

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

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

Томск, 28–30 мая 2020 г.

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

Томск
Издательство Томского государственного университета
2020

ББК 22.17–22.19
УДК 519.2, 519.7, 519.8
T78

**ЧЛЕНЫ КОЛЛЕГИИ, РУКОВОДИТЕЛИ НАУЧНЫХ РЕДАКЦИЙ
ПО НАПРАВЛЕНИЯМ:**

д-р техн. наук, проф. **А.А. Глазунов** – научная редакция «Механика, математика»; д-р техн. наук, проф. **Э.Р. Шрагер** – научная редакция «Механика, математика»; д-р техн. наук, проф. **А.М. Горцев** – научная редакция «Информатика и кибернетика»; д-р техн. наук, проф. **С.П. Сущенко** – научная редакция «Информатика и кибернетика»; д-р физ.-мат. наук, проф. **В.Г. Багров** – научная редакция «Физика»; д-р физ.-мат. наук, проф. **А.И. Потекаев** – научная редакция «Физика»; д-р биол. наук, проф. **С.П. Кулижский** – научная редакция «Биология»; д-р геол.-минер. наук, проф. **В.П. Парначев** – научная редакция «Науки о Земле, химия»; канд. хим. наук, доц. **Ю.Г. Слижов** – научная редакция «Науки о Земле, химия»; д-р филол. наук, проф. **Т.А. Демешкина** – научная редакция «История, филология»; д-р ист. наук, проф. **В.П. Зиновьев** – научная редакция «История, филология»; д-р экон. наук, проф. **В.И. Канов** – научная редакция «Юридические и экономические науки»; д-р юрид. наук, проф. **В.А. Уткин** – научная редакция «Юридические и экономические науки»; д-р ист. наук, проф. **Э.И. Черняк** – научная редакция «Философия, социология, психология, педагогика, искусствознание»; д-р психол. наук, проф. **Э.В. Галажинский** – научная редакция «Философия, социология, психология, педагогика, искусствознание»

НАУЧНАЯ РЕДАКЦИЯ ТОМА:

д-р техн. наук, проф. **А.М. Горцев**, д-р техн. наук, проф. **С.П. Сущенко**, д-р физ.-мат. наук, доц. **Ю.Г. Дмитриев**, д-р физ.-мат. наук, доц. **С.П. Моисеева**, д-р физ.-мат. наук, проф. **В.В. Конев**, д-р техн. наук, проф. **А.Ю. Матросова**, д-р техн. наук, проф. **А.А. Назаров**, д-р техн. наук, проф. **К.И. Лившиц**, канд. техн. наук **С.А. Останин**, канд. физ.-мат. наук **А.С. Морозова**, канд. техн. наук **А.С. Шкуркин**, канд. техн. наук **И.С. Шмырин**.

T78 Труды Томского государственного университета. – Т. 305. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем : материалы Международной научной конференции. Томск, 28–30 мая 2020 г. / под общ. ред. И.С. Шмырина. – Томск : Издательство Томского государственного университета, 2020. – 322 с.

ISBN 978-5-94621-970-9

Сборник содержит материалы Международной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем», проводившейся 28–30 мая 2020 г. на базе Института прикладной математики и компьютерных наук Томского государственного университета. Материалы сгруппированы в соответствии с работавшими на конференции секциями.

Для научных работников, преподавателей, аспирантов, магистрантов и студентов.

УДК 539.3.004
ББК 22,25.22.251.22.62

ISBN 978-5-94621-970-9

© Томский государственный университет, 2020

II. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ИХ ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ

СОЗДАНИЕ ОБУЧАЮЩЕЙ ПРОГРАММЫ ДЛЯ ФОРМИРОВАНИЯ НАВЫКА ОПРЕДЕЛЕНИЯ ХАРАКТЕРИСТИК КРИВЫХ ВТОРОГО ПОРЯДКА И ИХ ПОСТРОЕНИЯ

Багдалов П.Д., Пахомова Е.Г.

Томский государственный университет
pablobagdal@gmail.com, peg@tpu.ru

Введение

Компьютерные технологии занимают всё большее место в жизни человека. В частности, в области образования. Мы все свидетели того, как люди, несмотря на пандемию, стремятся делиться знаниями и развиваться. Но не всем удаётся возобновить полноценный учебный процесс. В частности, проверку знаний студентов и анализ их ошибок. В привычном формате лицом к лицу преподаватель тратит время на каждого студента отдельно, сталкиваясь с типичными ошибками и объясняя их причину. В дистанционном формате по аудио/видео связи объяснить то же самое гораздо сложнее, а если приходится писать – у преподавателя и вовсе не останется личного времени. Что, в свою очередь, негативно влияет на продуктивность.

