|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Методы оптимизации» | | |
| **Метод штрафных функций** | | |
|  | | |
|  | Бригада 2 | Буданцев Дмитрий |
| Группа ПМ-13 | голубь Андрей |
| Вариант 2 |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Филиппова Елена Владимировна |
|  | лемешко борис юрьевич |
| Новосибирск,2024 | | |

1. **Цель работы**:

Ознакомиться с методами штрафных функций при решении задач нелинейного программирования. Изучить типы штрафных и барьерных функций, их особенности, способы и области применения, влияние штрафных функций на сходимость алгоритмов, зависимость точности решения задачи нелинейного программирования от величины коэффициента штрафа.

1. **Задания:**
   1. Применяя методы поиска минимума ***0-го порядка***, реализовать программу для решения задачи нелинейного программирования с использованием ***метода штрафных функций***
   2. Исследовать сходимость метода штрафных функций в зависимости
      1. от выбора штрафных функций,
      2. начальной величины коэффициента штрафа,
      3. стратегии изменения коэффициента штрафа,
      4. начальной точки,
      5. задаваемой точности .

Сформулировать выводы

* 1. Применяя методы поиска минимума ***0-го порядка***, реализовать программу для решения задачи нелинейного программирования с ограничением типа неравенства (*только задача а*) с использованием **метода барьерных функций**
  2. Исследовать сходимость метода барьерных функций (*только задача а*) в зависимости
     1. от выбора барьерных функций,
     2. начальной величины коэффициента штрафа,
     3. стратегии изменения коэффициента штрафа,
     4. начального приближения,
     5. задаваемой точности .

Сформулировать выводы

**Первая задача:**  

**Вторая задача**: 

**Метод минимизации 0-го порядка:** Симплексный метод Нелдера–Мида,

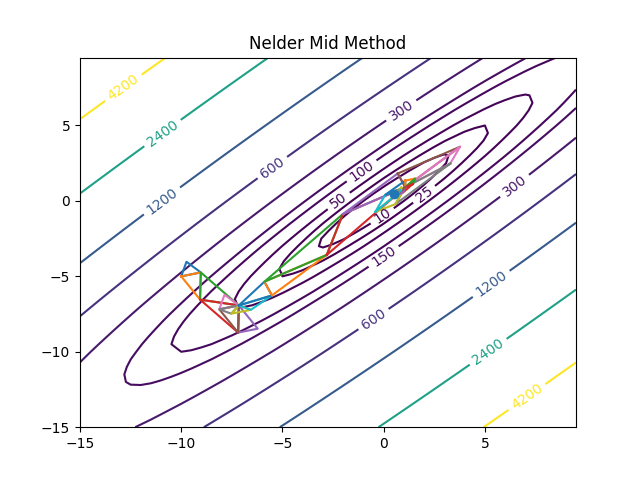
с коэффициентами 

1. **Исследование**
   1. **Метод штрафных функций**
      1. *Исследовать сходимость метода штрафных функций в зависимости от выбора штрафных функций*
         1. **Первая задача**

*Начальные условия: *

*Штрафная функция: *

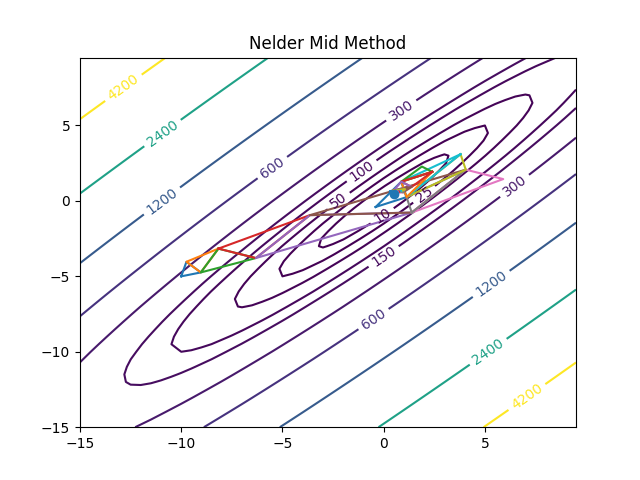
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ -10.000 -5.000 ] | 323 | 0 | 3 | 48 |
| **1** | [ -9.034 -4.741 ] | 251.0938841 | 71.90611589 | 5 | 44.32576539 |
| **2** | [ -9.069 -6.544 ] | 156.418747 | 94.67513714 | 7 | 49.83711731 |
| **3** | [ -7.154 -6.927 ] | 93.74593053 | 62.67281644 | 8 | 45.24432404 |
| **4** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 10 | 45.24432404 |
| **5** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 11 | 45.24432404 |
| **6** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 13 | 45.24432404 |
| **7** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 15 | 45.24432404 |
| **8** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 16 | 45.24432404 |
| **9** | [ -7.154 -6.927 ] | 93.74593053 | 0 | 18 | 45.24432404 |
| **10** | [ -5.520 -6.265 ] | 83.16102444 | 10.58490609 | 20 | 38.35513414 |
| **11** | [ -5.893 -5.356 ] | 68.31596585 | 14.84505859 | 22 | 36.74765649 |
| **12** | [ -2.811 -3.578 ] | 40.85519534 | 27.46077051 | 24 | 22.16553787 |
| **13** | [ -2.015 -0.871 ] | 25.50725048 | 15.34794486 | 26 | 11.65952327 |
| **14** | [ 1.067 0.907 ] | 1.078072994 | 24.42917749 | 28 | 0 |
| **15** | [ 1.067 0.907 ] | 1.078072994 | 0 | 29 | 0 |
| **16** | [ 1.067 0.907 ] | 1.078072994 | 0 | 31 | 0 |
| **17** | [ 1.067 0.907 ] | 1.078072994 | 0 | 32 | 0 |
| **18** | [ 1.067 0.907 ] | 1.078072994 | 0 | 34 | 0 |
| **19** | [ 1.067 0.907 ] | 1.078072994 | 0 | 35 | 0 |
| **20** | [ 1.067 0.907 ] | 1.078072994 | 0 | 37 | 0 |
| **21** | [ 1.067 0.907 ] | 1.078072994 | 0 | 39 | 0 |
| **22** | [ 1.067 0.907 ] | 1.078072994 | 0 | 41 | 0 |
| **23** | [ 1.067 0.907 ] | 1.078072994 | 0 | 42 | 0 |
| **24** | [ 1.067 0.907 ] | 1.078072994 | 0 | 44 | 0 |
| **25** | [ 0.575 0.329 ] | 1.000446282 | 0.077626712 | 46 | 0.289220327 |
| **26** | [ 0.697 0.693 ] | 0.480089012 | 0.52035727 | 48 | 0 |
| **27** | [ 0.697 0.693 ] | 0.480089012 | 0 | 49 | 0 |
| **28** | [ 0.697 0.693 ] | 0.480089012 | 0 | 51 | 0 |
| **29** | [ 0.697 0.693 ] | 0.480089012 | 0 | 52 | 0 |
| **30** | [ 0.697 0.693 ] | 0.480089012 | 0 | 54 | 0 |
| **31** | [ 0.520 0.499 ] | 0.253244342 | 0.22684467 | 56 | 0 |
| **32** | [ 0.520 0.499 ] | 0.253244342 | 0 | 58 | 0 |
| **33** | [ 0.520 0.499 ] | 0.253244342 | 0 | 60 | 0 |
| **34** | [ 0.520 0.499 ] | 0.253244342 | 0 | 62 | 0 |
| **35** | [ 0.520 0.499 ] | 0.253244342 | 0 | 64 | 0 |
| **36** | [ 0.520 0.499 ] | 0.253244342 | 0 | 65 | 0 |
| **37** | [ 0.520 0.499 ] | 0.253244342 | 0 | 67 | 0 |
| **38** | [ 0.520 0.499 ] | 0.253244342 | 0 | 69 | 0 |
| **39** | [ 0.514 0.489 ] | 0.245165824 | 0.008078518 | 69 | 0 |
| **=** [0.51386443 0.48871191] **=** 0.2451658239081109 | | | | | |

