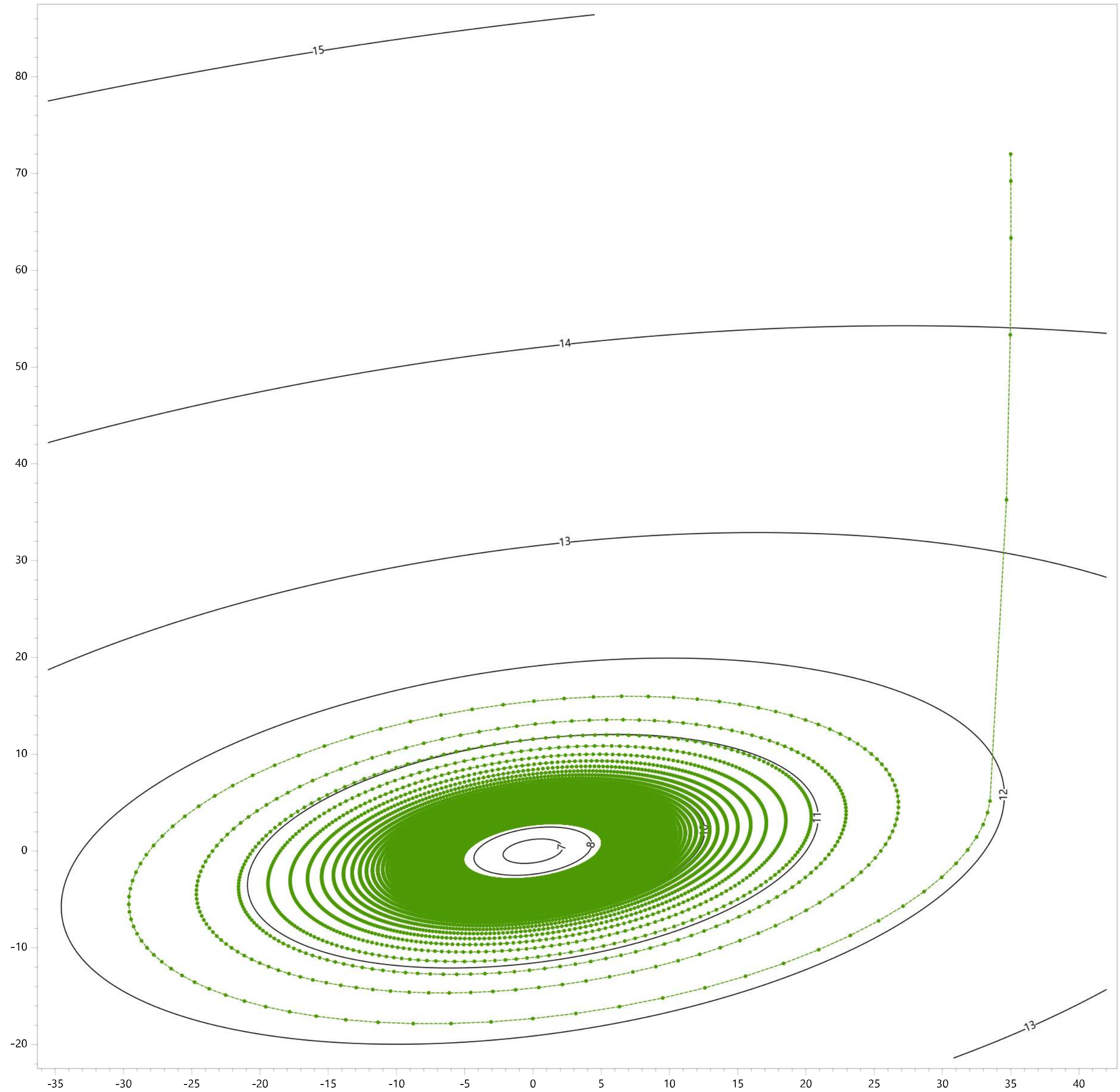


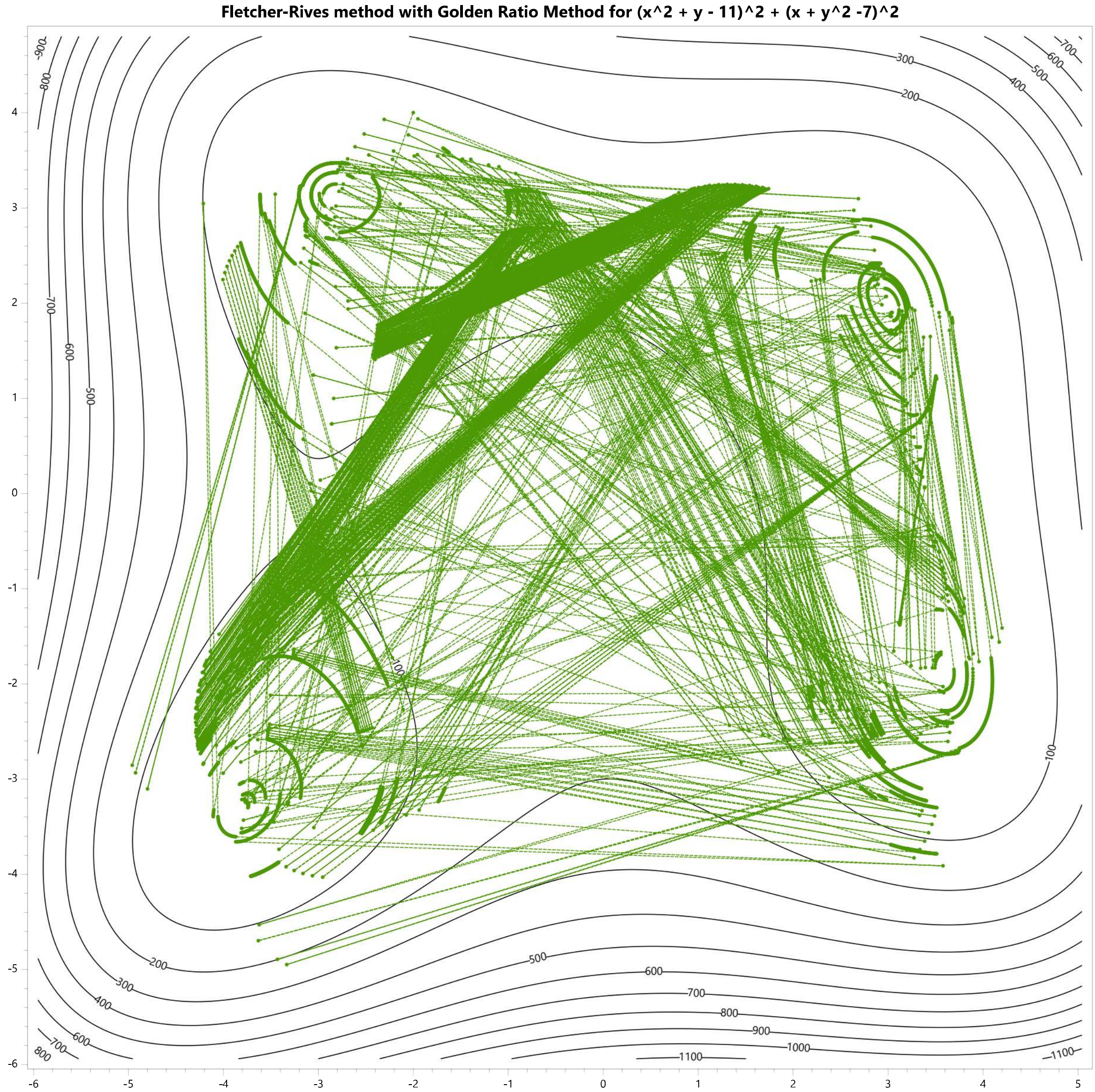
Fast GDM with Golden Ratio Method for 1000∗(x^2+xy+y^2)



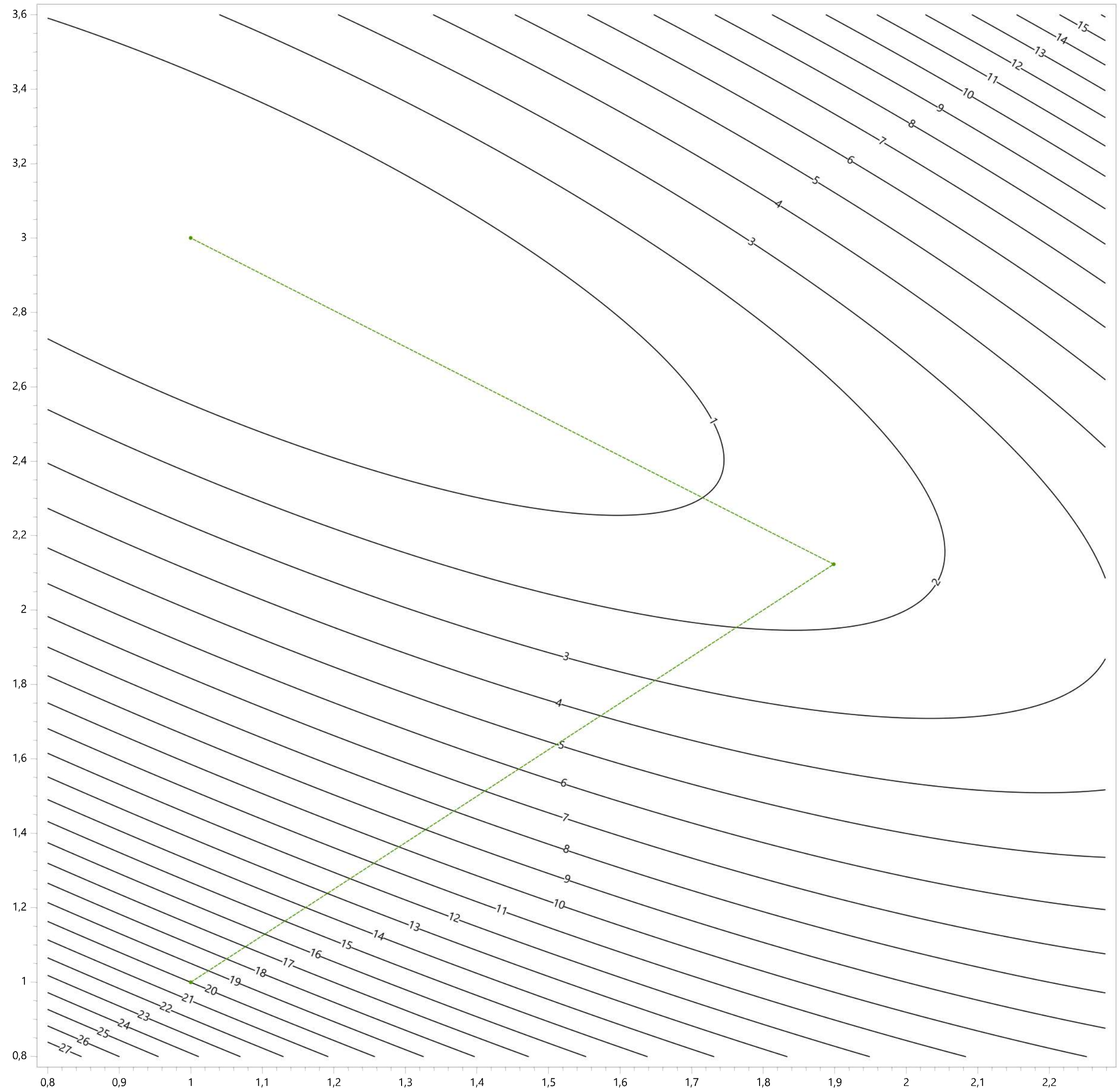

Fast GDM with Fibonacci Method for 1000∗(x^2+xy+y^2)
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
10000000
20000000
30000000
40000000
50000000
60000000
70000000
80000000
90000000
100000000
110000000
120000000
130000000
140000000
150000000
160000000
170000000
180000000
190000000
200000000
210000000
220000000
230000000
240000000
250000000
260000000
270000000
280000000
290000000
300000000
310000000
320000000

