Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пермская государственная сельскохозяйственная академия
имени академика Д.Н. Прянишникова»

В.В. Аюпов, А.В. Аюпов

Прикладная математика

Учебное пособие

Пермь
ИПЦ «Прокростъ»
2017

УДК 51
ББК 22.1
А 998

Рецензенты:

И.К. Березин, доктор технических наук, профессор, ведущий научный сотрудник ИМСС УрО РАН;

С.В. Лутманов, кандидат физико-математических наук, доцент кафедры механики и математического моделирования Пермского государственного национального исследовательского университета.

А 998 Аюпов, В.В.

Прикладная математика : учебное пособие / В.В. Аюпов, А.В. Аюпов; М-во с.-х. РФ; федеральное гос. бюджетное образов. учреждение высшего образов. «Пермская государственная с.-х. акад. имени акад. Д. Н. Прянишникова». – Пермь : ИПЦ «Прокростъ», 2017.– 147 с.

ISBN 978-5-94279-359-3

В учебном пособии обсуждаются предмет и содержание прикладной математики, ее основные понятия, определения и положения. Рассмотрены методологические вопросы теории моделирования, приведена классификация математических моделей, даны описания основных форм математических моделей, используемых при решении прикладных задач, а также вопросы компьютерного моделирования с применением универсальных систем компьютерной математики.

Учебное пособие предназначено для аудиторной (главы 1-3) и самостоятельной работы (главы 4-7) магистрантов, обучающихся по направлению 21.04.02 Землеустройство и кадастры Пермской ГСХА, для изучения дисциплины «Прикладная математика». Пособие может оказаться полезным студентам, магистрантам и аспирантам других направлений и специальностей сельскохозяйственных вузов при выполнении ими выпускных квалификационных работ.

УДК 51
ББК 22.1

Печатается по решению методического совета Пермской государственной сельскохозяйственной академии (протокол № 1 от 11.09. 2017 г.).

ISBN 978-5-94279-359-3

Оглавление

Введение

Среди различных характеристик современного периода можно выделить одну особую – это всеобщая *математизация*. На сегодняшний день математические расчеты прочно проникли в самые разнообразные отрасли знаний и научные дисциплины. Сейчас невозможно себе представить создание образцов новой техники, строительство зданий и сооружений, экономику, управление и другие сферы человеческой деятельности без применения математических моделей и методов их расчета.

В настоящее время применяемый *математический аппарат* стал значительно разнообразнее и сложнее, чем это было еще совсем недавно.

Вследствие этого повысились требования к математическому *образованию* специалистов в различных сферах деятельности, к выпускникам вузов инженерных специальностей и других направлений.

Можно отметить и другой существенный фактор, способствующий значительному повышению интереса к методам *математического моделирования* как в науке и технике, так в других областях – это развитие и широкое распространение средств вычислительной техники. С помощью моделей, реализованных на компьютере, можно изучать новые явления, решать практически все задачи анализа и проектирования сложных систем, осуществлять выбор наилучших вариантов решений, выполнять анализ и прогнозирование поведения систем и решать множество других задач.

Прикладная математика – это один из основных разделов математики, который включает создание, обоснование алгоритмов и их применение при решении задач в различных областях науки, техники и социально-экономической практике, в том числе в области землеустройства и кадастров.

Цель данного пособия – формирование системы компетенций для решения профессиональных задач студентами магистратуры направления 21.04.02 Землеустройство и кадастры с применением методов прикладной математики и средств компьютерного моделирования.

Целью изучения дисциплины «Прикладная математика» является изучение методов построения и анализа математических моделей, формирование у студентов магистратуры математической культуры, необходимой для успешного решения в будущем профессиональных и общественных задач, общих знаний и умений в области математического моделирования систем и мотивации к самообразованию.

Содержание учебного пособия построено на материалах различных литературных источников и авторских разработках по прикладной математике.