**

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ -10.000 -5.000 ] | 262419 | 0 | 3 | 262144 |
| **1** | [ -9.034 -4.741 ] | 223754.3282 | 38664.67183 | 5 | 223547.5601 |
| **2** | [ -8.163 -3.163 ] | 155830.684 | 67923.64418 | 7 | 155570.6802 |
| **3** | [ -6.313 -3.788 ] | 126268.3514 | 29562.3326 | 9 | 126190.2402 |
| **4** | [ -3.646 -0.944 ] | 32068.15417 | 94200.19722 | 11 | 31994.26046 |
| **5** | [ 1.387 -0.772 ] | 198.619153 | 31869.53502 | 13 | 151.3906916 |
| **6** | [ 4.055 2.072 ] | 43.60371393 | 155.0154391 | 15 | 0 |
| **7** | [ 4.055 2.072 ] | 43.60371393 | 0 | 17 | 0 |
| **8** | [ 1.129 0.252 ] | 7.769056234 | 35.8346577 | 18 | 0 |
| **9** | [ 1.129 0.252 ] | 7.769056234 | 0 | 20 | 0 |
| **10** | [ 0.871 1.275 ] | 3.256515748 | 4.512540486 | 22 | 0 |
| **11** | [ 0.871 1.275 ] | 3.256515748 | 0 | 23 | 0 |
| **12** | [ 0.871 1.275 ] | 3.256515748 | 0 | 25 | 0 |
| **13** | [ 0.871 1.275 ] | 3.256515748 | 0 | 27 | 0 |
| **14** | [ 1.382 0.927 ] | 2.931446988 | 0.325068759 | 29 | 0 |
| **15** | [ 0.491 0.687 ] | 0.855672504 | 2.075774485 | 31 | 0 |
| **16** | [ 0.491 0.687 ] | 0.855672504 | 0 | 33 | 0 |
| **17** | [ 0.491 0.687 ] | 0.855672504 | 0 | 34 | 0 |
| **18** | [ 0.491 0.687 ] | 0.855672504 | 0 | 36 | 0 |
| **19** | [ 0.491 0.687 ] | 0.855672504 | 0 | 38 | 0 |
| **20** | [ 0.494 0.615 ] | 0.52723309 | 0.328439414 | 40 | 0 |
| **21** | [ 0.494 0.615 ] | 0.52723309 | 0 | 42 | 0 |
| **22** | [ 0.682 0.652 ] | 0.433860187 | 0.093372904 | 43 | 0 |
| **23** | [ 0.682 0.652 ] | 0.433860187 | 0 | 45 | 0 |
| **24** | [ 0.682 0.652 ] | 0.433860187 | 0 | 47 | 0 |
| **25** | [ 0.588 0.616 ] | 0.386762954 | 0.047097233 | 49 | 0 |
| **26** | [ 0.588 0.616 ] | 0.386762954 | 0 | 50 | 0 |
| **27** | [ 0.588 0.616 ] | 0.386762954 | 0 | 52 | 0 |
| **28** | [ 0.509 0.531 ] | 0.286510908 | 0.100252046 | 54 | 0 |
| **29** | [ 0.509 0.531 ] | 0.286510908 | 0 | 56 | 0 |
| **30** | [ 0.509 0.531 ] | 0.286510908 | 0 | 57 | 0 |
| **31** | [ 0.509 0.531 ] | 0.286510908 | 0 | 58 | 0 |
| **32** | [ 0.509 0.531 ] | 0.286510908 | 0 | 60 | 0 |
| **33** | [ 0.524 0.505 ] | 0.258853973 | 0.027656936 | 62 | 0 |
| **34** | [ 0.524 0.505 ] | 0.258853973 | 0 | 64 | 0 |
| **35** | [ 0.510 0.488 ] | 0.246116046 | 0.012737926 | 66 | 0.002965985 |
| **36** | [ 0.510 0.488 ] | 0.246116046 | 0 | 68 | 0.002965985 |
| **37** | [ 0.517 0.485 ] | 0.245245588 | 0.000870459 | 70 | 0 |
| **38** | [ 0.517 0.485 ] | 0.245245588 | 0 | 72 | 0 |
| **=** [0.51660712 0.48508997] **=** 0.24524558795329576 | | | | | |