Fletcher-Rives method with Golden Ratio Method for log(x^2-xy+3y^2+3)+5

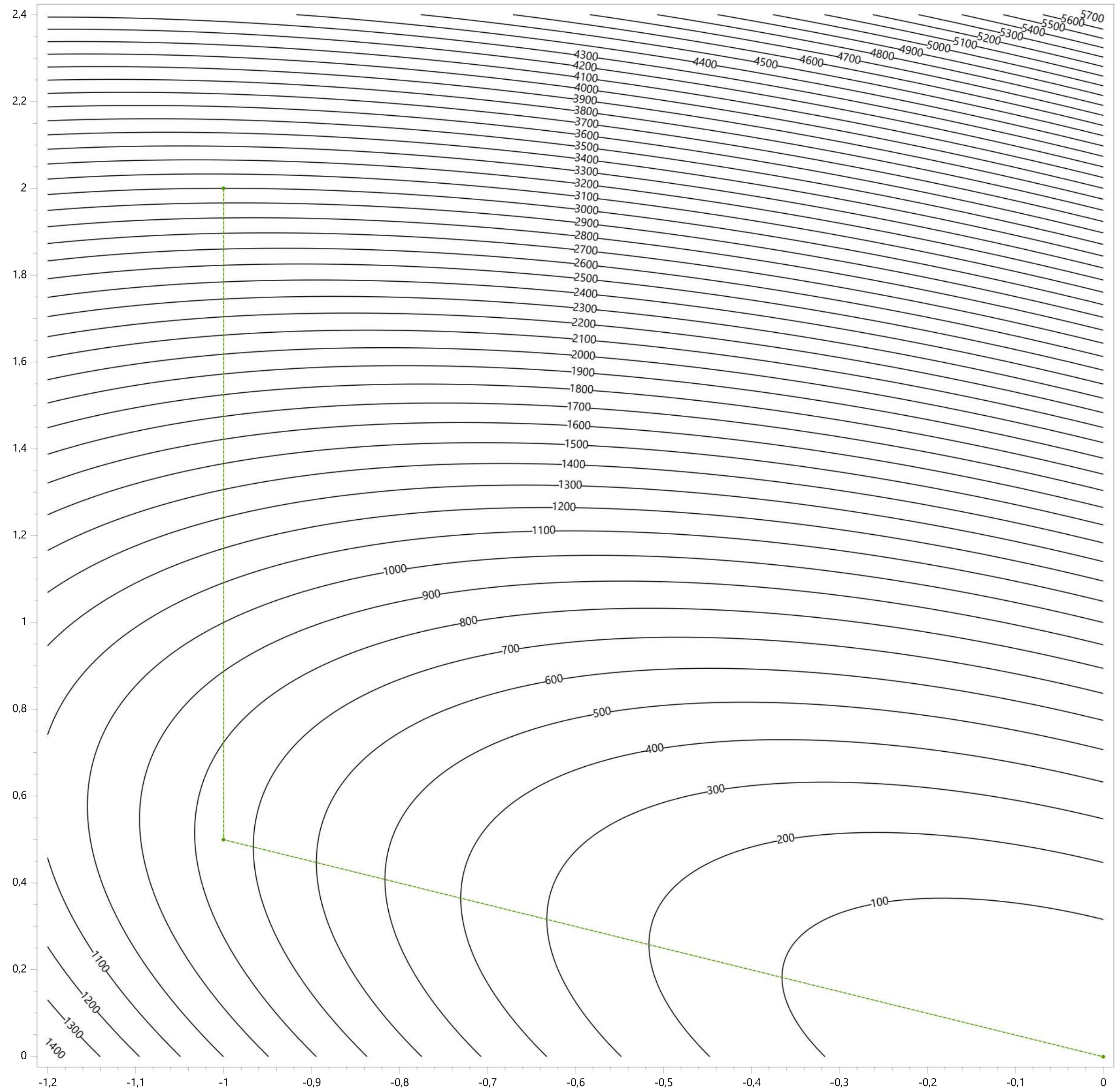



Fletcher-Rives method with Golden Ratio Method for (x^2 + y - 11)^2 + (x + y^2 -7)^2
100
200
300
400
500
600
700
800
900
1000
1100
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5

