**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное образовательное  
 учреждение высшего образования   
«Южный федеральный университет»**



**Кафедра «Прикладная информатика и инноватика»**

**Направление**

**09.03.03 "Прикладная информатика"**

**ОТЧЕТ КУРСОВОГО ЗАДАНИЯ**

**по дисциплине "Разработка приложений в среде MatLab"**

**Автор: Хамадов Константин Константинович**

**студент 3 курса 7 группы**

## Принял: Толмачев Сергей Алексеевич

**Ростов-на-Дону**

**2021**

**ТЕОРИЯ**

**Формулировка задания**

**Цель работы:** разработать графическое приложение в среде GUI MatLab для решения задач многомерной оптимизации (рис. 1).

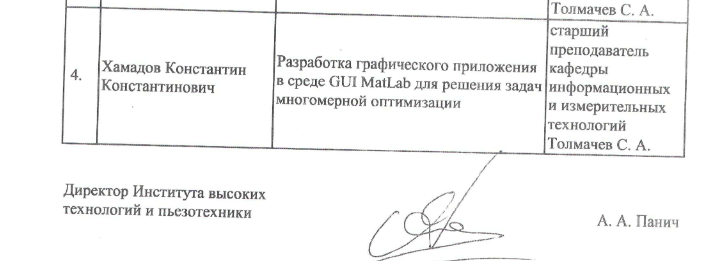


Рис. 1. Цель работы

**Оптимизация**

***Оптимизация*** – задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств. Теорию и методы решения задачи оптимизации изучает «математическое программирование».

***Методы оптимизации:***

- Методы исследования функций классического анализа

- Методы вариационного исчисления

- Динамическое программирование

- Принцип максимума

- Линейное программирование

- Методы нелинейного программирования

- Геометрическое программирование  
  
 Классификация оптимизации по математической природе оптимизируемых величин:

- Одномерная оптимизация

- Многомерная оптимизация

- Оптимальное управление

- Безусловная оптимизация

- Условная оптимизация

- Метод барьерных функций

**Задачи многомерной оптимизации**

**Общие положения**

*Математическая модель объекта оптимизации* описывает объект при помощи соотношений между величинами, характеризующими его свойства. Часть этих величин можно изменять в некоторых пределах, что порождает множество альтернатив.

Изменяемые при оптимизации величины, входящие в математическую модель объекта оптимизации, называют *параметрами оптимизации*, а соотношения, устанавливающие пределы возможного изменения этих параметров, – *ограничениями*.

В большинстве реальных задач оптимизации целевая функция зависит от нескольких проектных параметров. Такая задача называется многомерной. Многомерная оптимизация представляет собой поиск наименьшего или наибольшего значения целевой функции.

Если множество параметров оптимизации является подмножеством конечномерного линейного пространства, то говорят о *конечномерной задаче оптимизации* в отличие от *бесконечномерных задач*, которые рассматривают в вариационном исчислении и оптимальном управлении. При этом критерием оптимальности может быть требование достижения наибольшего или наименьшего значения одной или несколькими действительными (скалярными) функциями параметров оптимизации, выражающими количественно меру достижения цели оптимизации рассматриваемого объекта. Каждую из таких функций принято называть целевой.

Если целевая функция единственная, то задачу конечномерной оптимизации называют *задачей математического программирования*, а в противном случае — *задачей многокритериальной (векторной) оптимизации*.

**Постановка задач многомерной оптимизации**

При математической формулировке задачи условной оптимизации целевую функцию выбирают с таким знаком, чтобы решение задачи соответствовало поиску минимума этой функции. Поэтому формулировку общей задачи математического программирования обычно записывают так (1.1):

(1.1)

где — множество возможных альтернатив, рассматриваемых при поиске решения задачи. Любую точку называют допустимым решением задачи математического программирования, а само множество — множеством допустимых решений или, короче, *допустимым множеством*. Точку , в которой функция достигает своего наименьшего значения, называют оптимальным решением задачи. При отсутствии ограничений множество совпадает с областью определения целевой функции. Если же рассматриваемые альтернативы должны удовлетворять некоторым ограничениям, то множество допустимых решений сужается.