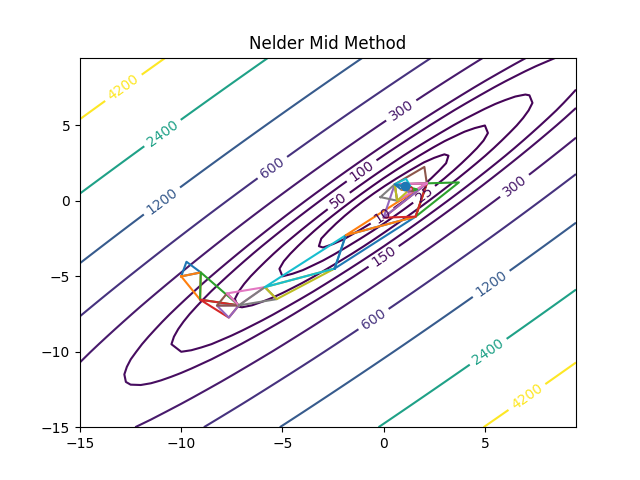
**

* + - 1. **Вторая задача**

*Начальные условия: *

*Штрафная функция: *

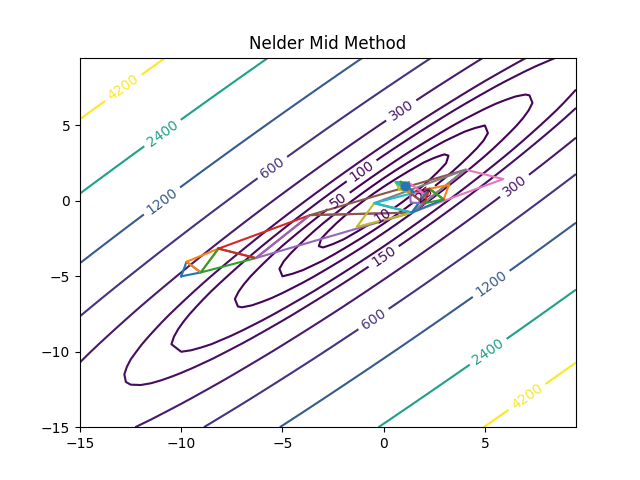
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ -10.000 -5.000 ] | 819 | 0 | 3 | 544 |
| **1** | [ -9.034 -4.741 ] | 711.5762828 | 107.4237172 | 5 | 504.8081641 |
| **2** | [ -9.069 -6.544 ] | 670.1775476 | 41.39873524 | 7 | 563.5959179 |
| **3** | [ -7.154 -6.927 ] | 563.1077296 | 107.069818 | 9 | 514.6061231 |
| **4** | [ -7.154 -6.927 ] | 563.1077296 | 0 | 11 | 514.6061231 |
| **5** | [ -7.154 -6.927 ] | 563.1077296 | 0 | 12 | 514.6061231 |
| **6** | [ -7.154 -6.927 ] | 563.1077296 | 0 | 14 | 514.6061231 |
| **7** | [ -5.870 -5.714 ] | 467.5873509 | 95.52037871 | 15 | 434.6915202 |
| **8** | [ -5.870 -5.714 ] | 467.5873509 | 0 | 17 | 434.6915202 |
| **9** | [ -2.423 -4.483 ] | 347.4970511 | 120.0902998 | 19 | 284.9664597 |
| **10** | [ -1.870 -2.275 ] | 203.4522122 | 144.0448389 | 21 | 196.6317359 |
| **11** | [ 1.577 -1.043 ] | 116.6775219 | 86.77469029 | 23 | 46.90667532 |
| **12** | [ 2.130 1.164 ] | 52.11910998 | 64.55841188 | 25 | 41.42804853 |
| **13** | [ 2.130 1.164 ] | 52.11910998 | 0 | 26 | 41.42804853 |
| **14** | [ 2.130 1.164 ] | 52.11910998 | 0 | 28 | 41.42804853 |
| **15** | [ 0.545 1.100 ] | 15.64176794 | 36.47734204 | 30 | 11.35080078 |
| **16** | [ 0.545 1.100 ] | 15.64176794 | 0 | 31 | 11.35080078 |
| **17** | [ 0.545 1.100 ] | 15.64176794 | 0 | 33 | 11.35080078 |
| **18** | [ 0.545 1.100 ] | 15.64176794 | 0 | 35 | 11.35080078 |
| **19** | [ 1.368 0.860 ] | 10.61088276 | 5.030885177 | 37 | 7.295461475 |
| **20** | [ 1.368 0.860 ] | 10.61088276 | 0 | 39 | 7.295461475 |
| **21** | [ 1.368 0.860 ] | 10.61088276 | 0 | 41 | 7.295461475 |
| **22** | [ 1.416 0.649 ] | 8.376252159 | 2.234630602 | 43 | 2.07078537 |
| **23** | [ 1.416 0.649 ] | 8.376252159 | 0 | 45 | 2.07078537 |
| **24** | [ 1.138 0.746 ] | 5.797498764 | 2.578753395 | 47 | 3.709245705 |
| **25** | [ 1.138 0.746 ] | 5.797498764 | 0 | 48 | 3.709245705 |
| **26** | [ 1.138 0.746 ] | 5.797498764 | 0 | 50 | 3.709245705 |
| **27** | [ 1.044 0.877 ] | 3.590875488 | 2.206623276 | 52 | 2.541915421 |
| **28** | [ 1.044 0.877 ] | 3.590875488 | 0 | 54 | 2.541915421 |
| **29** | [ 1.207 0.795 ] | 2.380929544 | 1.209945944 | 55 | 0.056267545 |
| **30** | [ 1.207 0.795 ] | 2.380929544 | 0 | 57 | 0.056267545 |
| **31** | [ 1.207 0.795 ] | 2.380929544 | 0 | 59 | 0.056267545 |
| **32** | [ 1.102 0.868 ] | 2.250381188 | 0.130548356 | 61 | 0.950991367 |
| **33** | [ 1.102 0.868 ] | 2.250381188 | 0 | 63 | 0.950991367 |
| **34** | [ 1.134 0.879 ] | 1.810381136 | 0.440000051 | 65 | 0.388117959 |
| **35** | [ 1.029 0.952 ] | 1.584503562 | 0.225877574 | 66 | 0.619140953 |
| **36** | [ 1.029 0.952 ] | 1.584503562 | 0 | 68 | 0.619140953 |
| **37** | [ 0.956 1.035 ] | 1.421414127 | 0.163089435 | 70 | 0.28729054 |
| **38** | [ 0.956 1.035 ] | 1.421414127 | 0 | 72 | 0.28729054 |
| **39** | [ 1.027 0.978 ] | 1.113056675 | 0.308357452 | 74 | 0.133376313 |
| **40** | [ 1.027 0.978 ] | 1.113056675 | 0 | 76 | 0.133376313 |
| **41** | [ 1.027 0.978 ] | 1.113056675 | 0 | 77 | 0.133376313 |
| **42** | [ 1.027 0.978 ] | 1.113056675 | 0 | 79 | 0.133376313 |
| **43** | [ 1.032 0.964 ] | 1.09801803 | 0.015038645 | 81 | 0.122525984 |
| **44** | [ 1.055 0.946 ] | 1.052445758 | 0.045572272 | 83 | 0.039021492 |
| **45** | [ 1.055 0.946 ] | 1.052445758 | 0 | 85 | 0.039021492 |
| **46** | [ 1.035 0.966 ] | 1.026763691 | 0.025682067 | 87 | 0.045812034 |
| **47** | [ 1.035 0.966 ] | 1.026763691 | 0 | 89 | 0.045812034 |
| **48** | [ 1.038 0.960 ] | 1.023205443 | 0.003558249 | 91 | 0.04005461 |
| **49** | [ 1.018 0.981 ] | 1.009082389 | 0.014123053 | 93 | 0.033264069 |
| **50** | [ 1.018 0.981 ] | 1.009082389 | 0 | 95 | 0.033264069 |
| **51** | [ 1.032 0.968 ] | 0.982447498 | 0.026634891 | 96 | 0.004576347 |
| **52** | [ 1.032 0.968 ] | 0.982447498 | 0 | 98 | 0.004576347 |
| **53** | [ 1.032 0.968 ] | 0.982447498 | 0 | 99 | 0.004576347 |
| **54** | [ 1.032 0.968 ] | 0.982447498 | 0 | 101 | 0.004576347 |
| **=** [1.03165744 0.96848557 **=** 0.9824474978644 | | | | | |