В результате изучения дисциплины студент магистратуры должен: выработать способность к использованию основных законов естествознания в сфере своей деятельности, готовность к участию в решении профессиональных задач, знать учебный материал, решать задачи на основе стандартных алгоритмов решения, овладеть навыками компьютерного моделирования и решения усложненных математических задач.

Для освоения материала данного учебного пособия студентам магистратуры достаточно знаний, полученных при изучении курсов высшей математики, информатики и курсов естественно-научных дисциплин.

1. Предмет и содержание прикладной математики

1.1. Определения математики

В настоящее время нет какого-то одного общего определения математики. Существует множество определений, отражающих различные аспекты математики, сложность ее содержания и предмета. Вот некоторые определения, данные классиками науки.

Одно из первых определений предмета математики дал Р. Декарт [Декарт, 1989]:

К области математики относятся только те науки, в которых рассматривается либо порядок, либо мера, и совершенно не существенно, будут ли это числа, фигуры, звёзды, звуки или что-нибудь другое, в чём отыскивается эта мера. Таким образом, должна существовать некая общая наука, объясняющая всё относящееся к порядку и мере, не входя в исследование никаких частных предметов, и эта наука должна называться не иностранным, но старым, уже вошедшим в употребление именем Всеобщей математики.

Определение Ф. Энгельса [Маркс, Энгельс, 1961]:

Математика... наука о количественных отношениях и пространственных формах действительного мира.

Формулировка Н. Бурбаки [Бурбаки, 1963]:

Математика – это учение об отношениях между объектами, которые в качестве аксиом положены в основание теории. Математика есть набор абстрактных форм – математических структур.

В свою очередь, Лобачевский Н.И. [Лобачевский, 1956]: определил математику как *язык науки*.

Но мало лишь правильно изъясняться на этом языке, нужно с помощью него уметь строить абстрактные конструкции (модели), работающие как в самой математике, так и в других науках.

Как обоснование того, что невозможно дать окончательное определение математики на все времена, высказывается довод, что любое определение математики заключает ее в какие-то границы, а математика может обобщить и изменить любую схему, поэтому такое определение обречено быть некорректным.

По мнению А.Н. Колмогорова [Колмогоров, 1988], определение Энгельса нуждается лишь в такой модернизации:

Математика – это наука о количественных отношениях и пространственно-подобных формах во всей их общности.

Во многих из определений математики речь идет только о *теоретической* «чистой» математике, хотя математика включает как чистую и прикладную части, так и метаматематику (совокупность формализуемых представлений о математике).

Прикладную математику, согласно одному из подходов, можно определить как совокупность теорий о системах (моделей) объективной действительности и мышления, полученных интерпретацией теорий чистой математики.

Метаматематика – теория, изучающая синтаксические (формальные), семантические (содержательные) и логические свойства теорий чистой математики, то есть метаматематика занимается непротиворечивостью математических теорий и математики в целом, их полнотой, независимостью систем аксиом.

1.2. Различие и единство теоретической и прикладной математики

Различие в *целях* и *направлении исследований* в *теоретической* и *прикладной* математике проявляется в специфических методах рассуждений, используемых в той и другой. Известно, что возникновение теоретической математики было связано с применением *дедуктивных методов* рассуждений. Вся догреческая математика (т. е. математика древнего Шумера, Вавилона и Египта) представляла собой совокупность различных приемов для решения сугубо практических задач по исчислению времени, измерению площадей и объемов, определению координат небесных тел для ориентирования на суше и на море и т. д. Решение этих задач опиралось на эмпирические наблюдения и *индуктивные обобщения*, результаты которых имеют не достоверный, а вероятностный характер, так как дальнейшие наблюдения могут привести к их исправлению, уточнению или даже опровержению. Вот почему древние греки, заимствовавшие многие математические сведения от египтян и частично – от вавилонян, по свидетельству Платона, значительно исправили и усовершенствовали их. Древнегреческие ученые стали обосновывать математическое знание не с помощью эмпирического опыта и индуктивного рассуждения, а посредством *логического доказательства*, т. е. *дедукции*, или вывода большинства математических утверждений из небольшого числа исходных посылок, которые считались очевидными. Начиная с древних греков, именно дедуктивный способ доказательства истин и абстрактный характер объектов исследования считаются отличительными особенностями *теоретической математики*.