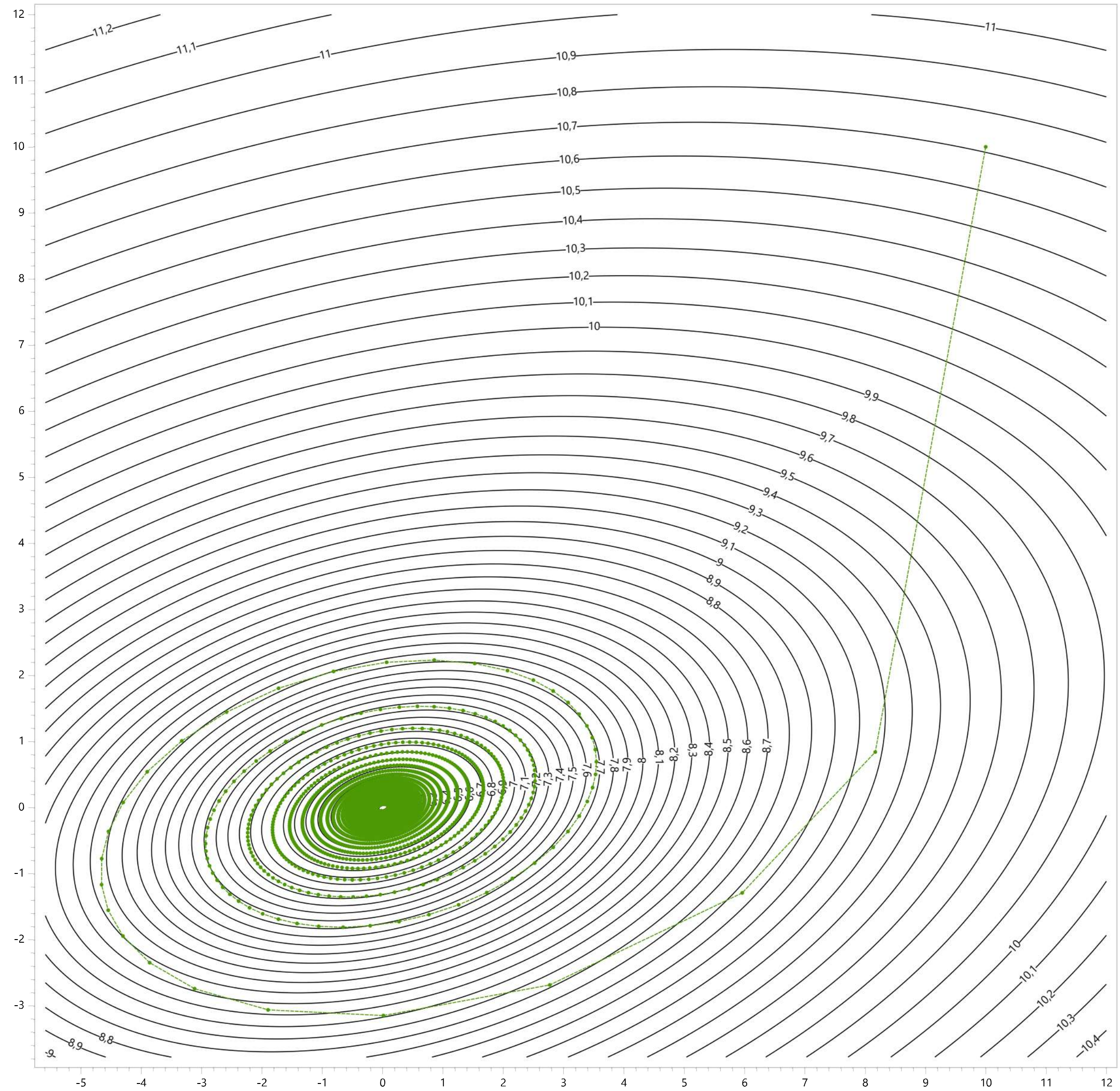
Fletcher-Rives method with Golden Ratio Method for (x+3∗y-12)^2+(2∗x+y+4)^2



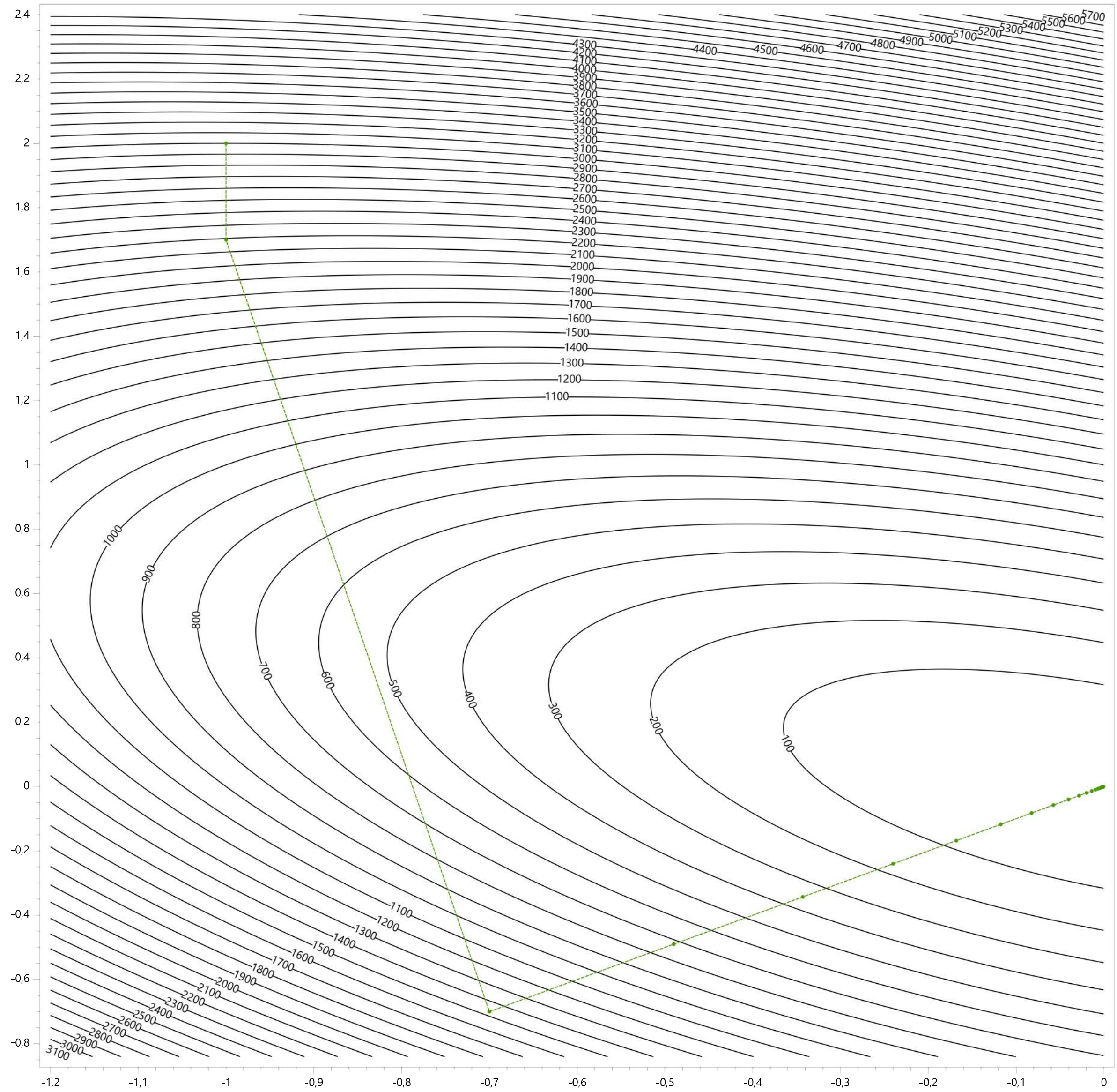
Fletcher-Rives method with Golden Ratio Method for 1000∗(x^2+xy+y^2)