****

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ -10.000 -5.000 ] | 36111 | 0 | 3 | 35836 |
| **1** | [ -9.034 -4.741 ] | 31065.24374 | 5045.756259 | 5 | 30858.47562 |
| **2** | [ -8.163 -3.163 ] | 22279.43069 | 8785.813047 | 7 | 22019.42687 |
| **3** | [ -6.313 -3.788 ] | 18236.01366 | 4043.417035 | 9 | 18157.90249 |
| **4** | [ -3.646 -0.944 ] | 5458.4365 | 12777.57716 | 11 | 5384.542789 |
| **5** | [ 1.387 -0.772 ] | 284.9176447 | 5173.518855 | 12 | 237.6891832 |
| **6** | [ 1.387 -0.772 ] | 284.9176447 | 0 | 14 | 237.6891832 |
| **7** | [ 1.387 -0.772 ] | 284.9176447 | 0 | 15 | 237.6891832 |
| **8** | [ 1.387 -0.772 ] | 284.9176447 | 0 | 17 | 237.6891832 |
| **9** | [ 1.387 -0.772 ] | 284.9176447 | 0 | 19 | 237.6891832 |
| **10** | [ 2.259 0.806 ] | 162.3837647 | 122.5338799 | 21 | 140.6404276 |
| **11** | [ 2.259 0.806 ] | 162.3837647 | 0 | 23 | 140.6404276 |
| **12** | [ 2.259 0.806 ] | 162.3837647 | 0 | 25 | 140.6404276 |
| **13** | [ 2.000 -0.160 ] | 49.8227889 | 112.5609758 | 27 | 3.16480543 |
| **14** | [ 1.293 0.547 ] | 9.018426629 | 40.80436227 | 29 | 3.16480543 |
| **15** | [ 1.293 0.547 ] | 9.018426629 | 0 | 30 | 3.16480543 |
| **16** | [ 1.293 0.547 ] | 9.018426629 | 0 | 31 | 3.16480543 |
| **17** | [ 1.293 0.547 ] | 9.018426629 | 0 | 32 | 3.16480543 |
| **18** | [ 1.293 0.547 ] | 9.018426629 | 0 | 34 | 3.16480543 |
| **19** | [ 1.293 0.547 ] | 9.018426629 | 0 | 36 | 3.16480543 |
| **20** | [ 1.092 1.054 ] | 3.784478581 | 5.233948047 | 38 | 2.658710983 |
| **21** | [ 1.092 1.054 ] | 3.784478581 | 0 | 40 | 2.658710983 |
| **22** | [ 0.889 1.028 ] | 2.107004773 | 1.677473808 | 42 | 0.858594318 |
| **23** | [ 0.889 1.028 ] | 2.107004773 | 0 | 43 | 0.858594318 |
| **24** | [ 0.889 1.028 ] | 2.107004773 | 0 | 45 | 0.858594318 |
| **25** | [ 0.889 1.028 ] | 2.107004773 | 0 | 47 | 0.858594318 |
| **26** | [ 1.054 0.983 ] | 1.180453427 | 0.926551346 | 49 | 0.164227591 |
| **27** | [ 1.054 0.983 ] | 1.180453427 | 0 | 51 | 0.164227591 |
| **28** | [ 1.054 0.983 ] | 1.180453427 | 0 | 53 | 0.164227591 |
| **29** | [ 1.054 0.983 ] | 1.180453427 | 0 | 54 | 0.164227591 |
| **30** | [ 1.054 0.983 ] | 1.180453427 | 0 | 56 | 0.164227591 |
| **31** | [ 1.003 0.991 ] | 0.9887174 | 0.191736027 | 58 | 0.004452308 |
| **32** | [ 1.003 0.991 ] | 0.9887174 | 0 | 60 | 0.004452308 |
| **33** | [ 1.003 0.991 ] | 0.9887174 | 0 | 62 | 0.004452308 |
| **34** | [ 1.003 0.991 ] | 0.9887174 | 0 | 64 | 0.004452308 |
| **35** | [ 1.037 0.963 ] | 0.98266727 | 0.006050131 | 64 | 4.16E-06 |
| **=** [1.03743299 0.96275013]**=** 0.9826672695079623 | | | | | |

****

**Вывод:** В результате по полученному исследованию можно сделать вывод, что на данных функциях, при каждой штрафной функции метод сошёлся с указанной точностью. Так же можно заметить, изменение траектории в зависимости от смены степени штрафных функций. При увеличении степени, метод в начальный момент, устремляется к минимальной точке. Это вызвано более большим штрафом в начальный момент времени.

* + 1. *Исследовать сходимость метода штрафных функций в зависимости* коэффициента штрафа
       1. **Первая задача**

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Iteration | *Выполнение условия* |  |
| *1* | [0.4975078 0.4742071] | 0.23030 | 37 | *Нет* | *0.0136* |
| *2* | [0.46448853 0.53250941] | 0.32983 | 46 | *Нет* | *0.08593* |
| *4* | [0.52276759 0.51257255] | 0.26377 | 40 | *Да* | *0.01986* |
| *8* | [0.51208788 0.49052073] | 0.24526 | 43 | *Да* | *0.00135* |

* + - 1. ***Вторая задача***

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Iteration | *Выполнение условия* |  |
| *1* | [0.98581011 0.94233614] | 0.90689 | 36 | *Нет* | *0.06871* |
| *2* | [1.02476199 0.97048629] | 0.97130 | 37 | *Нет* | *0.00430* |
| *4* | [1.0238093 0.97475916] | 0.97421 | 45 | *Да* | *0.00139* |

* + 1. *Исследовать сходимость метода штрафных функций в зависимости начальной точки*
       1. **Первая задача**

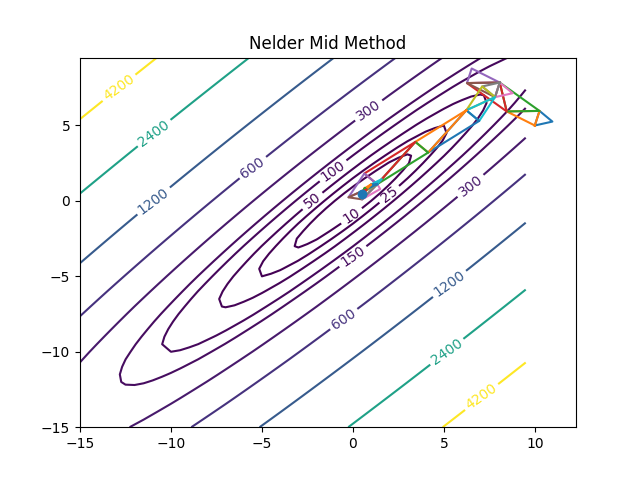
*До этого уже было показано, как метод сходится при условии, что начальная точка не находится в указанной области.*