В противоположность дедуктивному способу доказательства в *прикладной математике* допускается использование менее строгих способов рассуждения и образования понятий. Так в принципе происходит разделение математики на теоретические и прикладные отрасли исследования. Впервые такое разделение, перешедшее впоследствии в прямое противопоставление теоретической математики прикладной, возникло в античной Греции.

В XVI веке бурный рост производства и техники, мореплавания и торговли выдвинул перед естествознанием и математикой новые проблемы, прежнее противопоставление теоретической математики математике прикладной сменилось их гармоническим развитием в рамках *единого математического мышления*. Само математическое познание в период возрождения наук и искусств развивалось в тесном контакте с естествознанием, в особенности с механикой, астрономией, гидравликой, оптикой. Примечательно, что именно в это время математический метод стал систематически применяться в экспериментальном исследовании. Достаточно упомянуть в этой связи блестящие исследования Г. Галилея по экспериментальному изучению законов механического движения, в которых он стал впервые использовать математику для установления количественных зависимостей между величинами, характеризующими процессы механического перемещения тел. Область умозрения была ограничена выдвижением только таких гипотез, которые допускали эмпирическую проверку своих следствий. Для выведения соответствующих следствий наряду с правилами логики использовались и математические методы. Тесное взаимодействие чистой математики с прикладной, а также всей математики с естествознанием привело к решающему повороту в развитии математического познания. Этот поворот был вызван, прежде всего, необходимостью количественного исследования процессов, которые присущи разнообразным формам движения материи, и, в первую очередь, такой простейшей его форме, как механическое перемещение земных и небесных тел. Поворотным пунктом в математике стала декартова *переменная величина.* Благодаря этому в математику вошли такие понятия, как *изменение* и *движение,* вследствие чего появилась *необходимость в дифференциальном и интегральном исчислении*. Творцы анализа бесконечно малых И. Ньютон и Г. В. Лейбниц, как и большинство выдающихся математиков XVII–XVIII вв., были не только математиками, но и естествоиспытателями и поэтому охотно брались за решение прикладных задач.

Ньютон, как известно, пришел к созданию дифференциального и интегрального исчислений в связи с решением проблем земной и небесной механики. В дальнейшем плодотворное сочетание теоретических и прикладных исследований постоянно оказывало стимулирующее воздействие на развитие математического познания и, тем самым, расширяло рамки применения математических методов в других науках. Однако, начиная с середины XIX в., лавинообразный характер процесса накопления информации в науке, углубляющаяся дифференциация отдельных ее отраслей все настойчивее диктовали необходимость профессионального разделения труда между учеными, работающими в абстрактных разделах математики, и теми, кто занимался приложением ее методов к конкретным наукам и практике.

Положение еще более усложнилось после появления быстродействующих электронно-вычислительных машин (ЭВМ) и позднее – компьютеров. В первое время такие машины воспринимались как большие арифмометры, но вскоре стало очевидным, что они могут использоваться не только для расчетов, но и как эффективное средство научного исследования. Это, как и возникновение целого ряда новых прикладных отраслей математики, не говоря уже о кибернетике, во многом изменило характер и возможности прикладной математики. Именно в наше время дискуссия о соотношении теоретической и прикладной математики, их целей и методов исследования приобретает особую актуальность. Нередко можно слышать утверждение, что математика как наука едина, поэтому есть одна математика, являющаяся одновременно и теоретической и прикладной. В целом против этого возражать трудно, если иметь в виду общий предмет и задачи математического познания. Однако тезис о единстве математики иногда истолковывают неправильно. Вместо того чтобы видеть связь и различие между теоретической и прикладной математикой, их начинают противопоставлять друг другу, преувеличивать роль одной из них за счет другой. Это особенно характерно для тех ученых, которые считают математику чисто абстрактной конструкцией мысли, лишенной каких бы то ни было связей с реальной действительностью, естествознанием, техникой, экономикой и практикой в широком смысле слова. Таким образом, верная сама по себе идея о единстве математики приводит к отрицанию этого единства, если отсутствует диалектическое понимание различий и противоречий внутри этого единства.