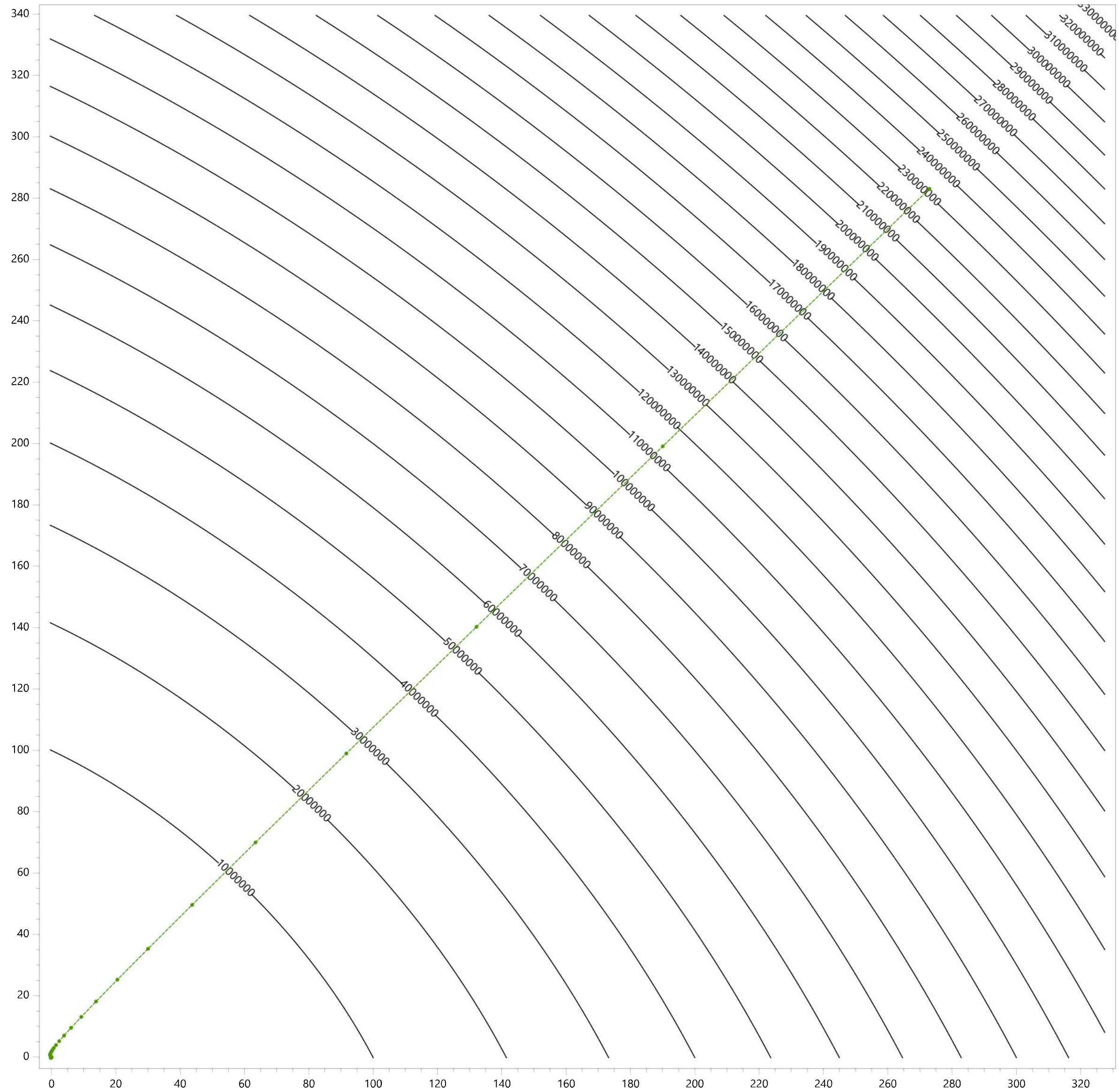
Fletcher-Rives method with Golden Ratio Method for log(x^2-xy+3y^2+3)+5