*Покажем, что произойдет, если поместить начальную точку в указанную область.*

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ 10.000 5.000 ] | 275 | 0 | 3 | 0 |
| **1** | [ 10.259 5.966 ] | 219.8815928 | 55.11840715 | 5 | 0 |
| **2** | [ 8.456 5.931 ] | 98.94236085 | 120.939232 | 7 | 0 |
| **3** | [ 8.073 7.846 ] | 62.07144664 | 36.87091421 | 8 | 0 |
| **4** | [ 8.073 7.846 ] | 62.07144664 | 0 | 10 | 0 |
| **5** | [ 8.073 7.846 ] | 62.07144664 | 0 | 12 | 0 |
| **6** | [ 7.814 6.880 ] | 56.05828393 | 6.013162707 | 14 | 0 |
| **7** | [ 7.814 6.880 ] | 56.05828393 | 0 | 15 | 0 |
| **8** | [ 7.814 6.880 ] | 56.05828393 | 0 | 17 | 0 |
| **9** | [ 6.236 6.009 ] | 36.61926635 | 19.43901758 | 18 | 0 |
| **10** | [ 6.236 6.009 ] | 36.61926635 | 0 | 20 | 0 |
| **11** | [ 4.140 3.206 ] | 19.00199957 | 17.61726678 | 22 | 0 |
| **12** | [ 3.433 3.913 ] | 17.61405237 | 1.387947199 | 24 | 0 |
| **13** | [ 1.337 1.110 ] | 1.746785593 | 15.86726678 | 25 | 0 |
| **14** | [ 1.337 1.110 ] | 1.746785593 | 0 | 27 | 0 |
| **15** | [ 1.337 1.110 ] | 1.746785593 | 0 | 29 | 0 |
| **16** | [ 1.337 1.110 ] | 1.746785593 | 0 | 31 | 0 |
| **17** | [ 1.337 1.110 ] | 1.746785593 | 0 | 32 | 0 |
| **18** | [ 1.337 1.110 ] | 1.746785593 | 0 | 34 | 0 |
| **19** | [ 1.004 1.098 ] | 1.292013582 | 0.454772012 | 36 | 0 |
| **20** | [ 1.004 1.098 ] | 1.292013582 | 0 | 38 | 0 |
| **21** | [ 0.755 0.763 ] | 0.582762673 | 0.709250909 | 40 | 0 |
| **22** | [ 0.755 0.763 ] | 0.582762673 | 0 | 42 | 0 |
| **23** | [ 0.553 0.654 ] | 0.529903393 | 0.05285928 | 44 | 0 |
| **24** | [ 0.657 0.576 ] | 0.396818693 | 0.1330847 | 46 | 0 |
| **25** | [ 0.529 0.541 ] | 0.294174008 | 0.102644685 | 48 | 0 |
| **26** | [ 0.529 0.541 ] | 0.294174008 | 0 | 50 | 0 |
| **27** | [ 0.529 0.501 ] | 0.258697831 | 0.035476177 | 52 | 0 |
| **28** | [ 0.529 0.501 ] | 0.258697831 | 0 | 54 | 0 |
| **29** | [ 0.529 0.501 ] | 0.258697831 | 0 | 55 | 0 |
| **30** | [ 0.529 0.501 ] | 0.258697831 | 0 | 57 | 0 |
| **31** | [ 0.529 0.501 ] | 0.258697831 | 0 | 59 | 0 |
| **32** | [ 0.515 0.494 ] | 0.248514391 | 0.01018344 | 61 | 0 |
| **33** | [ 0.515 0.494 ] | 0.248514391 | 0 | 62 | 0 |
| **34** | [ 0.515 0.494 ] | 0.248514391 | 0 | 64 | 0 |
| **35** | [ 0.520 0.484 ] | 0.247313447 | 0.001200945 | 64 | 0 |
| **=** [0.51999278 0.48440632] **=** 0.24731344654468876 | | | | | |



* + - 1. **Вторая задача**

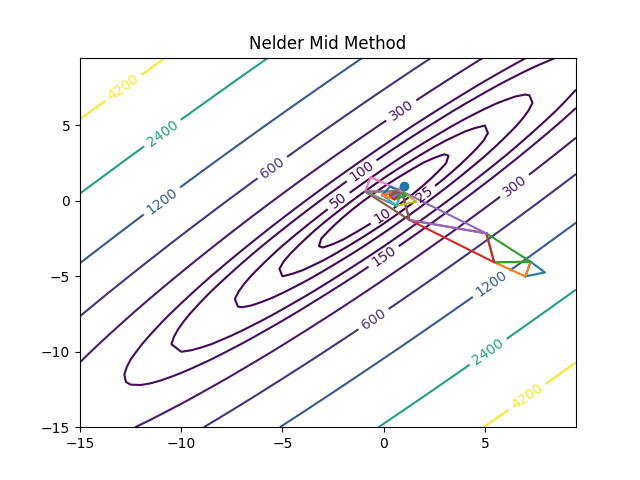
*До этого уже было показано, как метод сходится при условии, что начальная точка не находится в указанной области.*

*Покажем, что произойдет, если поместить начальную точку в указанную область.*

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ 7.000 -5.000 ] | 1465 | 0 | 3 | 0 |
| **1** | [ 7.259 -4.034 ] | 1292.792872 | 172.2071282 | 5 | 1.224744871 |
| **2** | [ 5.456 -4.069 ] | 924.4473975 | 368.3454743 | 7 | 0.612372436 |
| **3** | [ 5.073 -2.154 ] | 527.8588028 | 396.5885947 | 9 | 0.918558654 |
| **4** | [ 1.276 -1.266 ] | 68.23415595 | 459.6246468 | 11 | 1.990210416 |
| **5** | [ 0.893 0.648 ] | 1.476491777 | 66.75766418 | 13 | 0.459279327 |
| **6** | [ 0.893 0.648 ] | 1.476491777 | 0 | 15 | 0.459279327 |
| **7** | [ 0.893 0.648 ] | 1.476491777 | 0 | 16 | 0.459279327 |
| **8** | [ 0.893 0.648 ] | 1.476491777 | 0 | 18 | 0.459279327 |
| **9** | [ 0.893 0.648 ] | 1.476491777 | 0 | 19 | 0.459279327 |
| **10** | [ 0.893 0.648 ] | 1.476491777 | 0 | 21 | 0.459279327 |
| **11** | [ 0.893 0.648 ] | 1.476491777 | 0 | 22 | 0.459279327 |
| **12** | [ 0.893 0.648 ] | 1.476491777 | 0 | 24 | 0.459279327 |
| **13** | [ 0.893 0.648 ] | 1.476491777 | 0 | 25 | 0.459279327 |
| **14** | [ 0.893 0.648 ] | 1.476491777 | 0 | 27 | 0.459279327 |
| **15** | [ 0.893 0.648 ] | 1.476491777 | 0 | 29 | 0.459279327 |
| **16** | [ 0.646 0.722 ] | 1.211037662 | 0.265454115 | 31 | 0.631509074 |
| **17** | [ 0.996 0.837 ] | 1.119804682 | 0.09123298 | 33 | 0.167445588 |
| **18** | [ 0.996 0.837 ] | 1.119804682 | 0 | 35 | 0.167445588 |
| **19** | [ 1.012 0.901 ] | 1.023260933 | 0.096543749 | 37 | 0.087011904 |
| **20** | [ 1.012 0.901 ] | 1.023260933 | 0 | 39 | 0.087011904 |
| **21** | [ 0.895 0.857 ] | 0.996989278 | 0.026271655 | 40 | 0.24840254 |
| **22** | [ 0.895 0.857 ] | 0.996989278 | 0 | 42 | 0.24840254 |
| **23** | [ 0.895 0.857 ] | 0.996989278 | 0 | 44 | 0.24840254 |
| **24** | [ 0.958 0.895 ] | 0.987801025 | 0.009188254 | 46 | 0.147598801 |
| **25** | [ 0.958 0.895 ] | 0.987801025 | 0 | 47 | 0.147598801 |
| **26** | [ 0.958 0.895 ] | 0.987801025 | 0 | 49 | 0.147598801 |
| **27** | [ 1.026 0.975 ] | 0.977468953 | 0.010332072 | 51 | 0.000929734 |
| **28** | [ 1.026 0.975 ] | 0.977468953 | 0 | 53 | 0.000929734 |
| **29** | [ 1.026 0.975 ] | 0.977468953 | 0 | 54 | 0.000929734 |
| **=** [1.02560219 0.97532754] **=** 0.9774689528740341 | | | | | |

****

**Вывод:** По таблицам видно, что метод ведёт себя ожидаемо. Если начальное приближение находится в рассматриваемой области, то штрафные функции не прибавляют значение к целевой функции и таким образом, не как не влияют на неё, до того пока метод не перейдёт через границу области. Соответственно, если начальное приближение находится вне рассматриваемой области, то штрафные функции устремляют метод к рассматриваемой области.