Компьютер – это, в первую очередь, быстрый доступ к огромному объёму информации, её обработке, обмену и хранении. Так, имея доступ к сети общего пользования или загрузив данные с внешних носителей, любому человеку предоставляется возможность самостоятельно развивать свои навыки в той или иной области.

На данный момент, обладая базовыми навыками работы с компьютером, человек может найти в Интернет ответ практически на любой вопрос и этот факт на первый взгляд обещает блестящие перспективы дистанционному образованию. Но более пристальное изучение вопроса показывает, что информация в сети часто противоречива, в том числе в разделе точных наук, например в математике. Поэтому полностью самостоятельное самообразование посредством онлайн-платформ и тренажёров крайне затруднительно. Обучающийся может встретить в сети не вполне корректный алгоритм, или по неопытности переложить алгоритм, применимый для одного типа задач на другие, где он вовсе недопустим.

И всё-таки обучение с использованием программ и онлайн ресурсов допустимо в разумных пределах. На мой взгляд, если мы не подвергаем сомнению авторитетности университетского образования, то и знания преподавателей принимаем как достоверные. Поэтому допустимо использование, в первую очередь, тех ресурсов, которые рекомендуются непосредственно преподавателями.

Предлагаемое решение призвано улучшить жизнь студентов и преподавателей. Много времени уходит впустую. Почему так происходит? Ответ довольно простой: мы не пользуемся теми возможностями и технологиями, которые у нас есть. Речь об обучающих программах, которые могут если не заменить преподавателя, то частично делегировать его обязанности по проверке знаний студентов, избавляя от рутинной работы с однотипными ошибками.

Именно желание делегировать компьютеру частично или полностью такие функции преподавателя, как проверка полученных знаний и анализ ошибок обучающегося, привело к появлению программ, которые называются обучающими.

Существуют несколько видов обучающих программ: контролирующие, наставнические, имитационные и моделирующие, а также развивающие игры [1].

Контролирующие программы предназначены для проверки усвоенного учащимся материала. Как правило, такие программы в случайном порядке генерируют вопросы и задачи из некоторого банка заданий и сверяют введенный ответ с верным ответом.

Наставнические программы, это программы, которые предоставляют учащемуся теоретический материал, разбитый на блоки, и проверяют усвоение этого материала, предлагая в конце каждого блока ответить на контрольные вопросы. Ответы учащегося анализируются, и на основе этого анализа определяется следующий шаг в обучении. Таким образом эти программы осуществляют управление ходом обучения. Среди наставнических программ выделяют линейные, разветвленные, адаптивные и комбинированные. Линейная программа состоит из небольших последовательно сменяющихся блоков информации с контрольными вопросами. По результатам ответов учащийся либо возвращается к началу блока (среди ответов есть неверные), либо переходит к изучению следующего блока (все ответы верны). Разветвленные программы отличаются от линейных тем, что обучающемуся при неправильном ответе даётся возможность воспользоваться дополнительной информацией и исправить ответ. Адаптивные программы предполагают выбор уровня сложности учебной информации, возможность при необходимости обращаться к источникам информации. Комбинированные включают в себя компоненты линейных, разветвленных и адаптивных программ. Среди коммерческих обучающих программ наиболее распространены именно комбинированные программы.

Главное отличие контролирующей программы от наставнической: контролирующая программа способна только определить правильность ответа обучающегося, наставническая программа – способна проанализировать ответ учащегося и дать рекомендацию по дальнейшему ходу обучения.

Имитационные и моделирующие – позволяют использовать компьютер для эксперимента. Вводя команды с клавиатуры, обучающийся управляет ходом эксперимента.

Развивающие игры – создают определенную виртуальную среду и возможность влиять на эту среду со стороны пользователя, тем самым изучая эту среду. Чаще всего виртуальная среда частично или полностью моделирует наш реальный мир. Среди всех обучающих программ, развивающие игры самые сложные в реализации. Их создание требует совместной работы программистов, специалистов в предметной области, психологов [2].