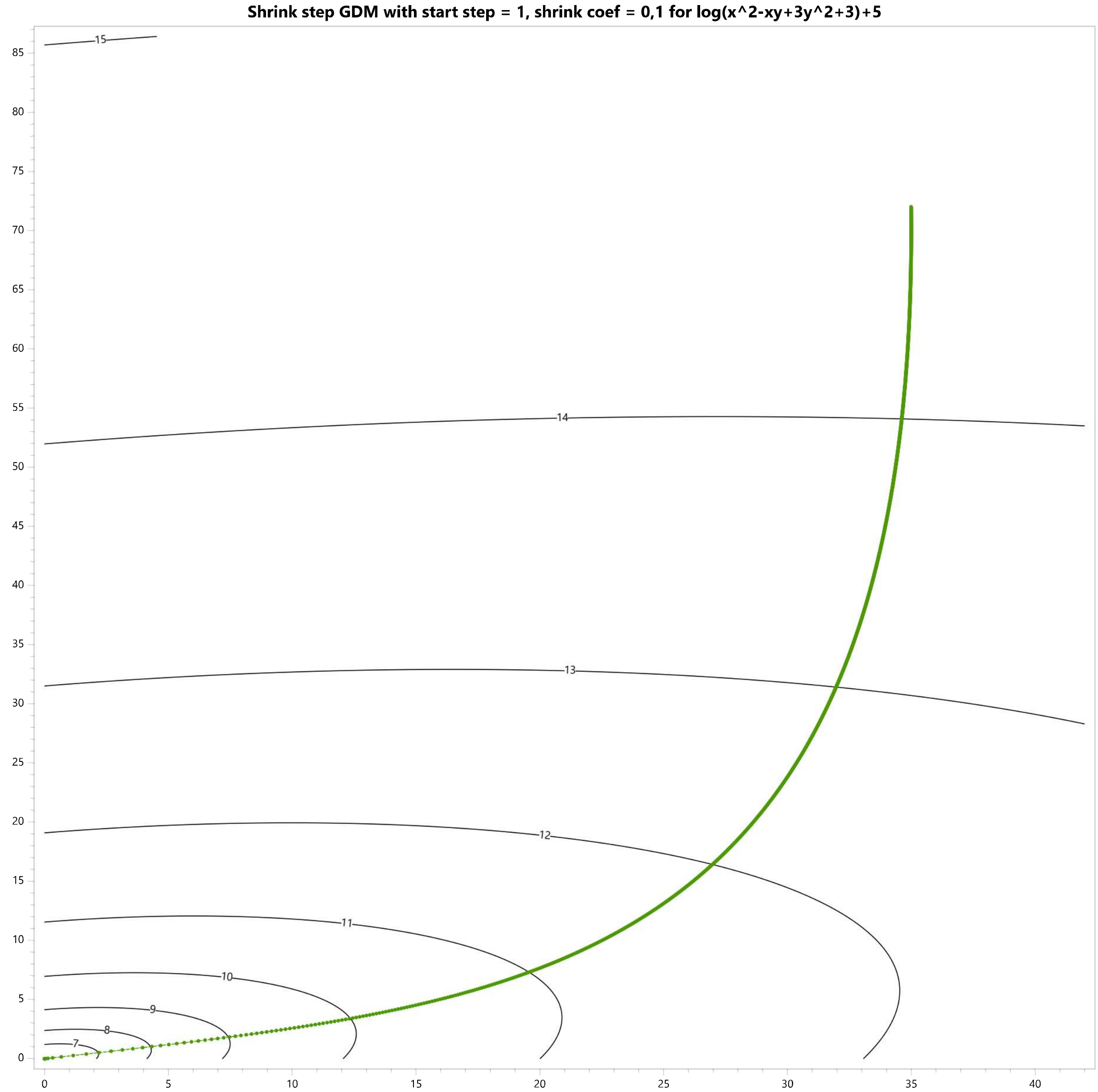


Shrink step GDM with start step = 1, shrink coef = 0,1 for 1000∗(x^2+xy+y^2)

