

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ
VI Международной молодежной
научной конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 24–26 мая 2018 г.

*Под общей редакцией
кандидата технических наук И.С. Шмырина*

Томск
Издательский Дом Томского государственного университета
2018

РАЗРАБОТКА И РЕАЛИЗАЦИЯ СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КУРСА «МЕТОДЫ ОПТИМИЗАЦИИ»

В.З. Тычинский

*Томский государственный университет
tychinskiy.vyacheslav@gmail.com*

Введение

Задачей методов оптимизации в общем случае является выбор какого-либо действия x^* из множества допустимых действий D , при этом соблюдаются два условия:

- 1) каждому действию x из D взаимно однозначно соответствует величина $f(x)$, характеризующая его качество;
2. $f(x)$ достигает экстремума в точке x^* .

Целями лабораторных работ курса «Методы оптимизации» является формирование у студентов общего представления о специфике оптимизационных задач, а также наглядная демонстрация преимуществ и недостатков тех или иных методов решения таких задач. Для выполнения этих целей в 2001 году было разработано специализированное программное обеспечение, однако при его анализе был обнаружен ряд недостатков:

- проблемы совместимости с современными операционными системами;
- отсутствие файлов с исходным кодом (для внедрения улучшений и исправлений);
- отсутствие кроссплатформенности;
- ошибки в реализациях некоторых алгоритмов, связанные с недостаточной точностью вычислений;
- неудобный интерфейс.

Ввиду вышеперечисленных причин имеется необходимость в разработке нового программного обеспечения, обладающего тем же функционалом, но лишённого вышеперечисленных недостатков. При этом такое решение предпочтительнее использования готовых пакетов прикладных программ, поскольку

- 1) даёт возможность выполнения лабораторных работ вне университета;
- 2) выполнение лабораторных работ возможно студентами, не обладающими опытом работы с пакетами прикладных программ;
- 3) программное обеспечение проще адаптировать при изменении инфраструктуры, в то время как в случае использования пакетов прикладных программ это может потребовать больших затрат.

Для разработки программного обеспечения был выбран язык C++, обеспечивающий хорошее быстродействие, что имеет существенное значение в тех лабораторных работах, где необходимо проводить большое количество вычислений. Кроме того, при разработке использованы библиотеки Qt, обеспечивающие кроссплатформенность и позволяющие уменьшить связность компонент программного обеспечения.

1. Постановка задачи

В лабораторных работах курса "Методы оптимизации" в настоящее время рассматриваются методы 5 разделов:

1. Минимизация функции градиентными методами
2. Минимизация функции овражными методами
3. Минимизация функции методами прямого поиска
4. Решение задачи линейного программирования симплекс-методом
5. Решение транспортной задачи методом потенциалов

В каждом из методов пользователем задаются различные исходные данные и параметры, необходимые для работы метода.

На основе этих входных данных требуется выполнить выбранный метод и вывести на экран результаты его работы (графически или в виде текстового отчёта). В некоторых методах требуется интерактивное взаимодействие с пользователем.

Для реализации этих методов программное обеспечение должно обладать следующими возможностями:

- работа с функциями многих переменных различного типа с высокой точностью (не менее $0.5 \cdot 10^{-5}$);
- изображение графика функции одной переменной и линий уровня функции двух переменных;
- включать в себя систему генерации различных видов функций в зависимости от введённого варианта;

Кроме того, для упрощения сопровождения и возможности последующего развития (например, добавление новых методов и алгоритмов при изменении программы курса), программное обеспечение должно быть легко модифицируемым и расширяемым. Этой цели можно достигнуть разделением программного обеспечения на несколько модулей, слабо связанных между собой.

Для осуществления обозначенной задачи служат следующие подзадачи:

1. Разработка общей архитектуры ПО.
2. Разработка и реализация базового модуля.
3. Разработка и реализация графического модуля.
4. Разработка и реализация модуля генерации отчётов.
5. Реализация всех разделов курса "Методы оптимизации".
 - 5.1. Разработка системы вариантов для генерации функций (необходимы для методов минимизации функций).
 - 5.2. Реализация всех 11 методов курса.

2. Анализ предметной области

С использованием [1–4] были изучены методы, рассматриваемые на лабораторных работах курса «Методы оптимизации» и проведён их анализ с целью определения необходимого функционала программного обеспечения и формулировки требований к его архитектуре.