В интернете для студентов, изучающих математику, предлагается множество онлайн-калькуляторов, дающих готовое решение. И они прекрасно выполняют свою задачу, если интересует только ответ, но они не решают главной задачи обучающей программы – формирование устойчивого навыка, который формируется только в том случае, когда обучающийся сам решает задачу, а не разбирает готовое решение. Поскольку процесс решения задачи, как правило, сопровождается ошибками, особенно на начальном этапе изучения темы, и эти ошибки для студента не очевидны, то наиболее эффективными будут обучающие программы наставнического типа, которые проанализируют ответ студента и дадут ему рекомендацию по дальнейшим действиям.

В области математики, по сравнению с количеством онлайн-калькуляторов, к использованию которых студенты часто прибегают, крайне мало наставнических программ. В частности, в области формирования навыка определения характеристик кривых второго порядка и их построения программ обнаружено не было ни в русскоязычном, ни в англоязычном интернете, в связи с чем было принято решение о разработке такой программы.

Следует заметить, что не для любого раздела математики можно создать эффективную обучающую программу. Это возможно только для тех разделов, где приобретение навыка связано с реализацией вычислительных алгоритмов. К таким разделам относится линейная алгебра, векторная алгебра, аналитическая геометрия и некоторые разделы математического анализа.

1. Постановка задачи

Цель работы – создать обучающую программу наставнического типа, которая бы позволяла приобрести навык определения характеристик кривых второго порядка и их построения.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ типовых ошибок, допускаемых при определении характеристик и построении.
2. Реализовать класс `SecondOrderCurve`, определяющий базовую структуру кривых.
3. Реализовать классы наследники `Ellipse`, `Circle`, `Hyperbola` и `Parabola`, отражающие специфику каждого типа кривой и генерирующие соответствующие значения параметров уравнения.
4. Реализовать выявление следующих ошибок студента:
 - а) ошибка определения названия кривой;
 - б) ошибка определения формы кривой;
 - в) ошибка соответствия введённого названия кривой к выбранной графической форме;
 - г) ошибка знака координат центра или вершины;
 - д) ошибка выделения полного квадрата;
 - е) арифметическая ошибка;
 - ж) ошибка определения полуосей или других параметров.
5. Реализовать пользовательский интерфейс включающий в себя:
 - а) главное меню;
 - б) справочный материал;
 - в) разобранный пример;
 - г) тренажер (генерация уравнения, поля для ввода ответа).

2. Анализ типовых ошибок студента

Поскольку определение характеристик кривых второго порядка – это определенный алгоритм, то если студент совершает ошибку, это значит, что он выполнил неверный шаг алгоритма или несколько шагов. Комбинаций ошибок может быть большое количество и все проанализировать крайне сложно, но, как показывает практика, наиболее часто студенты совершают следующие ошибки:

- а) неправильно определяет тип кривой;
- б) ошибается в знаках координат центра (вершины);
- в) путает оси;
- г) неправильно выделяет полный квадрат.

После того, как студент вводит ответ, программа сравнивает ответ студента с данными, полученными из сгенерированного объекта и, если ошибок нет, то сообщает об этом пользователю. Если ошибки есть, то для выявления каждой из ошибок используется специальный метод. Рассмотрим каждый из них подробнее.

1) Для выявления ошибки названия кривой, идёт сравнение выбранного пользователем варианта с тем, что прикреплено к сгенерированному объекту. Если форма кривой при этом будет определена правильно, то студент получит уведомление: «Неверное название».

2) Для выявления ошибки определения формы кривой используется аналогичный элемент объекта. Если пользователь правильно определил название, но ошибся в определении формы кривой, то получит уведомление «Неверная форма кривой», а если ошибся в обоих случаях, то идёт проверка соотношения введённого названия с указанной формой кривой. Если соответствие установлено, но пользователь получит уведомление: «Неверная кривая», а если указанная форма кривой не будет соответствовать

названию кривой, то получит уведомление: «форма кривой не соответствует указанному названию».

3) Для выявления ошибки знака координаты введённое значение сравнивается с член-данным объекта, хранящим координаты центра (фокусов). Студент получает соответствующее уведомление: «Ошибка знака в координатах центра (фокусов)».