Задачу вида 1.1 в дальнейшем будем называть задачей минимизации целевой функции на множестве . Но целевая функция может и не достигать на наименьшего значения. Тогда говорят о точной нижней грани функции на этом множестве и используют запись 1.2:

(1.2)

Отличие в том, что в первом случае предполагают существование точки , в которой целевая функция достигает своего наименьшего значения на множестве , а во втором случае такая точка может и не существовать.

Поэтому решение общей задачи математического программирования состоит в том, чтобы в первом случае найти точные (или с некоторой заданной точностью) значения координат точки и значение целевой функции .

Во втором случае построить такую последовательность точек которой бы соответствовала последовательность , сходящаяся к значению , и вычислить это значение с заданной точностью.

Отметим, что в большинстве прикладных задач имеет место первый случай, поэтому использование записи второго вида будем оговаривать особо.

Сформулируем *задачу многомерной безусловной оптимизации*: найти минимум функции , где при отсутствии ограничений на при этом – это скалярная целевая функция, непрерывно дифференцируемая.

При решении этого класса задача необходимо учитывать следующие факторы:

- Характер целевой функции решаемой задачи (одноэкстремальная или многоэкстремальная)

- Возможность получения в процессе оптимизации информации о производных целевой функции (возможность присутствует или отсутствует)

- Наличие различных подходов к организации итеративной процедуры поиска оптимума (методы, основанные на итеративном движении переменных в направлении, определяемом тем или иным способом)

Все методы решения задач безусловной оптимизации состоят в том, что мы строим последовательность точек таким образом, чтобы последовательность функций была убывающей (т. е. спускаемся вдоль функции). На k-м шаге ( k > 0 ) определяем вектор , в направлении которого функция уменьшается. В этом направлении делаем шаг величиной и получаем новую точку , в которой . Последовательность , удовлетворяющая этому

условию, называется *релаксационной последовательностью*, а соответствующие методы – *методами спуска*.

Методы решения делятся на методы с использованием информации о производных функции и без использования таковой. Различные методы спуска отличаются выбором направления и величины шага. Как правило, для нахождения используется процедура одномерного поиска.

**Методы прямого поиска**

В методах прямого поиска минимума целевой функции (или методах нулевого порядка) используют информацию только о значениях этой функции. Многие из этих методов не имеют строгого теоретического обоснования и построены на основе эвристических соображений. Поэтому вопросы сходимости методов прямого поиска еще мало изучены, а оценки скорости сходимости обычно отсутствуют. Вместе с тем эти методы идейно связаны с методами первого и второго порядков, что в ряде случаев позволяет оценивать эффективность алгоритмов прямого поиска применительно к минимизации некоторых классов функций. Распространенным способом оценки эффективности методов прямого поиска являются вычислительные эксперименты и сравнительный анализ методов по результатам таких экспериментов. Однако следует учитывать, что этот анализ не во всех случаях может приводить к однозначным выводам о преимуществах одного метода перед другим. Во-первых, это связано с тем, что сравнению обычно подвергаются не только методы, но и программные реализации соответствующих алгоритмов. Хороший метод можно „загубить" плохим программированием, неудачным выбором параметров алгоритма.

Во-вторых, методы могут вести себя по-разному на различных этапах процесса минимизации. Удовлетворительного способа преодоления указанных трудностей не существует. Единственное, что можно сделать в подобной ситуации, — привести данные о результатах вычислений в развернутой форме, позволяющей сравнивать методы по различным критериям. Кроме того, не следует забывать, что поиск решения всегда остается искусством, которому можно научиться лишь путем проб и ошибок, применяя различные методы при решении конкретных задач. К методам нулевого порядка относятся методы, не использующие производные для выбора направления спуска: метод Гаусса, метод вращающихся направлений (Розенброка); метод деформируемого многогранника (поиска по симплексу); метод Хука Дживса, метод Пауэлла.

**Метод Гаусса Зейделя (покоординатный спуск)**

Изображён на рис. 2. В качестве направлений поиска используются координатные векторы где . Таким образом, в поиске по направлению меняется только переменная остальные переменные фиксируются. Задаем начальную точку . Рассматриваем направление поиска . Находим параметр из условий одномерной минимизации . Обозначим промежуточную точку . Перейдем ко второму направлению . Находим параметр из условия одномерной оптимизации , обозначим , проходим по всем направлениям координатных осей, определяя , где находим из условия. Следующая точка . Рассматриваем ее как стартовую, далее повторяем процесс поиска по направлениям координатных осей. Процесс поиска останавливаем при выполнении условия

или , где ε – заданная точность. В качестве оптимального решения выбираем .