2.1. Градиентные методы поиска экстремума функции

Градиентные методы являются одной из наиболее распространенных групп методов поиска безусловного экстремума. Обозначим через $\nabla f(\mathbf{X})$ вектор-градиент функции $f(\mathbf{X})$. Пусть поиск экстремума начинается с некоторой произвольной точки $\mathbf{X}^{(0)}$. Тогда движение по градиентному методу описывается следующим рекуррентным соотношением: $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - t_k \cdot \nabla f(\mathbf{X}^{(k)})$, $t_k > 0$, $k = 0, 1, 2, \dots$, где t_k – величина шага на k -й итерации.

В различных вариантах градиентного метода используются различные способы выбора скаляра t_k .

Условием останова методов является ограниченность по модулю всех частных производных функции $f(\mathbf{X})$ в точке $\mathbf{X}^{(k)}$ (при заданной точности ε).

Исходя из задач, решаемых градиентными методами, можно сделать следующие заключения:

1. Разрабатываемое программное обеспечение должно обладать возможностью обработки функций многих переменных различных видов.
2. В тех алгоритмах, где скаляр t_k вычисляется на каждом шаге, необходимо решать задачу оптимизации функции одной переменной. Поскольку в зависимости от вида функций и выбора начальной точки значения градиента функции могут дос-

тигать больших значений ($> 10\,000$), возникает проблема точности вычисления скаляра t_k .

2.2. Овражные методы поиска экстремума функции

Градиентные методы показывают низкую эффективность при определении экстремума овражных функций (линии уровня таких функций вытянуты вдоль поверхности, называемой «дном оврага»). Овражные методы позволяют за небольшое количество итераций найти *окрестность* точки экстремума таких функций. Более точный поиск, если это необходимо, проводится с использованием других методов.

Основная идея овражных методов – осуществить спуск ко дну оврага, а затем двигаться вдоль него в направлении точки минимума.

Для реализации овражных методов используются те же структуры данных, что и для градиентных, поэтому их реализация не повлечёт за собой изменений структуры программного обеспечения.

2.3. Методы одномерного поиска экстремума функции

В рамках курса рассматриваются следующие методы.

I. Метод золотого сечения. В основе метода лежит принцип деления отрезка в пропорциях золотого сечения. На отрезке $[a, b]$, внутри которого находится искомым экстремум, выбираются две точки c и d ($c < d$), такие, что отношения длин отрезков $[a, b] : [c, b] = [a, b] : [a, d] = \frac{2}{\sqrt{5}-1}$ (пропорция золотого сечения). Затем в точках c и d сравниваются значения целевой функции: если $f(c) > f(d)$ (в случае поиска минимума), то $a = c$, в противном случае $b = d$.

II. Метод Дэвиса – Свенна – Кэмпбелла. Метод состоит из двух этапов:

1. Поиск отрезка, содержащего искомым экстремум: для заданных x_0 и Δx :

– $x_1 = x_0 + \Delta x$;

– если $f(x_1) \leq f(x_0)$, то $\Delta x = 2 \cdot \Delta x$, иначе $\Delta x = -\Delta x$;

– пока $f(x_{k+1}) < f(x_k)$, $\Delta x = 2 \cdot \Delta x$, $k = k + 1$;

– $x_{k+2} = x_{k+1} - \frac{\Delta x}{2}$.

2. Одна из точек x_{k-1} , x_k , x_{k+1} , x_{k+2} отбрасывается согласно методу золотого сечения, по оставшимся точкам проводится квадратичная интерполяция. Если заданная точность не достигнута, перейти к п. 1.

III. Метод Пауэлла. Метод использует схожий с методом ДСК принцип работы:

1. $x_1 = x_0 + \Delta x$.

2. Если $f(x_1) \leq f(x_0)$, то $x_2 = x_0 + 2 \cdot \Delta x$, иначе $x_2 = x_0 - \Delta x$.

3. Проводится квадратичная интерполяция по точкам x_0 , x_1 , x_2 .

4. Одна из четырёх точек отбрасывается по методу золотого сечения.

5. Если заданная точность не достигнута, перейти к п. 1.

Поскольку функции одной переменной являются подмножеством функций многих переменных, для реализации методов одномерного поиска возможно использование тех же структур данных, что и для градиентных методов.