Предмет исследований теоретической и прикладной математики в целом один и тот же, но *цели, средства и методы* познания во многом отличаются друг от друга. Именно это различие *содержания* теоретической и прикладной математики в рамках единого предмета исследования создает наиболее благоприятные условия для развития математики в том, и только в том случае, когда теоретические исследования способствуют приложению ее идей и теорий на практике и, наоборот, когда проблемы, возникающие в прикладных отраслях, стимулируют разработку новых методов и теорий чистой математики. При такой гармонии, непременным условием которой является подлинное понимание не только различий, но и глубокого единства между теоретическими и прикладными исследованиями в математике, преодолевается и другая ошибочная точка зрения на соотношение чистой и прикладной сторон математического познания. Эта точка зрения является большей частью реакцией на недооценку и игнорирование прикладной математики как специфической отрасли исследования. Ее сторонники, конечно, не отвергают теоретическую математику, но считают ее чрезмерно высокие абстракции и жесткие стандарты дедуктивных рассуждений непригодными для решения прикладных проблем. Отсюда и возникает весьма скептическое отношение к логической строгости чистой математики, которая выражается в доказательстве многочисленных теорем существования без какой-либо попытки найти (вычислить или построить) искомый математический объект (решение задачи, корень уравнения, конструктивное доказательство теоремы). Такой подход, подчеркивающий приоритет прикладных исследований над теоретическими, редко, конечно, высказывается открыто самими математиками, но его нередко защищают потребители математики, число которых с расширением математизации научно-технического знания неуклонно растет. Для людей, которые пользуются готовыми выводами, формулами и уравнениями математики, наиболее ценными представляются именно конечные продукты математического исследования, допускающие применение в их собственной области и на практике. В силу профессиональной ограниченности они забывают о том, что это было бы невозможно без предварительной теоретической работы в области чистой математики. Таким образом, нет непреодолимой грани между прикладной и теоретической математикой.

Вполне сознавая относительность такого разграничения, попытаемся все же подробнее охарактеризовать цели, средства и методы исследования, т.е. содержания прикладной математики. В принципе математические методы применимы там, где научное исследование явлений достигло такой степени зрелости, что необходимо требует использования математических понятий и теорий. В силу многообразия областей применения идей и методов прикладную математику иногда отождествляют, с одной стороны, со всеми теми отраслями знания, где применяются эти методы, с решением математических проблем, возникших вне самой математики. При таком расширительном понимании прикладной математики игнорируется качественное различие между математическим и конкретно-научным исследованиями, что не способствует установлению правильного взаимоотношения между ними. В действительности возможность применения математики в той или иной науке не превращает эту науку в отрасль математики, так как математические методы играют в ней подчиненную роль и служат для выражения количественных отношений. С другой стороны, нередко прикладную математику понимают слишком узко, отождествляя ее с вычислительной математикой. Выражение результатов исследований в числах, использование различных вычислительных методов и техники, несомненно, имеют чрезвычайно важное значение для количественного анализа явлений. И все же прикладная математика не сводится к вычислительной математике хотя бы потому, что для применения математики к решению проблем, возникших вне ее рамок, необходимы многие другие, не вычислительные методы. Прежде чем использовать вычислительные методы и технику, нужно описать поставленную задачу на математическом языке, т. е. выразить зависимости между величинами, характеризующими исследуемый процесс, с помощью уравнений и их систем, а нередко и других математических структур. Такое описание принято называть *математической моделью процесса*, о чем мы подробно будем говорить во второй главе. Здесь же отметим, что построение и анализ математической модели требуют привлечения разнообразных математических методов. Как видно, современная прикладная математика охватывает широкий круг вопросов, связанных с использованием математических методов и вычислительной техники. Однако применение математических методов для количественного описания и анализа исследуемых процессов имеет гораздо большее значение, чем использование вычислительных методов и техники. Именно применение математических методов для построения и анализа математической модели реальных процессов в первую очередь определяет возможность их математизации.