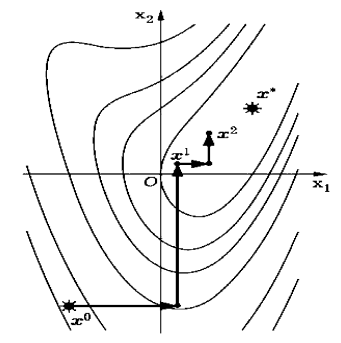


Рис 2. Метод покоординатного спуска.

**Достоинством** метода покоординатного спуска является его простота при определении перемещения в пространстве переменных. Для гладких функций метод обеспечивает сходимость к точке локального минимума.

**Недостаток**. При минимизации «овражных» функций алгоритм будет делать очень мелкие шаги и может остановиться далеко от оптимума.

**Методы первого порядка**

Методы 1-го порядка используют информацию о производной функции. Если ограниченная снизу целевая функция  является дифференцируемой на множестве  , то алгоритм поиска точки х\* ее минимума можно построить, используя информацию, по крайней мере, о градиенте этой функции. Такие методы называются градиентными. Градиентные методы безусловной оптимизации используют только первые производные целевой функции и являются методами линейной аппроксимации на каждом шаге, т. е. целевая функция на каждом шаге заменяется касательной гиперплоскостью к ее графику в текущей точке. Во всех этих методах предполагается, что f (x), ∇f существуют и непрерывны. Все эти методы основаны на итерационной процедуре, определяемой формулой  , где  величина шага,  вектор в направлении . Градиентные методы различаются только способом определения  , и  обычно определяется путём решения задачи оптимизации f (x) в направлении  . Направление  зависит от того, как аппроксимируется функция f (x).

**Метод наискорейшего спуска**

Изображён на рис. 3. Впервые такой метод рассмотрел и применил еще О. Коши в XVIII в. Идея его проста: градиент целевой функции в любой точке есть вектор в направлении наибольшего возрастания значения функции. Следовательно, антиградиент будет направлен в сторону наибольшего убывания функции и является направлением наискорейшего спуска. Антиградиент (и градиент) ортогонален поверхности уровня в точке .

Пусть в точке x требуется определить направление наискорейшего спуска (то есть направление наибольшего локального уменьшения . Разложим в ряд Тейлора в окрестности точки x и отбросим члены второго порядка по Δx и выше …

Локальное уменьшение определяется вторым слагаемым, т. е. наибольшее уменьшение будет тогда, когда будет иметь наибольшую отрицательную величину. Этого можно добиться выбором . Этот случай соответствует наискорейшему локальному спуску .

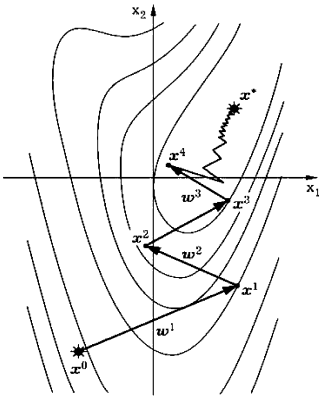


Рис. 3. Метод наискорейшего спуска.

*Недостатки*: одним из недостатков этого метода является то, что он сходится к любой стационарной точке, в том числе и седловой, которая не может быть решением. Также отмечена очень медленная сходимость наискорейшего спуска в общем случае. Дело в том, что спуск является "наискорейшим" в локальном смысле. Если гиперпространство поиска сильно вытянуто ("овраг"), то антиградиент направлен почти ортогонально дну "оврага", т. е. наилучшему направлению достижения минимума. В этом смысле прямой перевод английского термина "steepest descent", т. е. спуск по наиболее крутому склону, более соответствует положению дел, чем термин "наискорейший", принятый в русскоязычной специальной литературе. Метод обладает большой надёжностью, но медленную сходимость вблизи точки минимума устранить нельзя. Поэтому метод самостоятельно обычно не используется, а используется как предварительная процедура для более сложных методов.