2.4. Симплекс-метод решения задачи линейного программирования

Каноническая форма записи ЗЛП имеет следующий вид:

$$L(x_1, \dots, x_n) = \sum_{i=1}^n c_i \cdot x_i \rightarrow \min_{x_i, i=1, \dots, n},$$

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = \overline{1, m},$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Множество D векторов $\mathbf{X} = (x_1, x_2, \dots, x_n)$, удовлетворяющих ограничениям и условиям неотрицательности переменных ЗЛП, называют областью определения задачи. Любой вектор \mathbf{X} из D называется допустимым решением или планом ЗЛП.

Идея симплекс-метода заключается в последовательном переборе точек, подозрительных на экстремум, причём при переходе от одной точки к другой значение линейной формы не должно увеличиваться.

Поскольку симплекс-метод работает с матричными структурами, очевидно, что при его реализации необходимо создание новой структуры данных (т.к. структуры данных, применяемые в рассмотренных ранее методах, неприменимы к матрицам).

2.5. Транспортная задача и её решение методом потенциалов

Пусть A_1, A_2, \dots, A_n – поставщики определённого однородного товара, a_i – запасы товара у поставщика A_i , $i = \overline{1, n}$; B_1, B_2, \dots, B_m – потребители этого товара, b_j – потребность в товаре у потребителя B_j , $j = \overline{1, m}$; c_{ij} – стоимость перевозки одной единицы товара от поставщика A_i потребителю B_j . Необходимо перевезти весь товар от поставщиков к потребителям так, чтобы исчерпать все его запасы поставщиков, удовлетворить потребности в товаре у потребителей и при этом минимизировать суммарную стоимость перевозок.

Для решения транспортной задачи методом потенциалов необходимо соблюдение двух условий:

1. Задача является сбалансированной, т.е. суммарные запасы товара всех поставщиков совпадают с суммарными потребностями в товаре всех потребителей.
2. Найден опорный план задачи, например, методом минимального тарифа, при котором на каждом шаге (до тех пор, пока весь товар не будет распределён) выбирается клетка с минимальным значением тарифа и для неё устанавливается максимально возможный объём перевозки.

Для определения потенциалов каждому поставщику A_i и каждому потребителю B_j сопоставляются величины u_i и v_j соответственно так, чтобы для всех базисных клеток плана было выполнено соотношение: $u_i + v_j = c_{ij}$, при этом u_1 полагается равным нулю.

Затем для каждой свободной клетки плана вычисляется величина $\Delta c_{ij} = c_{ij} - (u_i + v_j)$.

План является оптимальным, если все $\Delta c_{ij} \geq 0$. В противном случае для ячейки с отрицательной разностью Δc_{ij} можно построить цикл, в котором кроме этой клетки все остальные являются базисными. Такой цикл всегда существует и единственен. Затем среди чётных элементов цикла пересчёта определяется величина ρ , соответствующая объёму наименьшей из перевозок. После этого происходит переход к следующему опорному плану: объём перевозок нечётных элементов цикла увеличивается на величину ρ , а чётных – уменьшается.

Несмотря на то, что данный метод также работает с данными в виде таблицы (матрицы), использование структуры данных симплекс-метода не представляется возможным, поскольку в транспортной задаче ячейки таблицы имеют более сложную структуру, совмещая в себе величины c_{ij} , Δc_{ij} , x_{ij} (объём перевозки от i -го поставщика j -ому потребителю в опорном плане).

3. Общая архитектура программного обеспечения

В результате анализа предметной области можно сделать вывод, что программное обеспечение должно обладать возможностью обработки различных типов данных (в

случае методов минимизации функций – работа с функциями различных видов, в случае ЗЛП и транспортной задачи – обработка данных в виде таблиц). Поэтому оптимальным решением является разделение множества всех задач на несколько независимых компонент (модулей), каждая из которых содержит реализацию метода или класса схожих методов.

Общая структура программного обеспечения представлена следующей схемой.