4) Маркером ошибки выделения полного квадрата является не совпадение абсолютных величин координат центра (вершины). При наличии этого маркера, программой генерируется «неверные ответы» и сравнивает их с введенными студентом данными. Их совпадение означает, что такого рода ошибка действительно допущена и пользователь получает уведомление «Ошибка выделения полного квадрата». Ошибка выделения полного квадрата наиболее частая арифметическая ошибка, но возможны и другие типы арифметических ошибок. Поэтому, если «неверные ответы» не совпали с ответами студента, то выводится сообщение об арифметической ошибке, хотя в перспективе возможно внедрить и другие более редкие, но тоже возможные ошибки.

3. Интерфейс программы

В текущей реализации рассматривается обработка только эллипса. Все варианты допускаемых ошибок и последствий учитываются в построенной блок-схеме (рис. 1, 2)

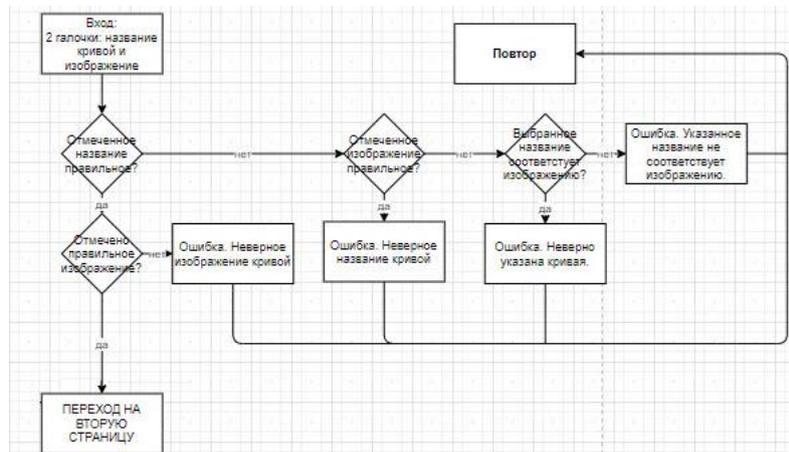


Рис 1. Блок-схема логики обработки ошибок первой страницы упражнений

Интерфейс программы интуитивно понятен. Это важно для обучающей программы и влияет на ее эффективность [3]. Работа программы начинается с открытия главного меню (рис. 3). В главном меню студенту предлагается выбрать следующие действия:

- 1) Начать
- 2) Теория
- 3) Пример
- 4) Выход

Выбор действия осуществляется нажатием соответствующей кнопки.

Кнопки «Теория» и «Пример» однотипные и предоставляют возможность ознакомления с теоретическим материалом и разобранным примером. По окончании работы с соответствующими окнами происходит возврат в главное меню. Такая организация сделана для того, чтобы во время выполнения собственного задания студент не переписывал решение по образцу, а думал самостоятельно.

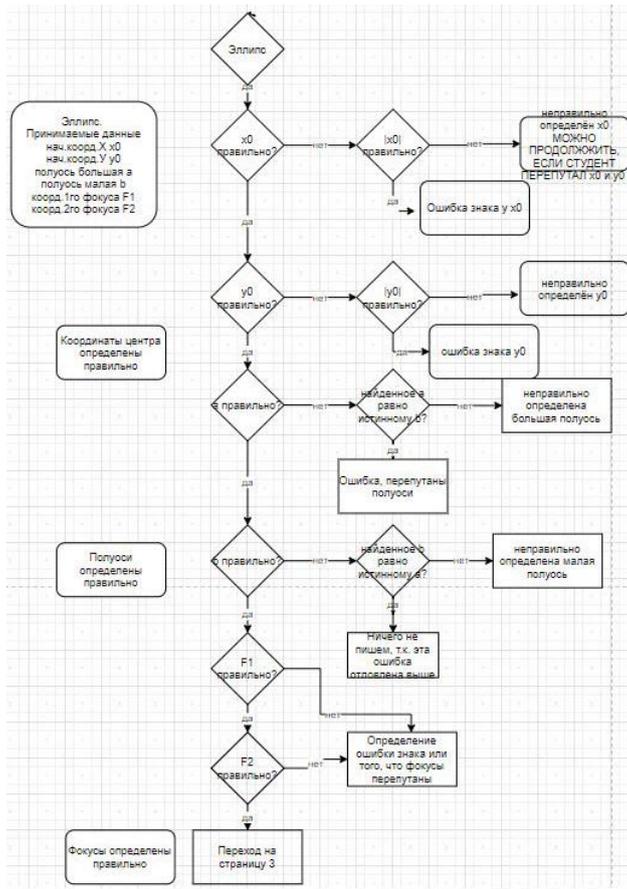


Рис 2. Блок-схема логики обработки ошибок второй страницы упражнения (для эллипса)

