УДК 372.851+004.02

ЛАБОРАТОРНЫЕ РАБОТЫ ПО МАТЕМАТИКЕ ДЛЯ СТУДЕНТОВ-ПРОГРАММИСТОВ

Шилов Н.В.[1], Конюхов И.В.[2] Холодов Я.А.[3], Ягудин З.Р.[4], Масягин С.В.[5]

[1,2,3,4,5] *АНО ВО «Университет Иннополис», Иннополис;*

[1]n.shilov@innopolis.ru, [2]i.konyukhov@innopolis.ru, [3]ya.kholodov@innopolis.ru, [4]z.yagudin@innopolis.university, [5]s.masiagin@innopolis.ru

Аннотация

В данной работе рассматриваются особенности преподавания математических дисциплин в АНО ВО «Университет Иннополис», где выпускники получают специальность 09.03.01 «Информатика и вычислительная техника». Рассматривается подход к унификации лабораторных работ по математическим дисциплинам с предметами, связанными непосредственно с разработкой программного обеспечения, с целью снижения общей учебной нагрузки в течение первых семестров бакалаврской программы, а также опциональные лабораторные работы, призванные популяризировать как математику, так и искусство программирования.

Ключевые слова: *математический анализ, дискретная математика, линейная алгебра, дифференциальные уравнения, преподавание математических дисциплин, преподавание основ программирования, алгоритмы и структуры данных*

Преподаватели математики в ВУЗах обычно «мотивируют» студентов-программистов на изучение математики сложившейся культурной и образовательной традицией, эффективным обучением думать и практической целесообразностью/необходимостью (включая личное трудоустройство, а также приложения математики в современных науке и технологиях). Однако, зачастую преподаватели математики забывают о возможности интегрировать преподавание математики с освоением студентами-программистами современного (наукоемкого и технологического) программирования. В настоящей статье авторы делятся некоторым личным опытом организации «программистских лабораторных работ по математике» в рамках курсов линейной алгебры, математического анализа, дискретной математики и дифференциальных уравнений в ИТ-вузе «Университет Иннополис».

Целью работ является освоение на практике (основного или дополнительного) материала курса с использованием «правильных» методов и технологий программирования. Наиболее удачной (с точки зрения организации учебно-образовательного процесса и организации труда и отдыха студентов) авторы считают ситуацию, когда лабораторные работы по математике засчитываются и как лабораторные работы по таким базовым программистским курсам. Например, для того чтобы решить проблему высокой нагрузки студентов первого курса, связанную с преподаванием большого количества предметов одновременно и, следовательно, большого числа домашних заданий, срок сдачи которых выпадает в семестре примерно на одно и то же время, преподавателями дисциплин «Линейная алгебра» и «Введение в программирование» было принято решение об унификации и объединении лабораторных работ по этим курсам.

ОБЯЗАТЕЛЬНЫЕ ЛАБОРАТОРНЫЕ РАБОТЫ В КУРСАХ «ЛИНЕЙНАЯ АЛГЕБРА» И «ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ»

Курс «Линейная алгебра» в Университете Иннополис читается студентам бакалавриата во втором семестре первого года обучения. Предполагается, что за первый семестр они уже успели познакомиться с основами аналитической геометрии, понимают основные концепции векторных пространств, умеют пользоваться базовыми приемами решения систем линейных уравнений размерностью до 4х4, знают, как построить такие системы, в каких задачах они возникают и т. д. Таким образом, курс «Линейная Алгебра», в свою очередь, уже нацелен на углубленное изучение математических основ решения систем линейных уравнений высоких порядков. Преподавателями курса большое внимание уделяется эффективности применяемых алгоритмов, оценке их вычислительной сложности, а также особенностям компьютерной реализации. Очевидно, что подобное невозможно без применения современные языков программирования. Важно отметить, что решение задач в данном курсе не требует написания «больших» программ, что делает код «обозримым» и не требует длительного времени для его разработки.