В последние годы в связи с бурным развитием вычислительной техники, – указывает академик Ю. А. Митропольский [Митропольский, 1964], – этот аспект оказывается второстепенным и вторичным при попытке объяснить причины математизации современного мира. И с этим нельзя не согласиться.

И. И. Блехман, А. Д. Мышкис и Я. Г. Пановко [Блехман, Мышкис, Пановко, 1976], опубликовавшие специальную работу, посвященную анализу предмета, логики и особенностям подходов прикладной математики, также считают отождествление прикладной математики с вычислительной математикой – точкой зрения узкой и создающей одностороннюю ориентацию. Поэтому они определяют прикладную математику как науку *об оптимальных,* грубо говоря, практически приемлемых *методах решения математических задач, возникающих вне математики.* Таким образом, прикладная математика оказывается для них математикой, опосредствованной практикой.

В этом определении обращается внимание на то, что задачи, которые решает прикладная математика, возникают в других науках под воздействием практических потребностей. Такое опосредованное влияние практики стимулирует и облегчает разработку математических методов, так как проблемы, возникающие в рамках конкретных наук, уже значительно упрощены и схематизированы: отделены второстепенные факторы и выделены существенные зависимости между величинами. Прикладная задача должна быть решена не только правильно, но и своевременно, экономно, с точки зрения затраченных усилий, точность ее решения должна соответствовать поставленной цели и т. д. Подобные требования к задаче авторы называют оптимальным решением. Это идеал, к которому следует стремиться. Но что лежит в основе такого идеала? Какие предпосылки необходимы для получения оптимального решения задачи в прикладной математике? Прежде всего, необходимо, чтобы математическая модель была адекватна поставленной задаче, т. е. правильно описывала на математическом языке уравнений и других абстрактных структур существенные связи исследуемых процессов. Располагая такой моделью, можно изучать ее особенности при оптимальных условиях. Поскольку математическая модель служит основой для применения математики к решению конкретных задач, возникающих в естествознании, технике и экономике, постольку прикладную, математику можно определить как науку об общих принципах построения, анализа и проверки математических моделей. Только благодаря замене реального объекта соответствующей ему моделью, – указывают А. Н. Тихонов и Д. П. Костомаров [Тихонов, Костомаров, 1979], – появляется возможность сформулировать задачу его изучения как математическую и воспользоваться для анализа универсальным математическим аппаратом, который не зависит от конкретной природы объекта.

1.3. Методологические подходы к изучению прикладной математики

Современный инженер не может работать без знания основ математики и информатики. Поэтому современная, работающая математика соответствует современной технике, средствам наблюдения и эксперимента. То есть современная математика является в значительной мере прикладной.

Но для того чтобы эффективно применять математику на практике и правильно использовать ее методы, необходимо знать теоретические основы математики, используемый в исследованиях математический аппарат, уметь разрабатывать математические модели, исследовать их, делать выводы и правильно формулировать практические рекомендации. То есть нужно изучать теоретическую математику.

Выделим пять методологических подходов (*методов), в соответствии с которыми следует изучать математику: исторический, познавательный, прикладной, предметный и топологический*.

