

**СТРУКТУРА И СОДЕРЖАНИЕ ПРОГРАММНОГО СРЕДСТВА
ПРОВЕРКИ РЕШЕНИЯ ЭКСТРЕМАЛЬНЫХ ЗАДАЧ
МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ
МЕТОДОМ НАИСКОРЕЙШЕГО СПУСКА
ДЛЯ ЭЛЕКТРОННЫХ СИСТЕМ ОБУЧЕНИЯ**

Адаптивная система контроля выполнения заданий в ходе образовательного процесса при использовании электронных систем обучения очевидно имеет немаловажное значение. Особую актуальность вопрос контроля приобретает в случае дистанционного обучения, связанного с большим количеством обучающихся.

Впрочем, наличие такого программного средства в составе системы обучения могло бы не только преподавателям, но и студентам.

Нам представляется, что такого рода программы не требуют какого-то изысканного интерфейса взаимодействия с пользователем (преподавателем или студентом). Рассмотрим, например, задачу поиска экстремума некой целевой функции, заданной в виде квадратичной формы [1] методом наискорейшего спуска. На наш взгляд этот метод требует обязательного рассмотрения в ходе изучения курса «Методы математического программирования». Рассматриваемый метод относится к классу безусловных [2], поэтому координаты начальной точки поиска решения можно задавать произвольно, предоставляя эту возможность пользователям. Если в вариантах заданий для студентов ограничиться представлением целевых функций в виде квадратичных форм:

$$F(x) = C^T x + x^T D x = \sum_{i=1}^n c_i x_i + \sum_{i,k=1}^n x_i d_{ik} x_k,$$

то для задания целевой функции в программе требуется задание матриц коэффициентов **C** и **D**, где **C** – матрица коэффициентов в линейной части выражения для целевой функции, **D** – симметричная матрица в нелинейной части того же выражения.

Практика проведения практических занятий по курсу «Методы математического программирования» на кафедре ИСиТ ИИТ показывает, что для уяснения сути метода наискорейшего спуска достаточно решения задач с квадратичными функциями цели.

Рассматриваемый метод является градиентным, но для квадратичных функций, заданных через **C** и **D**, координаты градиента можно легко высчитать, используя выражение:

$$\frac{\partial F}{\partial x} = C + 2Dx. \quad (1)$$

Тогда для текущего x^k , вычисляем градиент по формуле:

$$\nabla F(x^k) = C + 2Dx^k, \quad (2)$$

т. е. «запрограммировать» следует сложение/умножение массивов.

В общем, структуру программы можно представить в виде следующей схемы (рисунок 1).

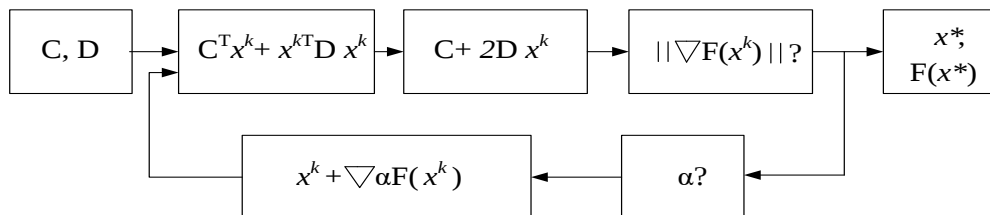


Рисунок 1 – Структура программного средства решения экстремальной задачи

Выход из цикла поиска решения на наш взгляд лучше осуществлять по «малости» градиента, высчитав его «норму».

Расчёт шага движения можно проводить «пробами» α^n с запоминанием значения, при котором

$$\nabla F(x^{k+1})^T \nabla F(x^k) = 0$$

в соответствии с блок-схемой алгоритма, изображенного на рисунке 2.

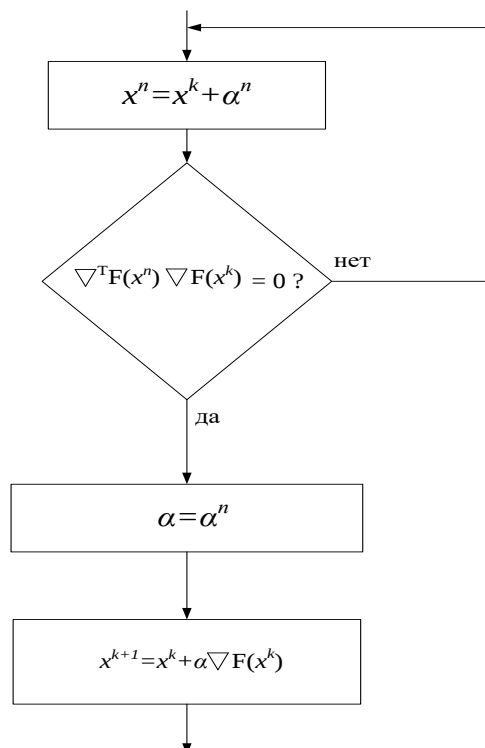


Рисунок 2 – Блок-схема алгоритма поиска шага

Задача вычисления шага в соответствии с рисунком 2 является подзадачей общей задачи поиска решения.

Литература

1. Методы оптимизации: учебник и практикум / под ред. Ф. П. Васильева. – М. : Юрайт, 2019. – 375 с.
2. Гончаров, В. А. Методы оптимизации: учебное пособие для вузов / В. А. Гончаров. – М. : Юрайт, 2014. – 191 с.