Курс «Введение в программирование» в АНО ВО «Университете Иннополис» является базовым курсом бакалаврской программы и читается в течение всего первого года обучения. В рамках первого семестра студенты знакомятся с парадигмой объектно-ориентированного программирования на высокоуровневом языке Eiffel. Преимуществом такого подхода является то, что студенты, которые, как правило, уже знакомы с базовыми приемами процедурного программирования на других распространенных языках, учатся декомпозировать задачу не на набор отдельных команд и функций, а на отдельные классы, объекты и их отношения. Учащиеся знакомятся с базовыми принципами такого подхода (инкапсуляцией, наследованием, полиморфизмом), особенностями взаимодействия объектов в программе (ассоциацией, композицией, агрегацией), изучают приемы написания безопасного кода с использованием механизма контрактов. Во втором семестре, в свою очередь, студенты изучают элементы низкоуровневого программирования на языках C и C++. Основными моментами здесь являются изучение механизма ссылок и указателей, позволяющими тонко управлять процессом распределением памяти, а также изучение более сложных программных конструкций, таких как перегрузка операторов, исключения и механизм шаблонов.

Одной из первых тем курса «Линейная алгебра» является решение систем линейных алгебраических уравнений методом Гаусса и Гаусса-Жордана. Математические операции, необходимые для приведения исходной матрицы коэффициентов системы уравнений к единичному виду, легко можно записать в форме умножения матриц, используя специальные матрицы исключения, перестановки и масштабирования. Стоит заметить, что подобные матрицы являются производными от единичной матрицы. Здесь целесообразно сразу попросить студентов написать программную реализацию класса «Прямоугольная матрица» на языке C++, реализовать стандартные математические операции сложения, умножения, вычитания, транспонирования, учитывая все особенности таких операций и, при необходимости, генерируя соответствующие исключения. Кроме того, сразу возникает вопрос типизации данного класса: с каким конкретным типом данных мы работаем? Прослеживается прямая связь с темой механизма шаблонов из курса «Введение в программирование», студентам предлагается реализовать собственный класс рациональных чисел со всеми необходимыми операциями и, используя механизм шаблонов, поработать как с вещественнозначными матрицами, так и с матрицами рациональных чисел.

Матрицы исключения, перестановки и масштабирования являются, как уже было сказано, производными от единичной матрицы, которая, в свою очередь, является квадратной матрицей с равным количеством строк и столбцов. Студенты должны реализовать соответствующие классы «Квадратная матрица», «Единичная матрица», «Матрица исключения», «Матрица перестановки», «Матрица масштабирования», используя механизм наследования. Для вычисления определителя и обращения матрицы можно также попросить использовать уже реализованные ранее алгоритмы. В этом задании студенты также знакомятся с особенностями полиморфизма объектов в языке C++.

На основе математических операций, описанных выше, во втором задании студентам предлагается реализовать прямой метод решения систем линейных уравнений методом Гаусса-Жордана, а также выполнить декомпозицию матрицы на произведение нижнетреугольной и верхнетреугольной матриц.

Следующее задание посвящено приближенному решению систем линейных уравнений методом Якоби. Здесь опять же студенты работают не с процедурной реализацией алгоритма, а в объектно-ориентированном стиле, при котором необходимо определить новый класс «Вектор», для которого определить операцию вычисления нормы (длины). Подобный подход позволяет сформировать устойчивый навык мышления в терминах объектов и отношений, нежели функций и процедур.

В четвертом задании необходимо реализовать метод наименьших квадратов, который является базовым методом регрессионного анализа, применяющемся для оценки неизвестных параметров регрессионных моделей по выборочным данным. Метод наименьших квадратов основан на минимизации суммы квадратов отклонений некоторых функций от искомых переменных и может использоваться для аппроксимации точечных значений некоторой функции. В данном задании особенно полезной является графическая визуализация полученного результата. С этой целью студентам предлагается использовать библиотеку GNU Plot. На график наносится исходное множество («облако») точек и кривая искомой функции, что позволяет наглядно продемонстрировать смысл метода.

