Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**Национальный исследовательский университет ИТМО**

**Мегафакультет трансляционных информационных технологий**

**Факультет информационных технологий и программирования**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**По дисциплине «Прикладная математика»**

**Градиентный спуск**

**Вариант 10**

Мельник Денис Александрович

Хакимов Руслан Венирович

Чечулин Лев Олегович

Выполнили

Проверил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Санкт-Петербург, 2022 г.

заданиеИзображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Немного теории

Градиент функции – вектор, координаты которого являются частными производными функции на соответствующие оси. Обозначается для точки point: grad(point).

Градиентный спуск – метод оптимизации многомерной функции, при помощи нахождения градиента функции и итеративного перехода в сторону минимума.

Переход осуществляется за счёт поиска коэффициента перед градиентом (alpha).

new point = point – alpha \* grad(point).

Так осуществляется переход к новой точке, более ближней к минимуму.

Основная проблема в том, что коэффициент alpha неизвестен, для его поиска используются различные методы:

1. Всегда используется один и тот же alpha.
2. С каждой итерацией alpha уменьшается в k раз.
3. Поиск производится при помощи одномерной минимизации функции

f(alpha) = point – alpha \* grad(point)

уже известными нам способами.

Ход работы

Для реализации градиентного спуска понадобилась функция поиска градиента в точке, для чего необходима переменная shift, которая будет обозначать сдвиг по каждому аргументу для поиска частных производных.

Изображение выглядит как текст

Автоматически созданное описание

Так же понадобятся ещё 2 переменных, отвечающих за точность: accuracyFunction и accuracyAlpha, которые отвечают за необходимую точность найденного значения функции и точность коэффициента alpha.

Рассмотрим результаты различных методов поиска величины шага на функции эллиптического параболоида: f (x, y) =

1. **Когда alpha = const**.

Если alpha = 0.1 – 33 итерации, если 0.01 – 293 итерации, а при alpha > 0.2 – улетает в космос.

Alpha = 0.1

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.100000 New point is {1.386 -0.944006 } Function decrease: 0.905504**