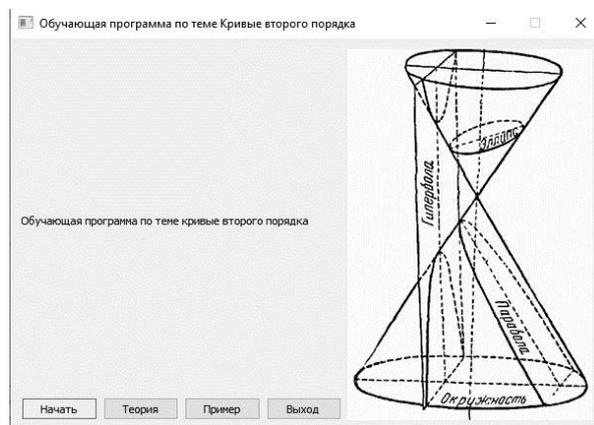


Рис 3. Главное меню обучающей программы

Кнопка «Начать» приводит на единый для всех типов кривых экран: выбор названия кривой и её формы (рис. 4,5) Уже на этом этапе, по опыту преподавателей, у обучаемых возникают сложности. При допущении ошибки мы дадим понять, какого рода ошибка допущена и предложим изучить теорию.

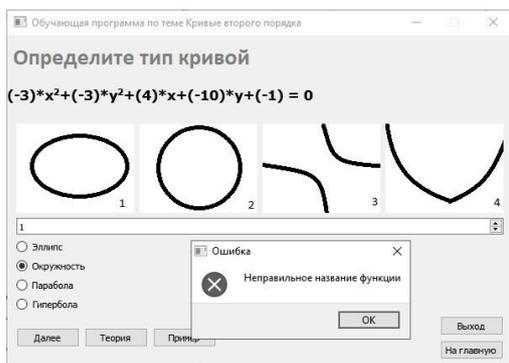


Рис 4. Первое окно программы. Ошибка названия

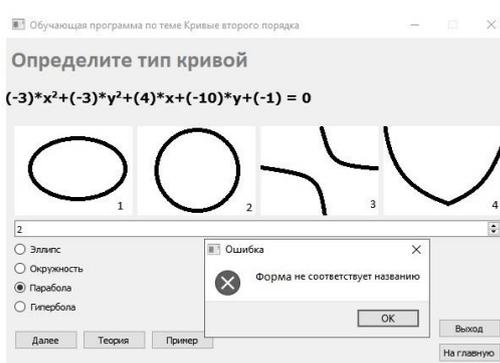


Рис.5. Первое окно программы. Ошибка соответствия

Кнопка «Далее» переводит на специализированный для каждой кривой экран (рис. 6). В нём студент задаёт все запрашиваемые параметры и характеристики. Нередки случаи, когда решение правильно, но в начале допущена ошибка знака или перепутаны полуоси, вследствие чего перепутаны координаты фокусов. Или, наоборот, оси определены правильно, но при дополнении до полного квадрата – арифметическая ошибка (рис. 7)