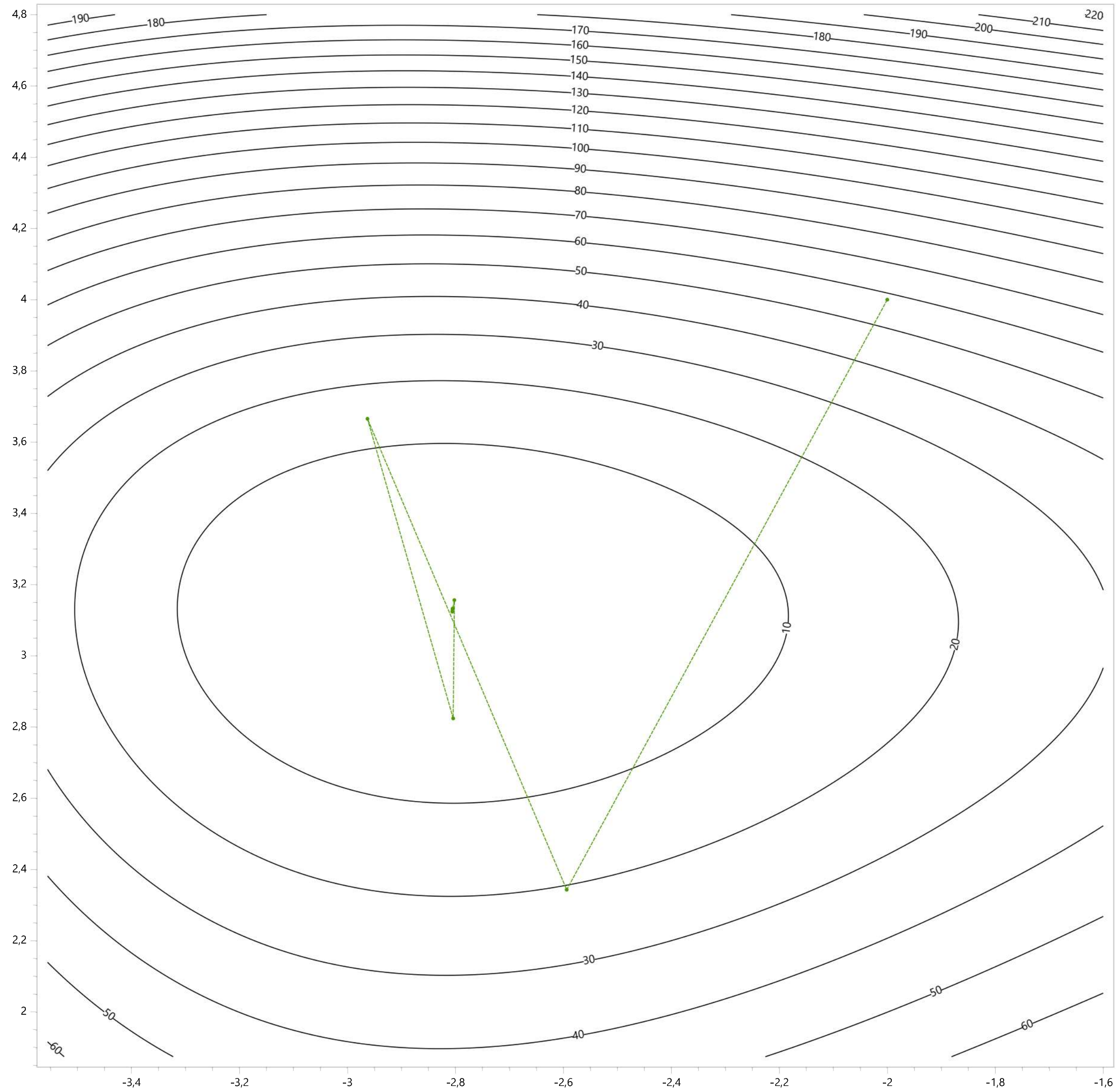


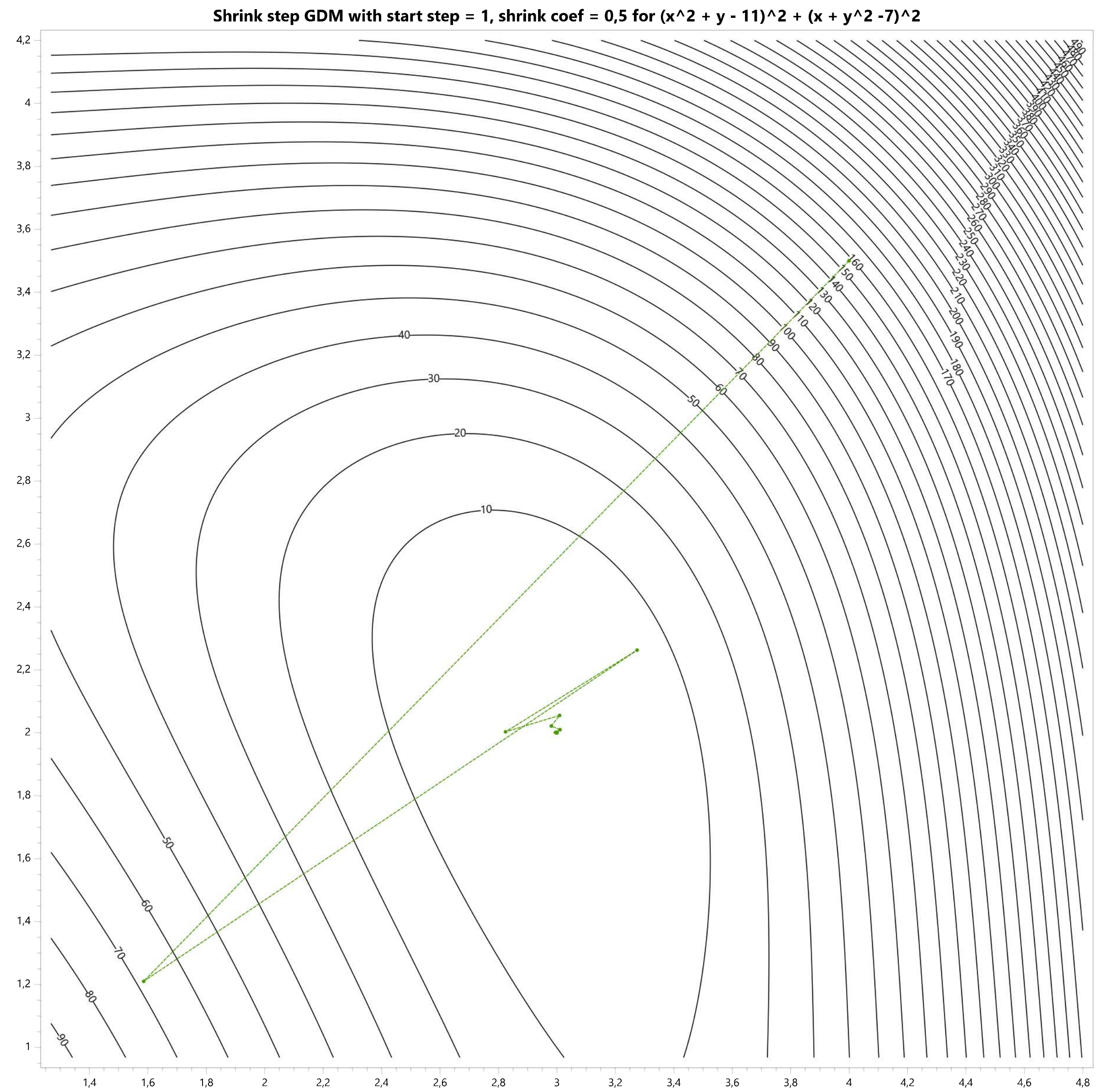
Shrink step GDM with start step = 1, shrink coef = 0,1 for 1000*(x^2+xy+y^2)