**2 iteration: Alpha is 0.100000 New point is {1.2948 -0.855205 } Function decrease: 0.106405**

**3 iteration: Alpha is 0.100000 New point is {1.22184 -0.872965 } Function decrease: 0.0491692**

**4 iteration: Alpha is 0.100000 New point is {1.16347 -0.869413 } Function decrease: 0.030712**

**5 iteration: Alpha is 0.100000 New point is {1.11677 -0.870123 } Function decrease: 0.0196252**

**6 iteration: Alpha is 0.100000 New point is {1.07942 -0.869981 } Function decrease: 0.0125588**

**7 iteration: Alpha is 0.100000 New point is {1.04953 -0.87001 } Function decrease: 0.00803755**

**8 iteration: Alpha is 0.100000 New point is {1.02563 -0.870004 } Function decrease: 0.00514398**

**9 iteration: Alpha is 0.100000 New point is {1.0065 -0.870005 } Function decrease: 0.00329211**

**10 iteration: Alpha is 0.100000 New point is {0.991199 -0.870005 } Function decrease: 0.00210692**

**11 iteration: Alpha is 0.100000 New point is {0.978958 -0.870005 } Function decrease: 0.0013484**

**12 iteration: Alpha is 0.100000 New point is {0.969165 -0.870005 } Function decrease: 0.000862959**

**13 iteration: Alpha is 0.100000 New point is {0.961331 -0.870005 } Function decrease: 0.000552278**

**14 iteration: Alpha is 0.100000 New point is {0.955064 -0.870005 } Function decrease: 0.000353446**

**15 iteration: Alpha is 0.100000 New point is {0.95005 -0.870005 } Function decrease: 0.000226195**

**16 iteration: Alpha is 0.100000 New point is {0.946039 -0.870005 } Function decrease: 0.000144757**

**17 iteration: Alpha is 0.100000 New point is {0.94283 -0.870005 } Function decrease: 9.2638e-05**

**18 iteration: Alpha is 0.100000 New point is {0.940263 -0.870005 } Function decrease: 5.92832e-05**

**19 iteration: Alpha is 0.100000 New point is {0.93821 -0.870005 } Function decrease: 3.79371e-05**

**20 iteration: Alpha is 0.100000 New point is {0.936567 -0.870005 } Function decrease: 2.42765e-05**

**21 iteration: Alpha is 0.100000 New point is {0.935252 -0.870005 } Function decrease: 1.55343e-05**

**22 iteration: Alpha is 0.100000 New point is {0.934201 -0.870005 } Function decrease: 9.93986e-06**

**23 iteration: Alpha is 0.100000 New point is {0.93336 -0.870005 } Function decrease: 6.35983e-06**

**24 iteration: Alpha is 0.100000 New point is {0.932687 -0.870005 } Function decrease: 4.06894e-06**

**25 iteration: Alpha is 0.100000 New point is {0.932148 -0.870005 } Function decrease: 2.60305e-06**

**26 iteration: Alpha is 0.100000 New point is {0.931718 -0.870005 } Function decrease: 1.66509e-06**

**27 iteration: Alpha is 0.100000 New point is {0.931373 -0.870005 } Function decrease: 1.06497e-06**

**28 iteration: Alpha is 0.100000 New point is {0.931098 -0.870005 } Function decrease: 6.81028e-07**

**29 iteration: Alpha is 0.100000 New point is {0.930877 -0.870005 } Function decrease: 4.35417e-07**

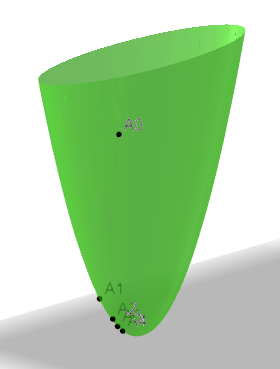
**30 iteration: Alpha is 0.100000 New point is {0.930701 -0.870005 } Function decrease: 2.78314e-07**

**31 iteration: Alpha is 0.100000 New point is {0.93056 -0.870005 } Function decrease: 1.77839e-07**

**32 iteration: Alpha is 0.100000 New point is {0.930447 -0.870005 } Function decrease: 1.13591e-07**

**33 iteration: Alpha is 0.100000 New point is {0.930356 -0.870005 } Function decrease: 7.25176e-08**

**Result: {0.930356 -0.870005 }**

Изображение выглядит как стрела

Автоматически созданное описаниеИзображение выглядит как стрела

Автоматически созданное описание Изображения первых 5 точек для разных ракурсов.

Разность функции достигла своего предела за 33 итерации, что довольно отличный показатель.

Alpha = 0.2

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.200000 New point is {1.272 -1.38801 } Function decrease: -0.580681**

**2 iteration: Alpha is 0.200000 New point is {1.1352 -0.144795 } Function decrease: -1.47066**

**3 iteration: Alpha is 0.200000 New point is {1.05312 -1.8853 } Function decrease: -3.00251**

**4 iteration: Alpha is 0.200000 New point is {1.00387 0.551406 } Function decrease: -5.92768**

**5 iteration: Alpha is 0.200000 New point is {0.974319 -2.85998 } Function decrease: -11.6343**

**6 iteration: Alpha is 0.200000 New point is {0.956589 1.91596 } Function decrease: -22.8081**

**7 iteration: Alpha is 0.200000 New point is {0.945951 -4.77036 } Function decrease: -44.7068**

**8 iteration: Alpha is 0.200000 New point is {0.939569 4.59049 } Function decrease: -87.6247**

**9 iteration: Alpha is 0.200000 New point is {0.935739 -8.5147 } Function decrease: -171.747**

**10 iteration: Alpha is 0.200000 New point is {0.933442 9.83256 } Function decrease: -336.621**

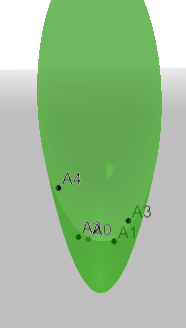
**11 iteration: Alpha is 0.200000 New point is {0.932063 -15.8536 } Function decrease: -659.78**

**12 iteration: Alpha is 0.200000 New point is {0.931236 20.107 } Function decrease: -1293.16**

**Result: {0.931236 20.107 }**

**Изображение выглядит как посуда, обеденный сервиз, чаша, фарфор

Автоматически созданное описание**

****

В итоге значение функции только увеличивается, и отдаляется от точки минимума. Причём т. к. градиент направлен в сторону минимума функции, то каждый раз меняется сторона оврага.

1. **Когда ,**

где k – коэффициент уменьшения шага.

Т. к. мы столкнулись с тем, что для больших alpha новая точка находится намного дальше от минимума, чем предыдущая, то можно попробовать этот коэффициент периодически уменьшать, например в 2 раза.