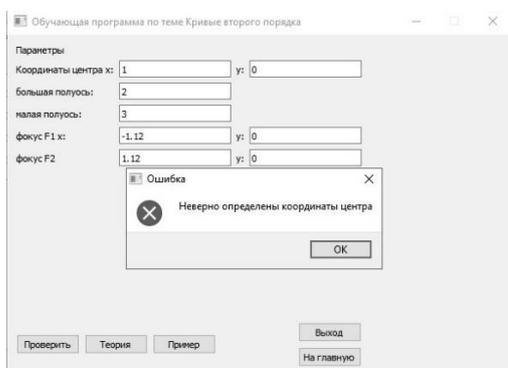


Рис 6. Ошибка знака координат

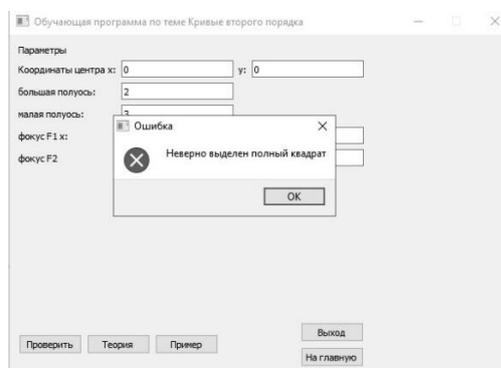


Рис.7. Ошибка выделения полного квадрата

Преподаватели с большим стажем знают все типичные ошибки студента в своей области. Обработывая в программе такие сценарии, мы поможем как студенту, так и преподавателю. Студент сможет быстрее понять, в чём он ошибается и на что следует обратить внимание. Это позволит быстрее сформировать необходимый ему навык, т.к. в случае неудачного решения задачи место ошибки и ее тип определяется немедленно, а значит можно исправить ошибку и продолжить дальнейшую работу. А преподаватель избавляется от необходимости выявлять в работах студентов типовые ошибки. Это уменьшает нагрузку на преподавателя, ведь значительную часть работы преподавателей занимают проверка работ студентов и работа со студентами над их ошибками.

При этом надо отметить, что полностью заменить преподавателя программа не сможет. Она выявляет именно типичные ошибки. Но существуют и другие ошибки. Например, ошибки, которые связаны с непониманием студентом теоретического материала. И здесь не поможет программа, здесь необходим человек, который поймет, в чем проблема и сможет помочь эту проблему преодолеть.

Заключение

В результате выполненной работы по созданию обучающей программы был проведен анализ существующих обучающих программ, анализ типовых ошибок, допускаемых при определении характеристик кривых второго порядка и их построении, реали-

зован класс SecondOrderCurve, Ellipse, создана работающая обучающая программа. Продолжается разработка классов Circle, Hyperbola, Parabola, будет произведена доработка существующего алгоритма, дизайна, а также адаптация по сложности. Такие программы нужны, их нужно развивать и популяризировать.

ЛИТЕРАТУРА

1. *Гефан Г.Д., Кузьмин О.В.* Методика построения контрольно-обучающих программ и их использование в преподавании математических дисциплин // Вестник Бурятского государственного университета. 2013. № 15. С. 23–28.
2. *Аязбаев Т.Л., Галагузова Т.А.* Технология создания компьютерных обучающих программ // Международный журнал экспериментального образования. 2015. – № 3 (часть 2) – С. 76-78.
3. *Афанасьев В.В., Тыщенко О.Б., Афанасьева И.В.* Анализ показателей эффективности обучающих программ // I Всероссийская научно-техническая конференция 'Компьютерные технологии в науке, проектировании и производстве'. Тезисы докладов, часть V, Нижний Новгород, 1999, 43с., стр. 15.

ДВУХКОМПОНЕНТНОЕ РАЗЛОЖЕНИЕ КАРДИОЛОГИЧЕСКОЙ КРИВОЙ

Безходарнов Н.И., Самохина С.И.

*Томский государственный университет
nblaaa@mail.ru*

Введение

Ещё с давних времён людей беспокоил вопрос о работе сердца. Оно является жизненно необходимым органом, по этой причине исследование его работы очень важно для всего человечества. Одним из аспектов является изучение сердечных мышц, которые выполняют основную работу. При обследовании таких мышц строят кардиологическую кривую, для анализа которой необходимо разложение на две компоненты [1,2], имеющие экспоненциальный вид. Данная статья и повествует об одном из способов решения данной проблемы.

1. Постановка задачи

Рассматривается кардиологическая кривая, которая задаётся последовательностью точек на плоскости. Чтобы провести её анализ, требуется получить разложение на две компоненты. Ставится задача получения аппроксимации данной кривой двумя компонентами экспоненциального вида [1–3].

Целью данного проекта является разработка приложения, реализующего разложение кардиологической кривой в функцию следующего вида:

$$y = a_1 e^{-b_1(x-c_1)^2} + a_2 e^{-b_2(x-c_2)^2} . \quad (1)$$

Для достижения данной цели были сформулированы следующие подзадачи:

1. Изучение теоретического материала по аппроксимации функций и нахождению точек минимума и максимума.
2. Выбор метода аппроксимации функции для реализации алгоритма.
3. Реализация алгоритма разложения

После выполнения данных пунктов была написана программа с удобным графическим интерфейсом, которая решает поставленную задачу.

2. Функциональные возможности программы

Реализация была выполнена в среде разработки Qt Creator с использованием библиотек Qt [4]. Данная среда является удобной в разработке графического интерфейса, а также способна обеспечить кроссплатформенность. Библиотеки Qt предоставляют широкий спектр возможностей для разработки различного рода приложений.