* + 1. *Исследовать сходимость метода штрафных функций в зависимости* задаваемой точности .
       1. **Первая задача**

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Iteration | Calc Func |  |
|  | 4 | [0.51477122 0.48598941] | 0.2444696308318764 | 50 | 87 | 0,00056 |
|  | 1 | [0.51157694 0.48853175] | 0.24397407830576112 | 59 | 109 | 7.16e-05 |
|  | 2 | [0.51208321 0.48793492] | 0.24391188526118746 | 119 | 214 | 9.45e-06 |
|  | 1 | [0.51225121 0.4877503 ] | 0.2439032979992676 | 78 | 145 | 8.59e-07 |
|  | 4 | [0.51219793 0.48780214] | 0.24390247095883708 | 90 | 158 | 3.2e-08 |

* + - 1. **Вторая задача**

*Начальные условия: *

*Штрафная функция: *

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Iteration | Calc Func |  |
|  | 1 | [1.02700176 0.97332707] | 0.9765041358192098 | 41 | 75 | 0,00089 |
|  | 32 | [1.02355822 0.97644384] | 0.9757063262973142 | 77 | 139 | 9.66e-05 |
|  | 4 | [1.02454357 0.97545671] | 0.9756113884308935 | 73 | 129 | 2.36e-06 |
|  | 1 | [1.02430179 0.97569825] | 0.9756101617165851 | 72 | 127 | 4.06e-07 |
|  | 4 | [1.02438278 0.97561722] | 0.9756097640690709 | 89 | 158 | 8.07e-09 |

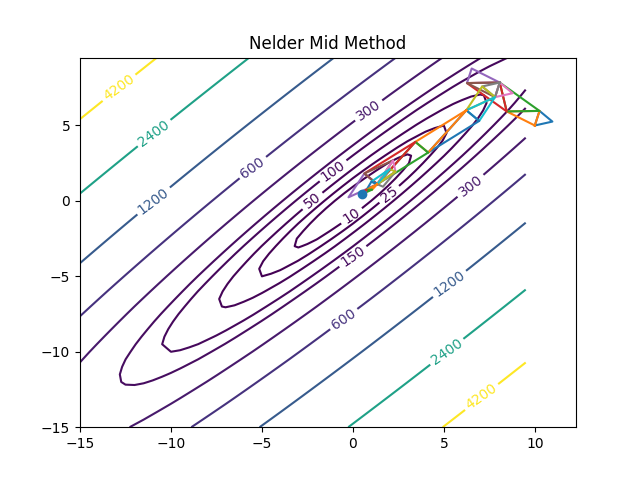
**Вывод:** Из таблиц, которые приведены выше ,видно то, что метод с использованием штрафных функций сходится при увеличении точности.

* 1. **Метод барьерных функций**
     1. *Исследовать сходимость метода барьерных функций в зависимости от выбора штрафных функций*

*Начальные условия: *

*Барьерная функция: *

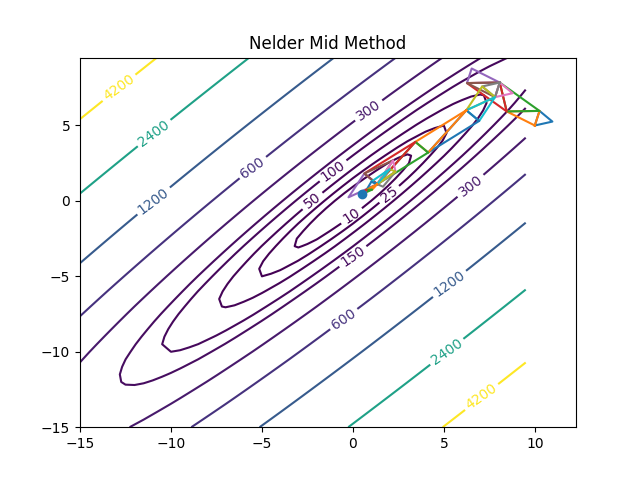
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ 10.000 5.000 ] | 274.9948456 | 0 | 3 | -0.005154409 |
| **1** | [ 10.259 5.966 ] | 219.8762746 | 55.11857095 | 5 | -0.005318207 |
| **2** | [ 8.456 5.931 ] | 98.9372938 | 120.9389808 | 7 | -0.005067053 |
| **3** | [ 8.073 7.846 ] | 62.06616811 | 36.87112569 | 8 | -0.005278527 |
| **4** | [ 8.073 7.846 ] | 62.06616811 | 0 | 10 | -0.005278527 |
| **5** | [ 8.073 7.846 ] | 62.06616811 | 0 | 12 | -0.005278527 |
| **6** | [ 7.814 6.880 ] | 56.05317271 | 6.012995398 | 14 | -0.005111219 |
| **7** | [ 7.814 6.880 ] | 56.05317271 | 0 | 15 | -0.005111219 |
| **8** | [ 7.814 6.880 ] | 56.05317271 | 0 | 17 | -0.005111219 |
| **9** | [ 6.236 6.009 ] | 36.61454006 | 19.43863265 | 18 | -0.004726296 |
| **10** | [ 6.236 6.009 ] | 36.61454006 | 0 | 20 | -0.004726296 |
| **11** | [ 4.140 3.206 ] | 18.99839074 | 17.61614931 | 22 | -0.003608831 |
| **12** | [ 3.433 3.913 ] | 17.61044354 | 1.387947199 | 24 | -0.003608831 |
| **13** | [ 1.337 1.110 ] | 1.746064785 | 15.86437876 | 25 | -0.000720808 |
| **14** | [ 1.337 1.110 ] | 1.746064785 | 0 | 27 | -0.000720808 |
| **15** | [ 1.337 1.110 ] | 1.746064785 | 0 | 28 | -0.000720808 |
| **16** | [ 1.337 1.110 ] | 1.746064785 | 0 | 30 | -0.000720808 |
| **17** | [ 1.337 1.110 ] | 1.746064785 | 0 | 32 | -0.000720808 |
| **18** | [ 1.337 1.110 ] | 1.746064785 | 0 | 33 | -0.000720808 |
| **19** | [ 1.337 1.110 ] | 1.746064785 | 0 | 34 | -0.000720808 |
| **20** | [ 1.337 1.110 ] | 1.746064785 | 0 | 36 | -0.000720808 |
| **21** | [ 0.753 0.736 ] | 0.546331339 | 1.199733446 | 38 | 0.001395123 |
| **22** | [ 0.753 0.736 ] | 0.546331339 | 0 | 40 | 0.001395123 |
| **23** | [ 0.691 0.557 ] | 0.492086481 | 0.054244858 | 42 | 0.002723821 |
| **24** | [ 0.554 0.597 ] | 0.378480921 | 0.11360556 | 44 | 0.003699386 |
| **25** | [ 0.557 0.497 ] | 0.288334477 | 0.090146445 | 46 | 0.005699817 |
| **26** | [ 0.557 0.497 ] | 0.288334477 | 0 | 50 | 0.005699817 |
| **27** | [ 0.557 0.497 ] | 0.288334477 | 0 | 52 | 0.005699817 |
| **28** | [ 0.557 0.497 ] | 0.288334477 | 0 | 54 | 0.005699817 |
| **29** | [ 0.557 0.497 ] | 0.288334477 | 0 | 56 | 0.005699817 |
| **30** | [ 0.557 0.497 ] | 0.288334477 | 0 | 57 | 0.005699817 |
| **31** | [ 0.557 0.497 ] | 0.288334477 | 0 | 59 | 0.005699817 |
| **32** | [ 0.509 0.505 ] | 0.263572728 | 0.024761749 | 61 | 0.00829617 |
| **33** | [ 0.523 0.485 ] | 0.259227931 | 0.004344797 | 63 | 0.009374229 |
| **34** | [ 0.523 0.485 ] | 0.259227931 | 0 | 65 | 0.009374229 |
| = [0.52325278 0.48498022] =0.24985370144719352 | | | | | |