Выберем в первой итерации alpha = 0.2. Если деления не происходит (k = 1), то мы улетаем в космос, иначе при делении на разные числа, будет разный результат:

k = 2:

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.200000 New point is {1.272 -1.38801 } Function decrease: -0.580681**

**2 iteration: Alpha is 0.100000 New point is {1.2036 -0.766404 } Function decrease: 1.58773**

**3 iteration: Alpha is 0.050000 New point is {1.17624 -0.828564 } Function decrease: 0.0683146**

**4 iteration: Alpha is 0.025000 New point is {1.16393 -0.840997 } Function decrease: 0.0111661**

**5 iteration: Alpha is 0.012500 New point is {1.15808 -0.845348 } Function decrease: 0.00410272**

**6 iteration: Alpha is 0.006250 New point is {1.15523 -0.847197 } Function decrease: 0.00181892**

**7 iteration: Alpha is 0.003125 New point is {1.15382 -0.848052 } Function decrease: 0.000861764**

**8 iteration: Alpha is 0.001563 New point is {1.15312 -0.848464 } Function decrease: 0.000419998**

**9 iteration: Alpha is 0.000781 New point is {1.15277 -0.848666 } Function decrease: 0.000207395**

**10 iteration: Alpha is 0.000391 New point is {1.1526 -0.848766 } Function decrease: 0.00010306**

**11 iteration: Alpha is 0.000195 New point is {1.15251 -0.848816 } Function decrease: 5.13726e-05**

**12 iteration: Alpha is 0.000098 New point is {1.15247 -0.848841 } Function decrease: 2.56471e-05**

**13 iteration: Alpha is 0.000049 New point is {1.15244 -0.848853 } Function decrease: 1.28138e-05**

**14 iteration: Alpha is 0.000024 New point is {1.15243 -0.848859 } Function decrease: 6.40445e-06**

**15 iteration: Alpha is 0.000012 New point is {1.15243 -0.848862 } Function decrease: 3.20162e-06**

**16 iteration: Alpha is 0.000006 New point is {1.15242 -0.848864 } Function decrease: 1.60066e-06**

**17 iteration: Alpha is 0.000003 New point is {1.15242 -0.848865 } Function decrease: 8.0029e-07**

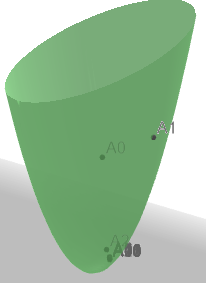
**18 iteration: Alpha is 0.000002 New point is {1.15242 -0.848865 } Function decrease: 4.00135e-07**

**19 iteration: Alpha is 0.000001 New point is {1.15242 -0.848865 } Function decrease: 2.00065e-07**

**20 iteration: Alpha is 0.000000 New point is {1.15242 -0.848865 } Function decrease: 1.00032e-07**

**21 iteration: Alpha is 0.000000 New point is {1.15242 -0.848865 } Function decrease: 5.00159e-08**

**Result: {1.15242 -0.848865 }**

**Изображение выглядит как стрела

Автоматически созданное описание**

Из-за того, что alpha слишком быстро уменьшается, значение точки не сходится к минимуму (просто не успевает дойти).

k = 1.5:

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.200000 New point is {1.272 -1.38801 } Function decrease: -0.580681**