*Достоинство*: на каждой k-ой итерации выполняется свойство убывания функции.

**Методы сопряженных градиентов**

В методе сопряженных направлений происходит отклонение направления наискорейшего спуска путем добавления к нему направления, используемого на предыдущем шаге. В методе сопряженного градиента строится последовательность направлений поиска , являющихся линейными комбинациями градиента текущего направления наискорейшего спуска, и предыдущих направлений поиска, т. е.

**Методы 2-го порядка**

Метод 2 порядка указан на рис. 4. Если целевая функция является дважды дифференцируемой в , то эффективность процесса поиска точки ее минимума можно повысить, используя информацию не только о градиенте этой функции, но и о ее матрице Гессе . Направление поиска, соответствующее наискорейшему спуску (методы, рассмотренные выше), связано с линейной аппроксимацией целевой функции. Методы, использующие вторые производные, возникли из квадратичной аппроксимации целевой функции: f (x), 27 которую можно получить при разложении функции в ряд Тейлора 2-го порядка (1.3),

(1.3)

где матрица Гессе (вторых производных). Минимум (если он существует) достигается там же, где и минимум квадратичной формы .

Если матрица Гессе целевой функции, вычисленная в точке , является положительно определенной, то точка минимума функции единственна и может быть найдена из условия, что ее градиент равен нулевому вектору: , следовательно, .

Алгоритм оптимизации, в котором направление поиска определяется из этого соотношения, называется *методом Ньютона*, а направление – *ньютоновским направлением*.

В задачах поиска минимума произвольной квадратичной функции с положительной матрицей вторых производных метод Ньютона дает решение за одну итерацию независимо от выбора начальной точки.

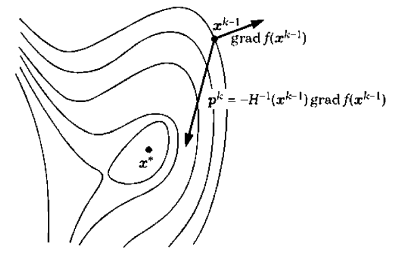


Рис. 4. Один из методов 2 порядка.

**Модификации метода Ньютона**

Метод Ньютона считается эталонным, с ним сравнивают все разрабатываемые оптимизационные процедуры. Для применения метода Ньютона матрица Гессе должна быть положительно определенной и хорошо обусловленной (определитель ее должен быть существенно больше нуля, т. е. отношение наибольшего и наименьшего собственных чисел должно быть близко к единице). Но матрица Гессе может быть вырожденной и не иметь обратной матрицы. Эту проблему можно решить, если направление спуска задавать вектором , где I — единичная матрица порядка n, а — параметр, выбираемый так, чтобы в точке матрица была положительно определена.

В связи с этим более серьезной проблемой является необходимость вычисления и обращения на каждой итерации матрицы порядка n, что в случае большой размерности пространства является достаточно трудоемкой операцией. На практике обычно не вычисляют матрицу, обратную к положительно определенной матрице , а вектор находят из решения системы линейных алгебраических уравнений (СЛАУ) . Эту СЛАУ можно решить различными численными методами, например прямыми и итерационными.

Эти приемы называются модификацией метода Ньютона, они используют ньютоновские направления по мере возможности и уклоняющиеся от них только тогда, когда это необходимо.

Общий принцип модификаций метода Ньютона состоит в следующем: на каждой итерации сначала строится некоторая "связанная" с , положительно определенная матрица , а затем направление спуска вычисляется по формуле . Так как положительно определена, то направление обязательно будет направлением спуска. Процедуру построения организуют так, чтобы она совпадала с матрицей Гессе, если она является положительно определенной. Эти процедуры строятся на основе некоторых матричных разложений. Опишем алгоритм варианта метода Ньютона поиска точки минимума дважды дифференцируемой целевой функции , в котором направление спуска определяется путем решения СЛАУ.

**Методы переменной метрики**

Среди алгоритмов многомерной минимизации следует выделить группу алгоритмов, которые объединяют достоинства метода наискорейшего спуска и метода Ньютона. Такие алгоритмы принято относить к так называемым квазиньютоновским методам. Особенность этих алгоритмов состоит в том, что при их применении нет необходимости вычислять и обращать матрицу Гессе целевой функции и в то же время удается сохранить высокую скорость сходимости алгоритмов, присущую методу Ньютона и его модификациям.