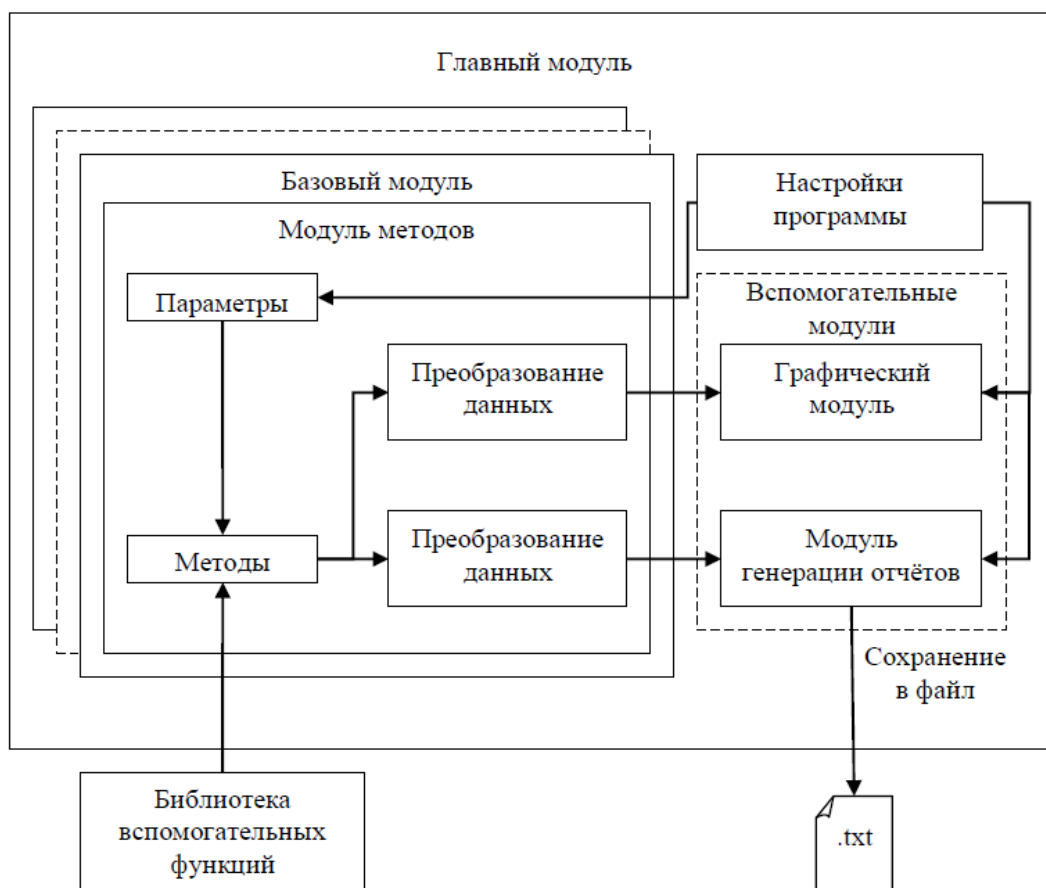


Рис. 1. Архитектура программного обеспечения

Рассмотрим назначение отдельных компонент программного обеспечения более подробно.

1. Основой для модулей методов является т.н. "Базовый модуль", в котором объявлены прототипы функций взаимодействия с модулями других групп. При реализации нового модуля методов он наследуется от базового модуля, и определяет необходимые ему функции. Такое решение позволит дополнительно уменьшить связность компонент между собой, что даёт ряд преимуществ:

- при изменении/добавлении модуля корректировке подвергается только он сам (поскольку интерфейсы взаимодействия при этом не меняются);

– подобное архитектурное решение позволяет с лёгкостью извлечь модуль из программного обеспечения и использовать его независимо/встроить в другое программное обеспечение.

2. Модуль методов содержит реализацию метода или нескольких схожих по решаемым задачам методов. После завершения работы метода модуль также отвечает за преобразование результатов его работы в типы данных, описанные в прототипах функций базового модуля для возможности их последующей обработки вспомогательными модулями.

3. Главный модуль устанавливает связи базовых модулей с вспомогательными модулями, содержит в себе настройки программного обеспечения и отвечает за главный интерфейс программного обеспечения.

4. Особенности реализации программного обеспечения

Связь модулей реализуется посредством системы сигналов и слотов, которую предоставляют библиотеки Qt. Данная система является крайне гибкой и позволяет устанавливать связи вида «один ко многим» и «многие к одному», не сообщая при этом объектам информацию о структуре объектов, с которыми они связаны посредством этой системы.

В целях повышения точности в ходе вычислений выражения для функции, её частных производных и точки минимума задаются аналитически. Для программного вычисления этих выражений используется польская инверсная запись, реализованная со следующим функционалом.

1. Допускаются целые, дробные и отрицательные числа.

2. Допускаются следующие операции: +, −, *, /, ^ (возведение в степень), % (остаток от деления).

3. Переменные-аргументы функции в её аналитическом выражении задаются в формате

«Имя аргументаИндекс аргумента»,

где именем аргумента является английская литера "x", индексом – цифра в диапазоне 1–9.