**2 iteration: Alpha is 0.133333 New point is {1.1808 -0.559201 } Function decrease: 1.08451**

**3 iteration: Alpha is 0.088889 New point is {1.13621 -0.890725 } Function decrease: 0.597376**

**4 iteration: Alpha is 0.059259 New point is {1.11177 -0.875991 } Function decrease: 0.0118442**

**5 iteration: Alpha is 0.039506 New point is {1.09741 -0.873153 } Function decrease: 0.00517073**

**6 iteration: Alpha is 0.026337 New point is {1.08859 -0.872158 } Function decrease: 0.00290648**

**7 iteration: Alpha is 0.017558 New point is {1.08302 -0.871704 } Function decrease: 0.00174595**

**8 iteration: Alpha is 0.011706 New point is {1.07944 -0.871466 } Function decrease: 0.00108808**

**9 iteration: Alpha is 0.007804 New point is {1.0771 -0.871329 } Function decrease: 0.000693949**

**10 iteration: Alpha is 0.005202 New point is {1.07557 -0.871246 } Function decrease: 0.000449271**

**11 iteration: Alpha is 0.003468 New point is {1.07456 -0.871195 } Function decrease: 0.000293746**

**12 iteration: Alpha is 0.002312 New point is {1.0739 -0.871162 } Function decrease: 0.000193316**

**13 iteration: Alpha is 0.001541 New point is {1.07345 -0.87114 } Function decrease: 0.000127774**

**14 iteration: Alpha is 0.001028 New point is {1.07316 -0.871126 } Function decrease: 8.46961e-05**

**15 iteration: Alpha is 0.000685 New point is {1.07296 -0.871117 } Function decrease: 5.62491e-05**

**16 iteration: Alpha is 0.000457 New point is {1.07283 -0.871111 } Function decrease: 3.74042e-05**

**17 iteration: Alpha is 0.000304 New point is {1.07274 -0.871107 } Function decrease: 2.4894e-05**

**18 iteration: Alpha is 0.000203 New point is {1.07269 -0.871104 } Function decrease: 1.65772e-05**

**19 iteration: Alpha is 0.000135 New point is {1.07265 -0.871102 } Function decrease: 1.10432e-05**

**20 iteration: Alpha is 0.000090 New point is {1.07262 -0.871101 } Function decrease: 7.35843e-06**

**21 iteration: Alpha is 0.000060 New point is {1.0726 -0.8711 } Function decrease: 4.90398e-06**

**22 iteration: Alpha is 0.000040 New point is {1.07259 -0.8711 } Function decrease: 3.26859e-06**

**23 iteration: Alpha is 0.000027 New point is {1.07259 -0.8711 } Function decrease: 2.17874e-06**

**24 iteration: Alpha is 0.000018 New point is {1.07258 -0.871099 } Function decrease: 1.45235e-06**

**25 iteration: Alpha is 0.000012 New point is {1.07258 -0.871099 } Function decrease: 9.68168e-07**

**26 iteration: Alpha is 0.000008 New point is {1.07257 -0.871099 } Function decrease: 6.45417e-07**

**27 iteration: Alpha is 0.000005 New point is {1.07257 -0.871099 } Function decrease: 4.30265e-07**

**28 iteration: Alpha is 0.000004 New point is {1.07257 -0.871099 } Function decrease: 2.86838e-07**

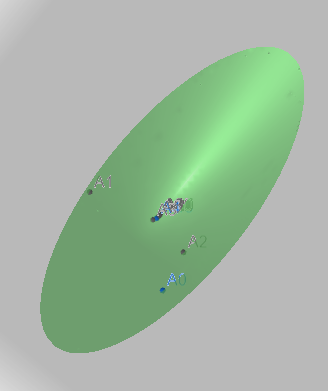
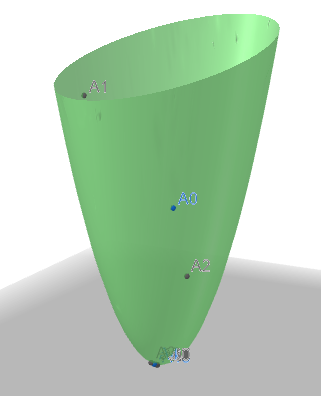
**29 iteration: Alpha is 0.000002 New point is {1.07257 -0.871099 } Function decrease: 1.91223e-07**

**30 iteration: Alpha is 0.000002 New point is {1.07257 -0.871099 } Function decrease: 1.27481e-07**

**31 iteration: Alpha is 0.000001 New point is {1.07257 -0.871099 } Function decrease: 8.49867e-08**

**Result: {1.07257 -0.871099 }**

Значение точки уже поближе к результату, но всё-равно очень в далёкой окрестности.



k = 1.1:

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.200000 New point is {1.272 -1.38801 } Function decrease: -0.580681**