**

*Начальные условия: *

*Барьерная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | Calc Func | Penalty Function |
| **0** | [ 10.000 5.000 ] | 275.0000044 | 0 | 3 | 4.36E-06 |
| **1** | [ 10.259 5.966 ] | 219.8815969 | 55.1184075 | 5 | 4.01E-06 |
| **2** | [ 8.456 5.931 ] | 98.94236541 | 120.9392314 | 7 | 4.56E-06 |
| **3** | [ 8.073 7.846 ] | 62.07145073 | 36.87091468 | 8 | 4.09E-06 |
| **4** | [ 8.073 7.846 ] | 62.07145073 | 0 | 10 | 4.09E-06 |
| **5** | [ 8.073 7.846 ] | 62.07145073 | 0 | 12 | 4.09E-06 |
| **6** | [ 7.814 6.880 ] | 56.05828839 | 6.013162341 | 14 | 4.46E-06 |
| **7** | [ 7.814 6.880 ] | 56.05828839 | 0 | 15 | 4.46E-06 |
| **8** | [ 7.814 6.880 ] | 56.05828839 | 0 | 17 | 4.46E-06 |
| **9** | [ 6.236 6.009 ] | 36.61927178 | 19.43901661 | 18 | 5.43E-06 |
| **10** | [ 6.236 6.009 ] | 36.61927178 | 0 | 20 | 5.43E-06 |
| **11** | [ 4.140 3.206 ] | 19.00200919 | 17.61726259 | 22 | 9.62E-06 |
| **12** | [ 3.433 3.913 ] | 17.61406199 | 1.387947199 | 24 | 9.62E-06 |
| **13** | [ 1.337 1.110 ] | 1.746827792 | 15.8672342 | 25 | 4.22E-05 |
| **14** | [ 1.337 1.110 ] | 1.746827792 | 0 | 27 | 4.22E-05 |
| **15** | [ 1.337 1.110 ] | 1.746827792 | 0 | 28 | 4.22E-05 |
| **16** | [ 1.337 1.110 ] | 1.746827792 | 0 | 30 | 4.22E-05 |
| **17** | [ 1.337 1.110 ] | 1.746827792 | 0 | 32 | 4.22E-05 |
| **18** | [ 1.337 1.110 ] | 1.746827792 | 0 | 33 | 4.22E-05 |
| **19** | [ 1.337 1.110 ] | 1.746827792 | 0 | 34 | 4.22E-05 |
| **20** | [ 1.337 1.110 ] | 1.746827792 | 0 | 36 | 4.22E-05 |
| **21** | [ 0.753 0.736 ] | 0.545060896 | 1.201766896 | 38 | 0.00012468 |
| **22** | [ 0.753 0.736 ] | 0.545060896 | 0 | 40 | 0.00012468 |
| **23** | [ 0.691 0.557 ] | 0.489608836 | 0.05545206 | 42 | 0.000246176 |
| **24** | [ 0.554 0.597 ] | 0.375187203 | 0.114421633 | 44 | 0.000405668 |
| **25** | [ 0.557 0.497 ] | 0.283764415 | 0.091422788 | 46 | 0.001129755 |
| **26** | [ 0.557 0.497 ] | 0.283764415 | 0 | 50 | 0.001129755 |
| **27** | [ 0.557 0.497 ] | 0.283764415 | 0 | 52 | 0.001129755 |
| **28** | [ 0.557 0.497 ] | 0.283764415 | 0 | 54 | 0.001129755 |
| **29** | [ 0.557 0.497 ] | 0.283764415 | 0 | 56 | 0.001129755 |
| **30** | [ 0.557 0.497 ] | 0.283764415 | 0 | 57 | 0.001129755 |
| **31** | [ 0.557 0.497 ] | 0.283764415 | 0 | 59 | 0.001129755 |
| **32** | [ 0.509 0.505 ] | 0.259545362 | 0.024219053 | 61 | 0.004268805 |
| **33** | [ 0.523 0.485 ] | 0.257267182 | 0.00227818 | 63 | 0.007413481 |
| **34** | [ 0.523 0.485 ] | 0.257267182 | 0 | 65 | 0.007413481 |
| **35** | [ 0.523 0.485 ] | 0.257267182 | 0 | 66 | 0.007413481 |
| = [0.52325278 0.48498022] = 0.24985370144719352 | | | | | |

**

**Вывод:** В результате по полученному исследованию можно сделать вывод, что на данных функциях, при каждой барьерной функции метод сошёлся с указанной точностью.

* + 1. *Исследовать сходимость метода барьерных функций в зависимости* коэффициента штрафа

*Начальные условия: *

*Барьерная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Iteration |  |  |
|  | [0.56292833 0.53748596] | 0.29536 | 34 | *0.05146* | *0.14365* |
|  | [0.55242498 0.5335936 ] | 0.28826 | 32 | *0.04436* | *0.07666* |
|  | [0.5318993 0.5102606] | 0.26504 | 36 | *0.02114* | *0.04947* |
|  | [0.52636053 0.49559485] | 0.25507 | 37 | *0.01117* | *0.02983* |
|  | [0.52325278 0.48498022] | 0.24985 | 36 | *0.01117* | *0.01874* |
|  | [0.52325278 0.48498022] | 0.24985 | 36 | *0.00595* | *0.00937* |

*Начальные условия:* 

*Барьерная функция: *

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | Iteration |  |  |
|  | [0.53486271 0.51749214] | 0.2708 | 34 | *0.02691* | *0.03730* |
|  | [0.5318993 0.5102606] | 0.2650 | 36 | *0.02114* | *0.02316* |
|  | [0.5318993 0.5102606] | 0.26504 | 35 | *0.02114* | *0.01158* |
|  | [0.51542748 0.50183677] | 0.25368 | 35 | *0.00978* | *0.01414* |
|  | [0.52636053 0.49559485] | 0.25507 | 37 | *0.01117* | *0.00555* |
|  | [0.52325278 0.48498022] | 0.24985 | 37 | *0.00595* | *0.00741* |

* + 1. *Исследовать сходимость метода штрафных функций в зависимости начальной точки*

До этого уже было показано, как метод сходится при условии, что начальная точка находится в указанной области.

Барьерные функции не предназначены для нахождения минимума при начальном приближении находящимся не в рассматриваемой области. Поскольку в таком случае мы выйдем из области определения барьерной функции. Таким образом, метод не прийдёт к решению задачи.