4. При создании нового варианта аналитический вид функции указывается с точностью до констант. Константы генерируются по формулам, которые задаёт преподаватель. Эти неопределённые константы в аналитическом выражении функции задаются в формате

«Имя константыИндекс константы»,

где именем константы могут являться любые строчные английские литеры (за исключением литеры "x"), индексом – цифра в диапазоне 1–9.

При задании формул возможно использование компонент составного ключа с целью генерации различных видов функций для студентов:

– номер студента в списке группы (имя ключа – "kl", возможный диапазон – 01–50);

– номер группы студента (имя ключа – "kg", возможный диапазон – 01–09);

– год поступления студента (имя ключа – "ky", возможный диапазон – 00–99).

Для работы с графикой используется библиотека QCustomPlot, обеспечивающая масштабирование и перемещение графика.

Для построения линий уровня функции использовался метод сканирования (построчной развертки). Пусть задан набор линий уровня функции: $C_1 < C_2 < \dots < C_N$. Исходная функция "просматривается" с достаточно малыми шагами h_x, h_y строка за строкой в пределах заданных границ $a_x \leq x \leq b_x, a_y \leq y \leq b_y$. При переходе значения функции $z = f(x, y)$ из одного интервала уровней $[C_{i-1}, C_i]$ в другой, на карте изображается точка. Сканирование в алгоритме производится вдоль "строк", параллельных оси Ox . В

связи с этим могут плохо воспроизводиться участки изолиний, близкие к горизонтальным, даже при малых шагах сканирования h_x , h_y . Алгоритм работает сравнительно медленно, поскольку для достаточно качественного "поточечного" воспроизведения изолиний желательно предусматривать мелкую сетку точек сканирования. Однако достоинством этого алгоритма является то, что он строит сразу всё семейство изолиний.

Заключение

Результатом данной работы является программное обеспечение, способное полностью заменить используемое в настоящее время программное обеспечение для лабораторных работ курса «Методы оптимизации». Оно не имеет проблем совместимости с современными ОС, а также может поддерживать несколько платформ. Кроме того, благодаря аналитическим выражениям для частных производных, данное программное обеспечение имеет возможность гораздо точнее вычислять их значения, что позволяет получать траектории методов, наиболее приближенные к оптимальным.

Ввиду наличия файлов с исходным кодом и модульной структуры, данное программное обеспечение имеет большой потенциал модернизации. Используемые архитектурные решения позволяют в короткие сроки расширить охват предметной области программным обеспечением или создать новое программное обеспечение, предназначенное для решения задач из иных предметных областей, (в первую очередь связанных с вычислениями).

ЛИТЕРАТУРА

1. Корицунов Ю.М. Математические основы кибернетики. – М.: Энергоатомиздат, 1972. – 376 с.
2. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.
3. Ахмеров Р.Р. Методы оптимизации гладких функций. [Электронный ресурс] – URL: http://w.ict.nsc.ru/books/textbooks/akhmerov/mo_unicode/index.html (16.06.2018)
4. Метод градиентного спуска [Электронный ресурс]. – URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%81%D0%BF%D1%83%D1%81%D0%BA%D0%B0 (16.06.2018)

ЧИСЛЕННЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ ЭЛЕКТРОМАГНИТНОГО РАССЕЯНИЯ НА ТОНКИХ ПАРАЛЛЕЛЬНЫХ ИДЕАЛЬНО ПРОВОДЯЩЕМ И ДИЭЛЕКТРИЧЕСКОМ ЦИЛИНДРАХ

А.Г. Дмитренко, О.М. Балашова

Томский государственный университет
dmitr@fpmk.tsu.ru, jkz@pisem.net

Введение

Значительный интерес для исследователей представляет изучение рассеяния электромагнитных волн в резонансной частотной области на структурах, состоящих из одного или нескольких тонких цилиндров конечной длины. Этот интерес обусловлен необходимостью решения таких практически важных проблем, как проблемы радиолокационной заметности, идентификации объектов, оценки рассеяния диэлектрическими или металлическими цилиндрическими деталями различных геометрически сложных тел и др.

Под тонким цилиндром обычно понимается цилиндрическое тело, поперечные размеры которого много меньше его длины и длины падающей волны. Анализ имеющейся в распоряжении авторов литературы показывает, что известны работы, например, [1–3], в которых рассмотрено рассеяние электромагнитной волны на одиночном тонком прямолинейном идеально проводящем цилиндре, а также работы, например, [4–6], в которых рассмотрено рассеяние электромагнитной волны на одиночном тонком