**2 iteration: Alpha is 0.181818 New point is {1.14763 -0.257815 } Function decrease: -0.569007**

**3 iteration: Alpha is 0.165289 New point is {1.07569 -1.47208 } Function decrease: 0.0997883**

**4 iteration: Alpha is 0.150263 New point is {1.0319 -0.386448 } Function decrease: 0.782881**

**5 iteration: Alpha is 0.136603 New point is {1.00406 -1.17911 } Function decrease: 0.834539**

**6 iteration: Alpha is 0.124184 New point is {0.985665 -0.718478 } Function decrease: 0.437928**

**7 iteration: Alpha is 0.112895 New point is {0.973095 -0.923757 } Function decrease: 0.121655**

**8 iteration: Alpha is 0.102632 New point is {0.964249 -0.857557 } Function decrease: 0.0170943**

**9 iteration: Alpha is 0.093301 New point is {0.957857 -0.871494 } Function decrease: 0.00131253**

**10 iteration: Alpha is 0.084820 New point is {0.95313 -0.869978 } Function decrease: 0.000254376**

**11 iteration: Alpha is 0.077109 New point is {0.949562 -0.870003 } Function decrease: 0.000152325**

**12 iteration: Alpha is 0.070099 New point is {0.946819 -0.870005 } Function decrease: 9.98057e-05**

**13 iteration: Alpha is 0.063726 New point is {0.944675 -0.870005 } Function decrease: 6.75317e-05**

**14 iteration: Alpha is 0.057933 New point is {0.942974 -0.870005 } Function decrease: 4.70276e-05**

**15 iteration: Alpha is 0.052666 New point is {0.941607 -0.870005 } Function decrease: 3.36046e-05**

**16 iteration: Alpha is 0.047878 New point is {0.940495 -0.870005 } Function decrease: 2.45753e-05**

**17 iteration: Alpha is 0.043526 New point is {0.939581 -0.870005 } Function decrease: 1.83501e-05**

**18 iteration: Alpha is 0.039569 New point is {0.938822 -0.870005 } Function decrease: 1.39608e-05**

**19 iteration: Alpha is 0.035972 New point is {0.938187 -0.870005 } Function decrease: 1.08022e-05**

**20 iteration: Alpha is 0.032702 New point is {0.937651 -0.870005 } Function decrease: 8.48632e-06**

**21 iteration: Alpha is 0.029729 New point is {0.937196 -0.870005 } Function decrease: 6.75911e-06**

**22 iteration: Alpha is 0.027026 New point is {0.936807 -0.870005 } Function decrease: 5.4506e-06**

**23 iteration: Alpha is 0.024569 New point is {0.936472 -0.870005 } Function decrease: 4.44492e-06**

**24 iteration: Alpha is 0.022336 New point is {0.936183 -0.870005 } Function decrease: 3.66171e-06**

**25 iteration: Alpha is 0.020305 New point is {0.935932 -0.870005 } Function decrease: 3.04426e-06**

**26 iteration: Alpha is 0.018459 New point is {0.935712 -0.870005 } Function decrease: 2.55201e-06**

**27 iteration: Alpha is 0.016781 New point is {0.935521 -0.870005 } Function decrease: 2.15548e-06**

**28 iteration: Alpha is 0.015256 New point is {0.935352 -0.870005 } Function decrease: 1.83299e-06**

**29 iteration: Alpha is 0.013869 New point is {0.935203 -0.870005 } Function decrease: 1.56838e-06**

**30 iteration: Alpha is 0.012608 New point is {0.935072 -0.870005 } Function decrease: 1.34949e-06**

**31 iteration: Alpha is 0.011462 New point is {0.934956 -0.870005 } Function decrease: 1.16705e-06**

**32 iteration: Alpha is 0.010420 New point is {0.934852 -0.870005 } Function decrease: 1.01391e-06**

**33 iteration: Alpha is 0.009472 New point is {0.93476 -0.870005 } Function decrease: 8.8455e-07**

**34 iteration: Alpha is 0.008611 New point is {0.934678 -0.870005 } Function decrease: 7.74614e-07**

**35 iteration: Alpha is 0.007829 New point is {0.934605 -0.870005 } Function decrease: 6.80672e-07**

**36 iteration: Alpha is 0.007117 New point is {0.934539 -0.870005 } Function decrease: 5.99988e-07**

**37 iteration: Alpha is 0.006470 New point is {0.93448 -0.870005 } Function decrease: 5.30364e-07**

**38 iteration: Alpha is 0.005882 New point is {0.934428 -0.870005 } Function decrease: 4.70024e-07**

**39 iteration: Alpha is 0.005347 New point is {0.93438 -0.870005 } Function decrease: 4.1752e-07**

**40 iteration: Alpha is 0.004861 New point is {0.934338 -0.870005 } Function decrease: 3.71666e-07**

**41 iteration: Alpha is 0.004419 New point is {0.934299 -0.870005 } Function decrease: 3.31484e-07**

**42 iteration: Alpha is 0.004017 New point is {0.934265 -0.870005 } Function decrease: 2.96163e-07**

**43 iteration: Alpha is 0.003652 New point is {0.934233 -0.870005 } Function decrease: 2.65025e-07**

**44 iteration: Alpha is 0.003320 New point is {0.934205 -0.870005 } Function decrease: 2.37502e-07**

**45 iteration: Alpha is 0.003018 New point is {0.93418 -0.870005 } Function decrease: 2.13116e-07**

**46 iteration: Alpha is 0.002744 New point is {0.934157 -0.870005 } Function decrease: 1.91461e-07**

**47 iteration: Alpha is 0.002494 New point is {0.934136 -0.870005 } Function decrease: 1.72192e-07**

**48 iteration: Alpha is 0.002268 New point is {0.934117 -0.870005 } Function decrease: 1.55015e-07**

**49 iteration: Alpha is 0.002061 New point is {0.9341 -0.870005 } Function decrease: 1.39675e-07**

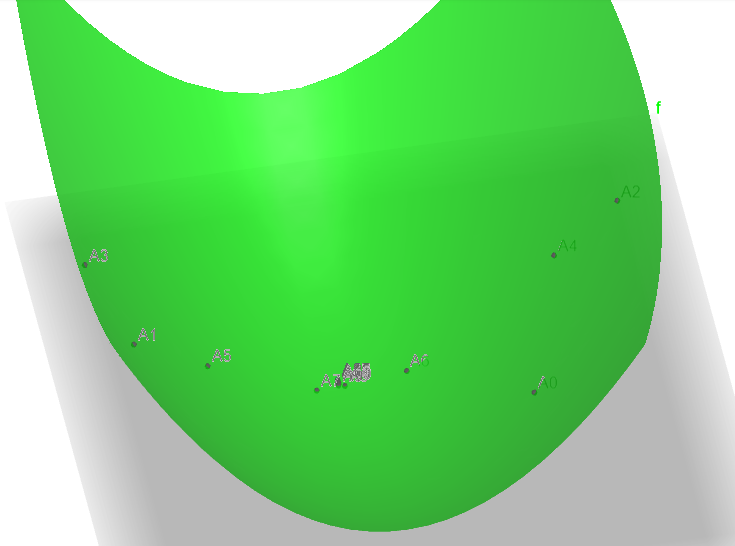
**50 iteration: Alpha is 0.001874 New point is {0.934085 -0.870005 } Function decrease: 1.25956e-07**

**51 iteration: Alpha is 0.001704 New point is {0.934071 -0.870005 } Function decrease: 1.13667e-07**

**52 iteration: Alpha is 0.001549 New point is {0.934058 -0.870005 } Function decrease: 1.02647e-07**

**53 iteration: Alpha is 0.001408 New point is {0.934047 -0.870005 } Function decrease: 9.27505e-08**

**Result: {0.934047 -0.870005 }**

****

Несмотря на то, что итераций получилось слишком много, точность значения очень радует и относительная погрешность найденной точки составляет меньше 1%.

3**. Когда поиск alpha производится через метод золотого сечения.**

Каждую итерацию будем производить минимизацию одномерной функции

f(alpha) = point – alpha \* grad(point)

При помощи метода золотого сечения.

Пронаблюдаем за кол-вом итераций (вызовов этого метода).

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.087729 New point is {1.39999 -0.88952 } Function decrease: 0.923125**

**2 iteration: Alpha is 0.387142 New point is {1.03608 -0.798859 } Function decrease: 0.181556**

**3 iteration: Alpha is 0.087715 New point is {1.01747 -0.873746 } Function decrease: 0.0338842**

**4 iteration: Alpha is 0.381827 New point is {0.950669 -0.856606 } Function decrease: 0.00623149**

**5 iteration: Alpha is 0.087855 New point is {0.947037 -0.870732 } Function decrease: 0.00121012**

**6 iteration: Alpha is 0.382407 New point is {0.934003 -0.867396 } Function decrease: 0.000236738**

**7 iteration: Alpha is 0.087748 New point is {0.9333 -0.870143 } Function decrease: 4.57103e-05**

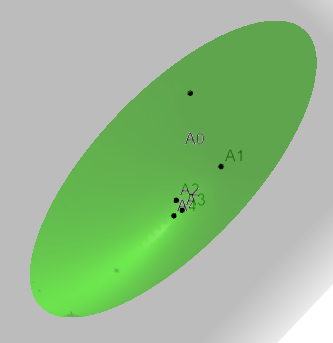
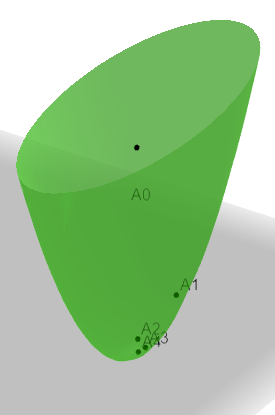
**8 iteration: Alpha is 0.386240 New point is {0.930747 -0.869503 } Function decrease: 8.96781e-06**

**9 iteration: Alpha is 0.086721 New point is {0.930616 -0.870025 } Function decrease: 1.65843e-06**

**10 iteration: Alpha is 0.422001 New point is {0.930092 -0.869922 } Function decrease: 3.38933e-07**

**11 iteration: Alpha is 0.081002 New point is {0.930076 -0.870003 } Function decrease: 3.9103e-08**

**Result: {0.930076 -0.870003 }**

****

Всего за 11 вызововметода золотого сечения мы достигли очень точной минимизации.