В этих методах обратная матрица Гессе аппроксимируется другой матрицей – метрикой. Метрика изменяется на каждой итерации, и поэтому методы так же называются методами с переменной метрикой.

Элементы релаксационной последовательности в алгоритмах квазиньютоновских методов минимизации непрерывно дифференцируемой в целевой функции строят в соответствии с рекуррентным соотношением , направление спуска на каждой k-й итерации задают в виде , где – метрика, а , где − корректирующая матрица. Нужно построить последовательность метрик , которая при имела бы предел, равный , где − матрица Гессе в точке минимума функции .

**Методы Пирсона**

Пирсон предложил несколько методов с аппроксимацией обратного гессиана без явного вычисления вторых производных, т. е. путем наблюдения за изменениями направления антиградиента. При этом получаются сопряженные направления. Эти алгоритмы отличаются только деталями. Приведем те из них, которые получили наиболее широкое распространение в прикладных областях.

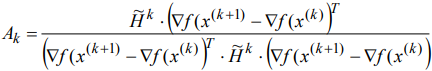
Они аппроксимируют матрицу Гессе или обратную к ней, но используют для этого только первые производные. В большинстве из методов применяются сопряжённые направления.

На каждом шаге , где  − направление поиска. В этом алгоритме обратный гессиан аппроксимируется матрицей, вычисляемой на каждом шаге по формуле



или в другой модификации

,

где .

В качестве начальной матрицы  выбирается произвольная положительно определенная, симметрическая матрица (обычно единичная). Данный алгоритм Пирсона часто приводит к ситуациям, когда матрица  становится плохо обусловленной, а именно − она начинает осциллировать, колеблясь между положительно определенной и не положительно определенной, при этом определитель матрицы близок к нулю. Для того чтобы избежать этой ситуации, необходимо через каждые n шагов перезадавать матрицу, приравнивая ее к .

**Методы случайного поиска**

Методы случайного поиска реализуют итеративный процесс движения оптимизационных переменных в пространстве с использованием случайных направлений. Одно из преимуществ этих методов – достаточная простота, методы обладают большим спектром возможных направлений движения.

**СОЗДАНИЕ ПРИЛОЖЕНИЯ**

Перед созданием приложения была сформирована **цель**: создать приложение, способное визуально представлять задаваемую функцию, вычислять глобальные минимум и максимум целевой функции при заданных параметрах, а также отображать на графике локальные и глобальные экстремумы.

**Было принято решение создать следующие элементы приложения:**

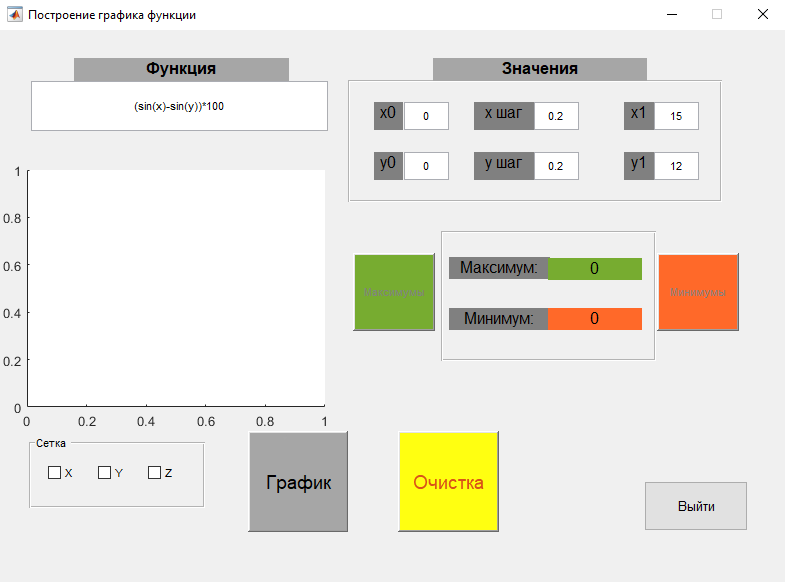
1. Поле с графиком функции от двух переменных (х и у).
2. Поле для ввода этой функции пользователем.
3. Поля для ввода стартовых и предельных значений переменных x и y, а также их шага.
4. Несколько переключателей, позволяющих включать и выключать показ сетки по осям x, y и z.
5. Кнопка построения графика.
6. Кнопки вычисления глобальных минимума и максимума функции.
7. Кнопки отображения на графике локальных экстремумов.
8. Кнопка очистки графика.
9. Кнопка выхода из приложения.

