**Лабораторная работа №3**

### **Решение задачи нелинейного программирования градиентными методами**

Присылаемый на проверку архив должен содержать 2 файла:

* файл отчета, содержащий титульный лист, условие задачи, описание используемого метода, обоснования выбора начального приближения решения исходной задачи, исходный текст программы (с указанием языка реализации), результаты работы программы с промежуточными вычислениями (можно в виде скриншотов);
* файл с исходным текстом программы (программу можно писать на любом языке программирования).

**Задание на лабораторную работу**

Написать программу, находящую решение задачи нелинейного программирования методом Эрроу-Гурвица с точностью 0.0001. В качестве значения  возьмите 0.001.

Вариант выбирается по последней цифре пароля.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

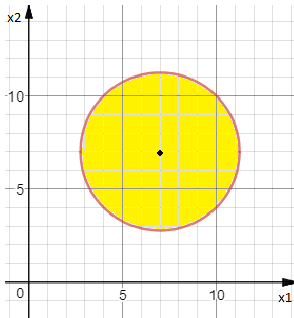
**Методические указания к выполнению лабораторной работы №3**

Разберем алгоритм, который Вы должны запрограммировать, на примере решения задачи:



Здесь разбирается алгоритм, Вам необходимо написать программу.

1. Построим область допустимых решений:



Возьмем любое допустимое решение системы ограничений в качестве начального приближения .

2. В качестве начального шага вычислений выберем .

3. Преобразуем ограничение-неравенство к виду , введем обозначение .

4. Определим частные производные от функций *f* и *g*:



5. Далее запускаем итерационный процесс. Координаты следующей точки будем находить по формулам:

,

где .

*Итерация 1*

Т.к. предыдущая точка  принадлежит области допустимых решений исходной задачи, то , то  и второе слагаемое в квадратных скобках равно 0. Заметим, что так будет для всех точек, принадлежащих области допустимых решений.

Координаты следующей точки:



Получили новую точку . Проверим принадлежит ли точка области допустимых решений (подставим в систему ограничений координаты найденной точки):

.

Следовательно, найденная точка принадлежит области допустимых решений и .

Проверим достижение требуемой точности: . Точность не достигнута. Продолжаем итерационный процесс.

*Итерация 2*

Находим координаты точки  ().



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка  не принадлежит области допустимых решений.

*Итерация 3*

Находим  (предыдущая точка была вне области), .

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка не принадлежит области допустимых решений.

*Итерация 4*

Находим  (предыдущая точка была вне области), 

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка не принадлежит области допустимых решений.

*Итерация 5*

Находим  (предыдущая точка была вне области), 

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка принадлежит области допустимых решений, .

Проверим достижение требуемой точности (сравниваем со значением функции в точке, которая последней попадала в область): . Точность не достигнута. Продолжаем итерационный процесс.

*Итерация 6*

Находим координаты точки  (, т.к. предыдущая точка попадала в область).



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка  не принадлежит области допустимых решений.

*Итерация 7*

Находим  (предыдущая точка была вне области), 

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка  не принадлежит области допустимых решений.

*Итерация 8*

Находим  (предыдущая точка была вне области), 

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка  не принадлежит области допустимых решений.

*Итерация 9*

Находим  (предыдущая точка была вне области), 

Находим координаты точки :



Получили новую точку . Проверим принадлежит ли точка области допустимых решений:

.

Следовательно, найденная точка принадлежит области допустимых решений, .

Проверим достижение требуемой точности (сравниваем со значением функции в точке, которая последней попадала в область): . Точность не достигнута. Продолжаем итерационный процесс.

И так далее.