* + 1. *Исследовать сходимость метода барьерных функций в зависимости* задаваемой точности .

*Начальные условия: *

*Барьерная функция: *

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Iteration | Calc Func |  |
|  |  | [0.51071421 0.48999053] | 0.2443854 | 47 | 85 | 0.00048 |
|  |  | [0.51262523 0.4874155 ] | 0.2439291 | 502 | 1908 | 2.67e-05 |
|  |  | [0.51262523 0.4874155 ] | 0.243929 | 1002 | 3908 | 2.67e-05 |

**Вывод:** Из таблицы, которая приведена выше, видно то, что метод с использованием барьерной функции сходится при увеличении точности до и дальше застревает и выходит по максимальному количеству итераций.

1. **Программа**

import numpy as np

from math import sqrt, log

import matplotlib.pyplot as plt

import pandas as pd

track\_method = np.array([])

Name\_OutFile = "OutputFile"

def penalty\_function(point):

        r0 = 1

        def g1(point):

            return -point[0] - point[1] + 1

        return r0 \* (g1(point) + abs(g1(point)))/2

def f(point):

    def objective\_function(point):

        return 10\*(point[1]-point[0])\*\*2 + point[1]\*\*2

    return objective\_function(point) + penalty\_function(point)

def fg(point):

    def objective\_function(point):

        return 10\*(point[1]-point[0])\*\*2 + point[1]\*\*2

    return objective\_function(point)

def matrix\_to\_string(vector, flag=False):

    if flag:

        out = ""

        for i in range(2):

            out += '[ '

            for j in range(2):

                out += "{0:.1f}".format(vector[i][j]) + " "

            out += "]\n"

    else:

        out = '[ '

        for i in range(2):

            out += "{0:.3f}".format(vector[i]) + " "

        out += "]\n"

    return out

def Nelder\_Mid(func, x0, eps, alpha : float = 1, beta : float = 0.5, gamma : float = 2, t : float = 1, max\_iter : int = 100, \_table : bool = False, \_track : bool = True):

    def norm(f\_res, f\_min):

        return sqrt(sum([(f\_res[i] - f\_min)\*\*2 for i in range(len(f\_res))]))

    if \_table:

        df = pd.DataFrame(columns=np.array(["xmin", 'fmin(x, y)', '|fi - fm1|', 'count calc func', 'penalty function']))

    N = len(x0)

    count\_iter = 0

    count\_calc\_func = 0

    if N < 2:

        return -1

    D = np.zeros((N, N+1))

    d1 = t\*(sqrt(N+1) + N - 1)/(N \* sqrt(2))

    d2 = t \* (sqrt(N + 1) - 1)/(N\*sqrt(2))

    for i in range(N):

        D[i, 0] = x0[i]

        for i in range(N):

            D[i, i+1] = x0[i] + d1

            for j in range(i+2, N+1):

                D[i, j] = x0[i] + d2

                D[j - 1, i + 1] = x0[j - 1] + d2

    if \_track:

        global track\_method

        track\_method = np.array([D])

    res\_f = np.array([func(D[:, i]) for i in range(N+1)])

    count\_calc\_func += N+1

    func\_c = 0

    index\_min = 0

    if \_table:

        old\_func = func(x0)

        df.loc[len(df)] = np.array([matrix\_to\_string(x0), old\_func, 0, count\_calc\_func, penalty\_function(x0)])

    while norm(res\_f, func\_c) >= eps and count\_iter < max\_iter:

        func\_c = min(res\_f)

        index\_max = np.where(res\_f == max(res\_f))[0][0]

        index\_min = np.where(res\_f == min(res\_f))[0][0]

        temp\_res = sorted(res\_f)

        xc = np.zeros((N,))

        for i in range(N + 1):

            if i != index\_max:

                xc += D[:, i]

        xc /= N

        xr = (1 + alpha)\*xc - alpha\*D[:, index\_max]

        f\_r = func(xr)

        count\_calc\_func += 1

        if f\_r < temp\_res[0]:

            xe = (1 - gamma)\*xc + gamma\*xr

            f\_e = func(xe)

            count\_calc\_func += 1

            if f\_e < f\_r:

                D[:, index\_max] = xe

                res\_f[index\_max] = f\_e

            else:

                D[:, index\_max] = xr

                res\_f[index\_max] = f\_r

        elif f\_r < temp\_res[1]:

            D[:, index\_max] = xr

            res\_f[index\_max] = f\_r

        else:

            if f\_r > res\_f[N]:

                D[:, index\_max] = xr

                temp\_res[N] = f\_r

            xs = beta\*D[:, index\_max] + (1-beta)\*xc

            f\_s = func(xs)

            count\_calc\_func += 1

            if f\_s < temp\_res[N]:

                D[:, index\_max] = xs

                res\_f[index\_max] = f\_s

            else:

                xl = D[:, index\_min]

                for i in range(N+1):

                    if i != index\_min:

                        D[:, i] = xl + (D[:, i] - xl)

                        res\_f[i] = func(D[:, i])

                        count\_calc\_func += 1

        count\_iter += 1

        if \_table:

            df.loc[len(df)] = np.array([matrix\_to\_string(D[:, index\_min]), func\_c ,abs(func\_c - old\_func), count\_calc\_func, penalty\_function(D[:, index\_min])])

            old\_func = func\_c

        if \_track:

            track\_method = np.vstack((track\_method, [D]))

    index\_min = np.where(res\_f == min(res\_f))[0][0]

    if \_table:

        df.loc[len(df)] = np.array([matrix\_to\_string(D[:, index\_min]), res\_f[index\_min], abs(res\_f[index\_min] - old\_func), count\_calc\_func, penalty\_function(D[:, index\_min])])

        return df, D[:, index\_min]

    return D[:, index\_min]

def create\_grafic\_track(res):

    x\_grid = np.arange(-15, 10, 0.5)

    y\_grid = np.arange(-15, 10, 0.5)

    x, y = np.meshgrid(x\_grid, y\_grid)

    func\_z = 10 \* (y - x) \*\* 2 + y \*\* 2

    cs = plt.contour(x, y, func\_z, zorder=1, levels=[10, 25, 50, 100, 150, 300, 600, 1200, 2400, 4200])

    plt.clabel(cs)

    if len(track\_method) > 0:

        for i in range(len(track\_method)):

            temp\_x, temp\_y = track\_method[i][0], track\_method[i][1]

            plt.plot(np.append(temp\_x, temp\_x[0]), np.append(temp\_y, temp\_y[0]), zorder=2)

        plt.scatter(res[0], res[1], zorder=3)

    plt.title("Nelder Mid Method")

    plt.show()

def out\_exel(df):

    if Name\_OutFile != "":

        with pd.ExcelWriter(Name\_OutFile + ".xlsx") as writer:

            df.to\_excel(writer, sheet\_name="Nelder Mid Method", index\_label='i', float\_format="%.8f")

def main():

    df, res = Nelder\_Mid(f, np.array([10, 5]), 1e-5, \_table=True, max\_iter=1000)

    out\_exel(df)

    print(res, len(df), fg(res), fg(res) - 0.243902439, sum(res), abs(penalty\_function(res)))

    create\_grafic\_track(res)

if \_\_name\_\_ == "\_\_main\_\_":

    main()