Курс «Дифференциальные уравнения» в АНО ВО «Университет Иннополис» также является базовым курсом бакалаврской программы и читается в течение первого семестре второго года обучения. Данный курс является более прикладным курсом, нежели классической математической дисциплиной, и направлен на то, чтобы продемонстрировать студентам, тот факт, что большое количество задач современной науки сводится к математическим моделям, записанным с использованием дифференциальных уравнений разных типов. Кроме того, в рамках данного курса вводятся базовые понятия численных методов, такие как сеточная аппроксимация, разностные аналоги производных, а также демонстрируются методы Эйлера и Рунге-Кутта численного решения обыкновенных дифференциальных уравнения первого порядка.

Во время вычислительного практикума студентам предлагается написать программу, содержащую графический пользовательский интерфейс для взаимодействия с пользователем и реализацию численных методов решения обыкновенных дифференциальных уравнений, описанных выше. С помощью разработанного программного обеспечения студенты должны выполнить анализ данных методов, а именно, оценить погрешность аппроксимации на различных шагах сетки.

Особое внимание опять же уделяется вопросам применения объектно-ориентированного подхода к разработке архитектуры приложения. Студенты второго курса параллельно посещают спецкурс «Шаблоны и принципы проектирования», где знакомятся с основами взаимодействия объектов в программе (построения взаимосвязей классов, объектов и интерфейсов). Курс построен на классической книге Э. Гамма, Дж. Влисидиса, Р. Хелма и Р. Джонсона «Шаблоны проектирования» [1], в которой приводятся различные способы построения архитектуры объектно-ориентированного приложения. Кроме того, большое внимание уделяется SOLID- принципам проектирования.

Как нельзя кстати эти методы приходятся к вычислительному практикуму по курсу «Дифференциальные уравнения». Не допускается применение в архитектуре программы так называемых «суперклассов», которые отвечают за весь необходимый функционал. Важно строго следить за соблюдением принципа единственной обязанности классов, когда каждый класс является отдельной функциональной единицей. Принцип подстановки Лисков можно удачно продемонстрировать при использовании унифицированного вызова конкретных реализаций численных методов решения. Также легко можно показать, как применяется принцип открытости/закрытости, когда программные сущности должны быть открыты для расширения, но закрыты для модификации, что буквально означает то, что программа собирается из небольших блоков подобно конструктору, причем каждый блок отвечает за целостность хранимой внутри него информации.

ОПЦИОНАЛЬНЫЕ ЛАБОРАТОРНЫЕ РАБОТЫ К КУРСАМ «МАТЕМАТИЧЕСКИЙ АНАЛИЗ» И «ДИСКРЕТНАЯ МАТЕМАТИКА»

Курс «Математический анализ» в АНО ВО «Университет Иннополис» читается в первом и втором семестрах первого года обучения. Лабораторные работы к этому курсу носят опциональный характер, они рассчитаны на добровольцев, готовых выполнить их во время 3-недельной летней учебной практики, но имеют образовательное значение: они призваны помочь студентам понять разницу между «идеальным» вычислительным математическим методом и его «реальным» воплощением в виде компьютерной программы, вызвать интерес к изучению дополнительных тем по курсу и решению интересных программистских задач.

Приведем примеры тем лабораторных работ к курсу анализа: «точное» (точнее, как можно более точное) вычисление частичных сумм гармонического ряда в границах, представимых типом int; вычисление значений логарифмической функции или тригонометрических функций для аргументов в некотором заданном диапазоне с некоторым шагом.

Кратко остановимся на одной из этих лабораторных работ – лабораторной работе на вычисление косинуса.

Как известно (и доказывается в курсах математического анализа) $\cos x = \sum_{n\in \boldsymbol{N}}(-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + \cdots + \frac{x^{4n}}{(4n)!} - \frac{x^{4n+2}}{(4n+2)!} + \cdots$ для любого $x \in \boldsymbol{R}$, причем, (в силу теоремы Лейбница для знакопеременных рядов) сумма остаточных членов по абсолютной величине не превосходит последнего суммированного члена ряда (что в данном случае можно записать как $\left|\sum_{m>n}(-1)^m \frac{x^{2m}}{(2m)!}\right| \le \frac{x^{2n}}{(2n)!}$ для любого $n \ge 0$). Задание лабораторной работы состоит в том, чтобы, используя данное разложение в ряд функции $\cos x$, вычислить ее значения с точностью 0.000001 для аргументов в диапазоне от 0 до 50 радиан с шагом 0.5 радиан. Цель лабораторной работы – познакомиться на практике с вычислениями (аналитических) функций на основе разложения в степенной ряд, с возникающими при этом сложности из-за разности между «идеальной» арифметикой вещественных чисел и машинной арифметикой в формате с плавающей запятой и связанных с ней ошибок округления.