Приложение создавалось при использовании MATLAB R2021a (приложения ЮФУ) – это версия MatLab с удалённым доступом, которую могут использовать студенты ЮФУ. Для начала работы с приложением в Command Window необходимо было подать запрос «guide», после чего программа дала возможность работы с графическими приложениями.  
  
Приложение состоит из 2 файлов:

Project.fig – файл с графическим интерфейсом программы.

Project.m – файл с кодом.

Внешний вид приложения:



**Кнопка «График»** изначально активна, но становится недоступна после нажатия на неё. Становится вновь активной при нажатии на кнопку «Очистка».

**Кнопка «Очистка»** изначально недоступна, но становится активной при нажатии на кнопку «График». После очистки снова становится неактивной.

**Кнопки «Максимумы»** и **«Минимумы»** изначально недоступны, но становятся активными при нажатии на кнопку «График». При нажатии на себя становятся неактивны. Также отключаются при нажатии на кнопку «Очистка».

**Кнопка «Выйти»** активна всегда.

**Чек-боксы «Сетка»** активны всегда.

**Поля ввода «Функция»** и **«Значения»** изначально доступны, но становятся неактивны при нажатии кнопки «График». Снова становятся активными при нажатии на кнопку «Очистка».

**Как работать с приложением**

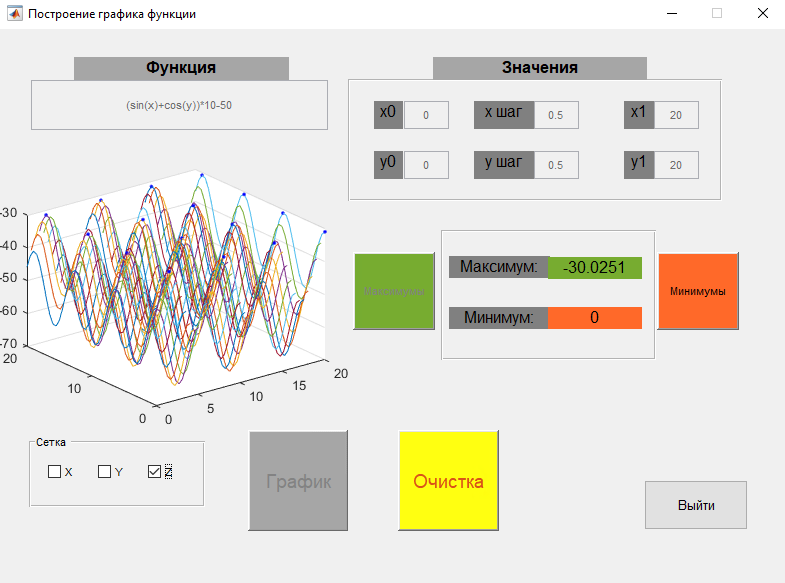
Введите исходные данные, а именно функцию и значения переменных. Не забывайте соблюдать правила синтаксиса MatLab при вводе функции. После чего нажмите на кнопку «График». В случае ошибки нажмите «Очистить» и начните заного.

Для удобства можете включить сетку.

После выше описанных действий появится график функции. На нём наглядно видно как функция ведёт себя на определённых участках заданной области определения. Для вычисления глобальных максимума и минимума воспользуйтесь соответствующими кнопками. При их нажатии на графике также появятся жирные цветные круги в местах экстремумов.

После всех действий можно либо начать заново, изменив значения входных аргументов, либо выйти из программы при помощи соответствующей кнопки.

Скриншот работы программы:



**Вывод:** во время работы мною были получены знания в области математического программирования, а также мною было создано графическое приложение в среде MatLab, способное выполнять задачи многомерной оптимизации.