4.  **Когда поиск alpha производится через метод Фибоначчи.**

Тут всё зависит от числа итераций n в самом методе Фибоначчи. Чем их больше, тем точнее ищется точка.

n = 10:

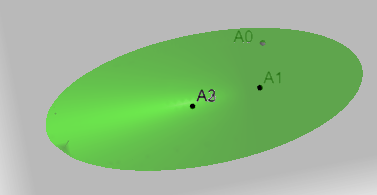
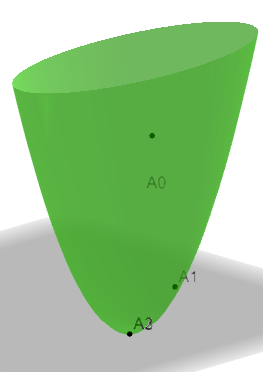
**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.081899 New point is {1.40663 -0.863636 } Function decrease: 0.918877**

**2 iteration: Alpha is 0.610966 New point is {0.824213 -0.910329 } Function decrease: 0.206474**

**3 iteration: Alpha is 0.000000 New point is {0.824213 -0.910329 } Function decrease: 0**

**Result: {0.824213 -0.910329 }**

****

n = 50:

**First point is {1.5 -0.5 }**

**1 iteration: Alpha is 0.087860 New point is {1.39984 -0.890106 } Function decrease: 0.923127**

**2 iteration: Alpha is 0.381949 New point is {1.04093 -0.797977 } Function decrease: 0.179745**

**3 iteration: Alpha is 0.087853 New point is {1.02143 -0.873912 } Function decrease: 0.0349766**

**4 iteration: Alpha is 0.382188 New point is {0.95154 -0.855995 } Function decrease: 0.00681121**

**5 iteration: Alpha is 0.087815 New point is {0.947756 -0.870759 } Function decrease: 0.0013221**

**6 iteration: Alpha is 0.383413 New point is {0.934136 -0.867292 } Function decrease: 0.000257617**

**7 iteration: Alpha is 0.087621 New point is {0.933411 -0.870145 } Function decrease: 4.9364e-05**

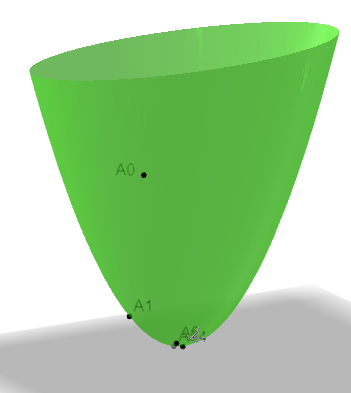
**8 iteration: Alpha is 0.389773 New point is {0.930748 -0.869492 } Function decrease: 9.64825e-06**

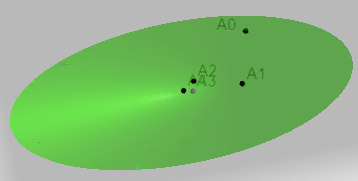
**9 iteration: Alpha is 0.086606 New point is {0.930618 -0.870025 } Function decrease: 1.72495e-06**

**10 iteration: Alpha is 0.423583 New point is {0.93009 -0.869923 } Function decrease: 3.41197e-07**

**11 iteration: Alpha is 0.080703 New point is {0.930075 -0.870002 } Function decrease: 3.83718e-08**

**Result: {0.930075 -0.870002 }**

****

****

Это всё происходит, потому что отрезок для поиска alpha располагается от 0 до максимального возможного значения такого, что new point не выйдет за пределы изначально заданного промежутка. Именно поэтому такой метод и не задаёт особо большую точность с каждым новым его вызовом.

5. **Метод сопряжённых градиентов.**

Метод Флетчера-Ривса.