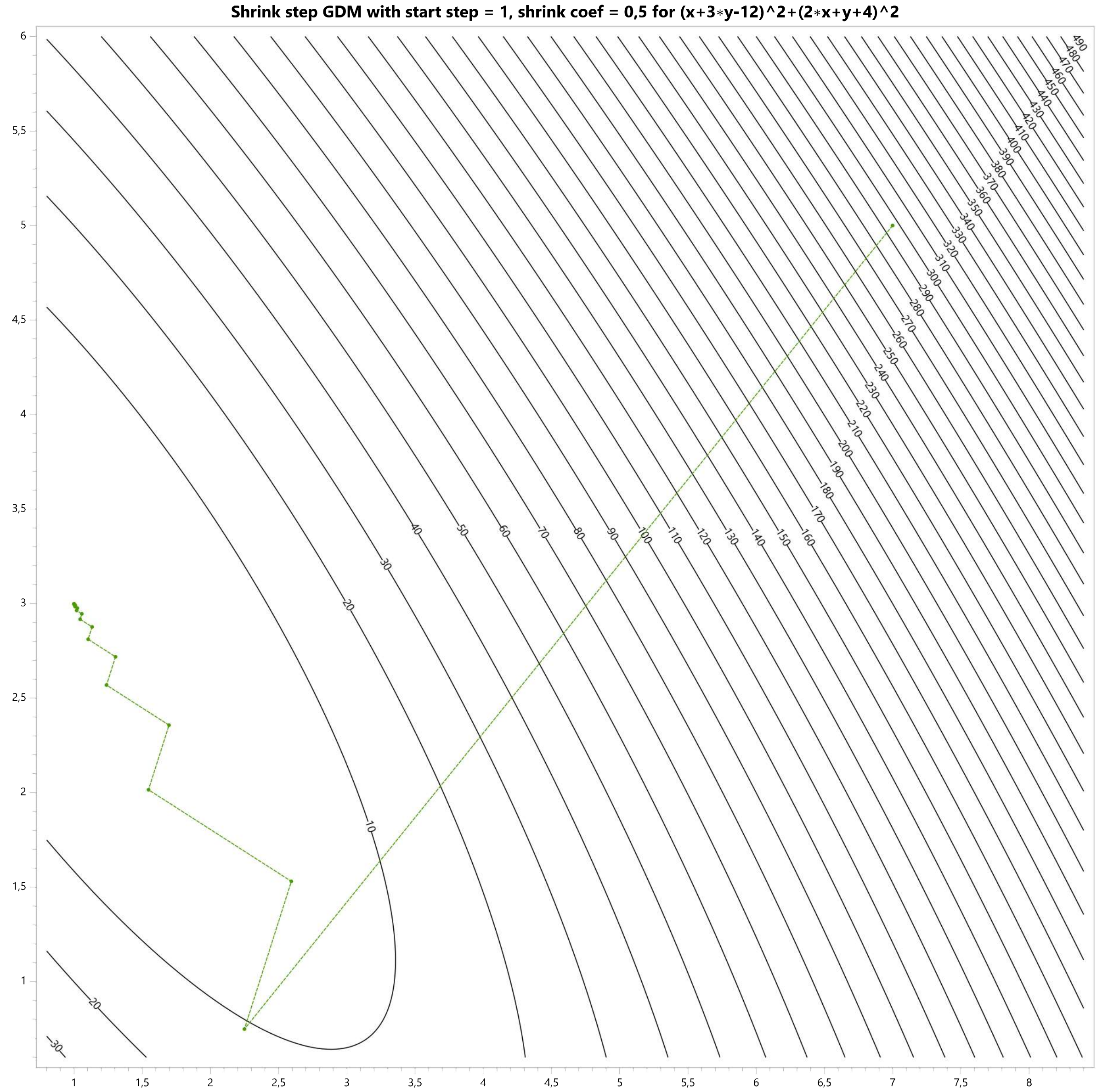

Shrink step GDM with start step = 1, shrink coef = 0,1 for log(x^2-xy+3y^2+3)+5
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
35
40
15
14
13
12
11
10
9
8
7

Shrink step GDM with start step = 1, shrink coef = 0,5 for (x^2 + y - 11)^2 + (x + y^2 -7)^2





Shrink step GDM with start step = 1, shrink coef = 0,5 for (x^2 + y - 11)^2 + (x + y^2 -7)^2
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
1,4
1,6
1,8
2
2,2
2,4
2,6
2,8
3
3,2
3,4
3,6
3,8
4
4,2
4,4
4,6
4,8
1
1,2
1,4
1,6
1,8
2
2,2
2,4
2,6
2,8
3
3,2
3,4
3,6
3,8
4
4,2



Анализ результатов

Метод градиентного спуска с постоянным шагом

Скорость сходимости метода зависит от выбора шага и точки: при достаточно пологой функции и маленьком шаге количество итераций очень сильно увеличивается. При этом при большом значении шага и близкой начальной точке количество итераций не сильно увеличивается, так как шаг быстро уменьшается. При приближении к точке минимума можно "проскачить" её и тем самым увеличивая количество итераций.

Метод градиентного спуска с дроблением шага

Метод аналогичен методу с постоянным шагом, при этом за счёт использования коэффициента дробления шага является более гибким.

Метод наискорейшего градиентного спуска

Как и отмечено в названии, этот метод действительно работает быстрее чем предыдущие методы. При этом количество итераций зависит от точности метода одномерной оптимизации - чем он точнее, тем лучше будет подобран коэффициент и тем меньше итераций потребуется для минимизации нашей функции. При этом при использовании различных одномерных методов результат оставался одинаковым или крайне близким. Метод гарантированно сходится для сильно выпуклых функций

Метод Флетчера-Ривса

На квадратичных функциях метод гарантированно сходится за n итераций (n - размерность вектора). Из-за потерь в точности вычислений на выбранном нами языке могут потребоваться дополнительные итерации для достижения заданной точности. На не квадратичных функциях сходимость не гарантируется