В историческом плане математику разделим на *элементарную, высшую и современную.*

Элементарная математика пользуется теми понятиями (абстракциями), которые сложились исторически до появления высшей математики. Элементарная математика охватывает в основном арифметику (элементарную теорию чисел), элементарную алгебру, элементарную геометрию, тригонометрию.

Под высшей математикой понимают совокупность математических дисциплин, возникших примерно в середине шестнадцатого века и входящих в учебные планы дисциплин высших учебных заведений. Как правило, это линейная алгебра, аналитическая геометрия, математический анализ и теория дифференциальных уравнений.

Современную математику условно определим как математический комплекс, включающий в себя все разделы элементарной, высшей математики, а также математические дисциплины и теории, разработанные с пятидесятых годов прошлого века до наших дней.

Познавательный подход представим следующими направлениями: *прагматизм, интуиционизм, конструкционизм, формализм, классицизм* [Аюпов, 2010].

Основной идеей прагматизма в математике является то, что математика должна быть лишь инструментом при решении проблем и задач, которые возникают в различных жизненных ситуациях, в процессе практической деятельности человека в постоянно меняющемся мире.

Интуиционизм предполагает в основании математики интуиционистскую логику, более ограниченную в средствах доказательства. Интуиционизм отвергает доказательство от противного, многие неконструктивные доказательства становятся невозможными, а многие проблемы теории множеств – неформализуемыми.

Конструктивная математика (конструкционизм) – близкое к интуиционизму течение в математике, изучающее конструктивные построения. Согласно критерию конструктивности "существовать – значит быть построенным". Критерий конструктивности – более сильное требование, чем критерий непротиворечивости.

Формализм – направление математики, в котором проблемы основания математики решаются при помощи формально-аксиоматических построений. Основоположником этого течения, возникшего в начале двадцатого века, является немецкий математик Давид Гильберт. Согласно формализму математика характеризуется скорее своим методом, нежели предметом изучения.

Классицизм означает использование аппарата классической – (высшей) – математики (математики конца XIX – начала XX в.). Высшая математика – курс обучения в высших учебных заведениях, включающий элементы высшей и линейной алгебры, аналитическую геометрию, дифференциальное и интегральное исчисления, дифференциальные уравнения, теорию множеств, теорию вероятностей и элементы математической статистики.

Прикладной подход представим двумя направлениями – теоретическая (чистая) и прикладная математика.

К теоретической математике относят арифметику, алгебру, функциональный анализ, анализ бесконечно малых величин, а также дифференциальное исчисление, интегральное исчисление и вариационное исчисление, теорию чисел, геометрию, тригонометрию и др.

Прикладная математика – это область математики, в которой рассматриваются приложения математического знания в других сферах деятельности. Примерами такого применения являются численные методы, оптимизация и исследование операций, моделирование сплошных сред, теория информации, теория игр, теория вероятностей и математическая статистика, финансовая математика, комбинаторика, теория графов и др.

Предметный подход. Предмет математики – это то, что изучает математика как наука. Одно из возможных определений предмета математики – изучение систем математических объектов, таких как точка, линия, поверхность, число, вектор, матрица, функция, оператор и др. В зависимости от рассматриваемых объектов и применяемых методов в предметном подходе можно выделить следующие направления математики: классическая, конечная, вычислительная, компьютерная и асимптотическая математика.

Топологический подход представлен дискретной и континуальной математикой.

Основные определения и понятия двух последних подходов и их разделов можно найти в приведенных литературных источниках, посвященных философским и методологическим вопросам математики [Математический энциклопедический словарь, 1988, Яблонский, 2003, Ерусалимский, 2009 и др.].

1.4. Предмет прикладной математики

Предмет математики – то, что изучает математика как наука, выраженное в наиболее общей форме. Одно из возможных определений предмета математики – это изучение систем математических объектов. Проблема определения предмета математики тесно связана с проблемой определения самой математики и ее сути, что она из себя представляет.

Существуют различные подходы к определению предмета математики. В частности, в литературе высказывается мнение, что предмет математики менялся на протяжении ее развития. При этом достаточно актуальными остаются мысли, высказанные древнегреческими философами.