Выполнение лабораторной работы начинается с дизайна алгоритма вычисления функции $\cos x$. На основании разложения в ряд легко прийти к очень простому алгоритму вычисления функции $\cos x$ с заданной точностью $\varepsilon > 0$, который легко реализовать программно и положить в основу лабораторной работы. Однако, если «запустить» эту программу, то получается «неожиданные» результаты: реализованная нами функция на значениях аргумента от 0 до 5 совпадает со стандартной функцией (с точностью, представимой типом данных float), на значениях аргумента где-то от 5 и до 15 отличается от значений стандартной функции на доли процента, на значениях аргумента после 15 и до 20 отличия достигают десятков процентов, а на значениях после 20 радиан значения реализованной функции вообще теряют смысл так как уходят за пределы интервала $[-1, +1]$ и достигают нескольких тысяч!

Тут на выручку приходит то, что математическая функция $\cos$ имеет период 2π: так как из экспериментов мы знаем, что реализованная нами функция для аргументов 0 до 5 радиан совпадает со стандартной функцией, то вместо вычислений для произвольного аргумента x из диапазона 0 до 50 радиан надо вычислять функцию для аргументов z из диапазона -5 до $+5$ радиан такого, что $x = z + 2k\pi$ для некоторого целого k. В частности, если принять 6.2831853072 в качестве приближенного значения 2π, то получим результаты с ошибкой вычислений (по сравнению к стандартной функции) составляет десятитысячную долю процента во всем диапазоне от 0 до 50 радиан.

Таким образом, при выполнении данной лабораторной работы студент может лучше понять разницу между вычислительным методом и его компьютерной реализацией (обусловленную в данном случае форматом используемых в компьютере чисел), проявить инженерные способности (как сконструировать алгоритм и критерии его корректности), развить навыки экспериментатора и тестировщика.

Теперь перейдем к лабораторным работам к курсу «Дискретная математика». Этот курс в АНО ВО «Университет Иннополис» читается в первом. Главная цель курса этого курса – познакомить студентов с «доказательством», как главным методом науки Математики и ее самыми базовыми понятиями такими как *множества*, *конечная комбинаторика*, *отношения*, *функции*, *графы*, *алгебраические структуры*, *логические формализмы*.

Программистским «напарниками» (и продолжением) курса «Дискретная математика» выступает курс по «Алгоритмы и структуры данных», который читается в АНО ВО «Университет Иннополис» во втором семестре первого года обучения (и в котором множество лабораторных работ – особенно по представлению и обходам графов). Поэтому лабораторные работы к курсу «Дискретная математика» в Университете Иннополис носят опциональный характер (так же как лабораторные работы к курсу «Математический анализ»), они рассчитаны на добровольцев, готовых выполнить их во время 3-недельной летней учебной практики. Эти работы призваны помочь студентам познакомиться с дополнительным материалом и заняться решением интересных программистских задач.

Приведем примеры тем лабораторных работ к курсу дискретной математики: визуализация и строгое математическое объяснение алгоритмических карточных фокусов; реляционная семантика программ над булевскими переменными; экспериментальная проверка законов для «почти всех графов» (например, что почти все графы являются связными).

Кратко остановимся на одной из этих лабораторных работ – лабораторной работе по экспериментальной проверке того, что почти все граф являются связными.

При знакомстве с основами теории графов после введения основных определений, изучается «джентельменский» набор свойств как-то связность, регулярность, полнота, диаметр. Обычно мы рассматриваем какие-то определенные «неслучайные» графы, которые могут лучшим образом проиллюстрировать данные свойства. Но какие свойства будут присущи случайно генерируемым графам? И что такое «случайный» граф?