Это ускоренный метод нахождения минимума многомерной функции. Достигается ускорения, за счёт меньшего кол-ва вызовов метода Золотого Сечения. Достигается это засчёт задания определённого направления, которое на каждой итерации будет содержать усреднённое значение всех градиентов в предыдущих точках, засчёт чего мы будем иметь более точное направление для поиска минимума. Обозначим это направление за S.

Изначально, S0 = - grad(point0)

В дальнейшем все итерации будут соответствовать данным рекуррентным формулам:

Fletcher-Reeves with GoldenAlpha:

**First point is {1.5 -0.5 }**

**First stream is {-1.14001 -4.44006 }**

**Alpha is 0.087729 Omega is 0.044657 New point is {1.39999 -0.88952 } New stream is {-0.990897 0.0358996 } Function decrease: 0.923125**

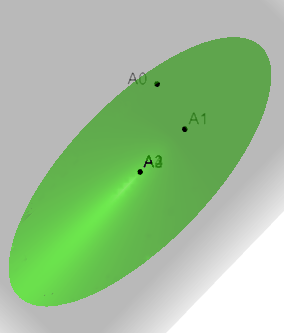
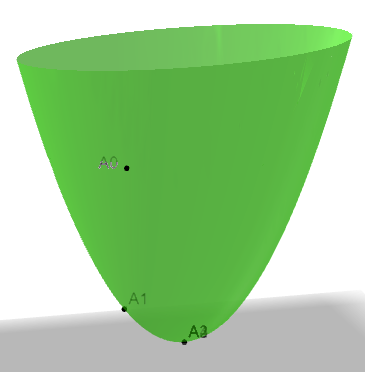
**Alpha is 0.474779 Omega is 0.000938 New point is {0.929531 -0.872476 } New stream is {-1.60342e-06 0.0296814 } Function decrease: 0.223138**

**Alpha is 0.083279 Omega is 0.000978 New point is {0.929531 -0.870004 } New stream is {0.000927715 1.45887e-05 } Function decrease: 3.67727e-05**

**Alpha is 0.504907 Omega is 0.012382 New point is {0.93 -0.869996 } New stream is {2.38364e-06 -0.000102663 } Function decrease: 2.19838e-07**

**Alpha is 0.034771 Omega is 0.345862 New point is {0.93 -0.87 } New stream is {-8.44512e-06 -9.55143e-05 } Function decrease: 7.65493e-11**

**Result: {0.93 -0.87 }**

****

То есть понадобилось в 2 раза меньше операций (5 вместо 11), чем в случае без направлений, а точность способа ни чуть ни теряется.

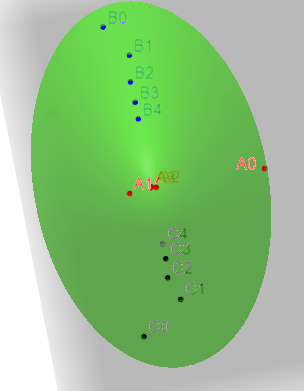
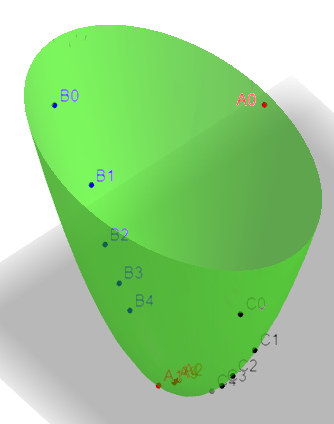
6.  **Когда у нас начало в трёх разных точках.**

Рассмотрим метод с постоянной alpha = 0.1 для разных точек.

Точка A(1; 0)

Точка B(-1; -1)

Точка C(2.673; -1.144)



Таким образом, можно сделать вывод, что для каждой ситуации подходит разная alpha. Т. к. точки B и C с более пологим градиентом находятся дальше от центра, чем A, соответственно, им нужно большее значение alpha для оптимального схождения.

Вывод

В результате выполнения работы был изучен градиентный спуск с использованием различных методов одномерной оптимизации и подбора шага. Изучены зависимости от выбранной точки, от характера функции, от выбранного метода. Было произведено сравнение некоторых характеристик методов, например, кол-во итераций (шагов) при разных условиях. Был изучен метод сопряжённых градиентов, реализовав который, мы поняли, что он значительно ускорил все предыдущие известные нам методы сходимости. Так же мы сделали визуализацию градиентного спуска и написали код с автоматическим его исчислением. После сравнения всех составляющих, мы можем уверенно считать, что градиентный спуск изучен со всех сторон.