Исходя из определения вероятности для конечного пространства, если рассматривать все графы фиксированного порядка n, то случайный граф – это любой из $2^{C_2^n}$ возможных графов, причем, вероятность любого такого графа $1/_{2^{C_2^n}}$. (Будем называть такой подход «равномерным».)

Однако есть еще одна модель случайного графа, принадлежащая П. Эрдёшу и А. Реньи. Эта модель несколько более сложная и менее интуитивная: помимо порядка графа в ней участвует вероятность p «существования» ребра между двумя любыми вершинами; в частности, если говорить про случайные графы, то нас интересует $p = 0{,}5$. (Подробнее – см. [3].)

Говорят, что *почти все* графы обладают некоторым свойством, если отношение числа графов порядка n, имеющих это свойство, к числу всех графов этого порядка стремится к 1 при $n \to \infty$, то есть, вероятность того, что случайный граф порядка n, обладает рассматриваемым свойством, стремится к 1 при росте n.

Оказывается, имеют место следующие совсем свойства почти всех графов [2]: *(а) почти все графы имеют диаметр* 2, *и (б) почти все графы связны.* (Легко видеть, что имеют место из первого свойства, следует второе.) Здесь уместно заметить, что с «человеческой» точки зрения второе свойство «очевидно», а первое – нет: если попросить группу студентов нарисовать граф, то почти наверняка будет нарисован связный граф, так как *психологически* именно ребра, соединяющие вершины, превращают множество вершин в единый объект «граф»; однако, зачастую люди рисуют не «граф общего вида», но дерево (диаметра больше 2).

Цель лабораторной работы по экспериментальной проверке свойства, что почти все графы являются связными: знакомство с математическим доказательством свойства, методами генерации случайных графов, пакетами визуализации (рисования) графов. Выполнение данной лабораторной начинается с разработки алгоритма для генерации случайного графа заданного порядка n. Можно выбрать для реализации оба варианта – «равномерный» и методом Эрдеша-Реньи.

Сначала обсудим равномерный метод генерации. Теоретически – все просто: строим все графы порядка n, а затем – выбираем «любой» из них (например, используя генератор псевдослучайных чисел). Однако, построить все графы даже сравнительно небольшого порядка (например, 10) – совсем не тривиальная задача из-за их количества (например, при $n = 10$ получается 55 возможных ребер, а значит $C_2^n = 2^{55}$ графов).

Поэтому в качестве возможного решения в лабораторной работе можно принять такой вариант: генерируется случайный битовый вектор длины C_2^n, представляющий собой характеристическую функцию ребер случайного графа порядка n. В качестве эксперимента в лабораторной работе таким образом было сгенерировано 100 графов порядка $n = 10$ и $\sim 90\%$ из сгенерированных графов оказались связными (что, на наш взгляд, хорошо подтверждает теоретический результат о том, что почти все графы являются связными).

Что касается генерации случайных графов в соответствии с моделью Эрдёша-Реньи, то задача более понятная: мы генерируем случайные биты и последовательно заполняем ими двоичный вектор длины C_2^n, представляющий собой характеристическую функцию ребер случайного графа порядка n. Результат анализа 100 графов, сгенерированных таким алгоритмом, несколько отличается от предыдущего, а именно ~95 % графов оказались связны (что, на наш взгляд, еще лучше подтверждает теоретический результат о том, что почти все графы являются связными).

Таким образом, при выполнении данной лабораторной работы студент может разобраться теоретически и экспериментально убедиться в корректности утверждения, что почти все графы связны, познакомиться с методами генерации случайных графов.

ЗАКЛЮЧЕНИЕ

В заключении стоит отметить, что представленный нами подход позволяет интегрировать математику и программирование, снижает общую нагрузку на студентов первых годов обучения при выполнении обязательных лабораторных работ, что особенно важно ввиду высокой плотности учебного процесса, когда в одно и то же время студенты осваивают сразу несколько базовых дисциплин. Кроме того, важно сразу продемонстрировать студентам, что знания, получаемые в ходе освоения различных предметов, не являются разобщенными, а тесно связаны, когда речь идет о разработке конкретного программного обеспечения для решения тех или иных прикладных задач. Однако важно подчеркнуть, что такой подход накладывает определенные квалификационные ограничения на преподавательский состав. Преподаватели математических дисциплин должны хорошо разбираться в основах программирования, в то время как преподаватели программирования должны уметь ответить на профильные вопросы, связанные с реализацией конкретных математических алгоритмов. В идеале на подобных курсах должен работать человек, который может синхронизировать подобный процесс для математических и компьютерных дисциплин.

СПИСОК ЛИТЕРАТУРЫ

1. *Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2001. – 368 с.

2. *Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И.* Лекции по теории графов. – М.: Наука, 1990. – 384 с.

3. *Райгородский А.М.* Модели случайных графов. – М.: МЦНМО, 2011. – 136 с.

COMPUTER PRACTICE IN UNDERGRADUATE MATHEMATICS FOR PROGRAMMERS

Shilov Nikolay[1], Konyukhov Ivan[2], Kholodov Yaroslav[3], Yagudin Zakhar[4], Masiagin Sergei[5]

Innopolis University, Innopolis

[1]n.shilov@innopolis.ru, [3]i.konyukhov@innopolis.ru, [2]ya.kholodov@innopolis.ru, [4]z.yagudin@innopolis.university, [5]s.masiagin@innopolis.ru

Abstract

We present an approach to programming lab assignments in undergraduate mathematical disciplines in Innopolis University for students with major in "Computer Science and Computer Engineering". First, the approach aims to unification of lab assignments in mathematical disciplines and programming ones to reduce the overall academic load during the first four semesters of studies. At the same time, the ap-

proach provides also optional labs for volunteers to popularize both mathematics and the art of programming.

Keywords: *real analysis, discrete mathematics, linear algebra, differential equations, introduction to programming, algorithms and data structures.*

REFERENCES

1. *Emelichev V.A., Mel'nikov O.I., Sarvanov V.I., Tyshkevich R.I.* Lekcii po teorii graphov. – M.: Nauka, 1990. – 384 s.

2. *Gamma E., Helm R., Johnson R.E., Vlissides J.* Design patterns: elements of reusable object-oriented softwareJanuary Boston, USA: Addison-Wesley Longman Publishing Co. – 1995. – 395 p.

3. *Raygorodsky A.M.* Modely sluchaynykh graphov. – M.: MCNMO, 2011. – 136 s.

СВЕДЕНИЯ ОБ АВТОРАХ

ШИЛОВ Николай Вячеславович – к.ф.-м.н., доцент, АНО ВО «Университет Иннополис», г. Иннополис.

Nikolay Vyacheslavovich SHILOV – PhD, Assistant Professor, Innopolis University, Innopolis.

email: n.shilov@innopolis.ru

КОНЮХОВ Иван Владимировичс – к.ф.-м.н., доцент, АНО ВО «Университет Иннополис», г. Иннополис.

Ivan Vladimirovich KONYUKHOV – PhD, Assistant Professor, Innopolis University, Innopolis.

email: i.konyukhov@innopolis.ru

ХОЛОДОВ Ярослав Александрович – к.ф.-м.н., профессор, АНО ВО «Университет Иннополис», г. Иннополис.

Yaroslav Alexandrovich KHOLODOV – PhD, Assosiate Professor, Innopolis University, Innopolis.

email: ya.kholodov@innopolis.ru

ЯГУДИН Захар Русланович – студент II-курса, АНО ВО «Университет Иннополис», г. Иннополис.

Zakhar Ruslanovich YAGUDIN – second year undergraduate, Innopolis University, Innopolis.

email: z.yagudin@innopolis.university

МАСЯГИН Сергей Владимирович – к.п.н., проректор, АНО ВО «Университет Иннополис», г. Иннополис.

Sergei Vladimirovich MASIAGIN – PhD, Vice-Rector, Innopolis University, Innopolis.

email: s.masiagin@innopolis.ru