Математические объекты. Математика непосредственно изучает системы математических объектов. Эти объекты определяются в рамках самой математики (базовые объекты – через системы аксиом). Проблема их связи с объективной реальностью («априорность математики») не имеет однозначного разрешения и выходит за рамки математики (изучается в философии математики). Математические объекты возникают как результат человеческого мышления и не существуют в объективной реальности. Идеализированные объекты появляются и в других науках, но в них они сохраняют большее сходство с реальностью, в математике их подобие объективной реальности минимальное. Математика изучает системы отношений между математическими объектами, которые также не существуют в материальном мире. Тем не менее, ряд математических теорий находит приложения в качестве основы для моделей процессов реального мира, что является основой развития ряда современных наук и математизации научного знания.

Проблема обоснования математики. С появлением все новых и более абстрактных математических теорий с XIX века появился вопрос об их *обосновании*. Очевидно, что проверить опытным путем такие теории нельзя. Поэтому обоснование математических теорий стало пониматься как получение доказательства их *непротиворечивости* и *полноты*. От работ Георга Кантора идет перевод оснований математики на язык теории множеств. Однако теория множеств столкнулась с некоторыми логическими парадоксами (так называемый парадокс брадобрея, парадоксы, связанные с бесконечностью, парадокс множества всех множеств и др.), что привело к необходимости пересмотра логических оснований математики. С состоянием дел на сегодняшний день по данному вопросу можно более подробно ознакомиться в учебном пособии Светлова В.А. [Светлов, 2016] и в других источниках [Математический энциклопедический словарь, 1988, Рузавин, 1983 и др.].

1.5. Цели и задачи изучения дисциплины «Прикладная математика»

Целью изучения дисциплины «Прикладная математика» является формирование у обучающихся математической культуры, необходимой для успешного решения в будущем профессиональных и общественных задач, общих знаний и умений в области прикладной математики, математического моделирования и мотивации к самообразованию.

Задачами изучения дисциплины являются:

– формирование умений решать оптимизационные задачи и применять их на практике;

– приобретение навыков поиска и оценки источников информации, анализа данных, необходимых для проведения различных расчетов;

– освоение методов решения математических прикладных задач с применением компьютерной системы «Mathematica» в сфере своей профессиональной деятельности.

1.6. Содержание дисциплины «Прикладная математика»

Содержанием науки является деятельность по получению новых знаний и ее результат.

Содержанием математики является система математических объектов и действий с ними, математических моделей и инструментов для их создания.

Содержание прикладной математики определяют как систему прикладных математических моделей и инструментов для их создания и применения.

В данном учебном пособии в качества содержания изучаемой дисциплины «Прикладная математика» рассматриваются следующие тематические разделы курса:

- введение в предмет дисциплины «Прикладная математика»;
- постановка и решение задач оптимизации;
- элементы линейного программирования;
- введение в компьютерную систему «Mathematica»;
- решение задач математического анализа в компьютерной системе «Mathematica»;
- построение графиков функций и работа с графическими объектами в системе «Mathematica»;
- решение дифференциальных уравнений с применением компьютерной системы «Mathematica»;
- решение оптимизационных задач с применением компьютерной системы «Mathematica»;
- решение задач линейного программирования в компьютерной системе «Mathematica».

2. Математическое моделирование

Определение прикладной математики как *науки о математических моделях* дает возможность понять, во-первых, взаимосвязь между нею и другими науками и, в конечном счете, со всей общественной практикой. Во-вторых, при таком определении становится более ясным ее отношение к чистой, или теоретической, математике. В-третьих, такое понимание полнее раскрывает специфику работы математика-прикладника, которому для оптимального решения задач и построения математической модели необходимо либо самому глубоко изучить конкретные факты, либо работать в тесном контакте со специалистами определенной отрасли знания. Наконец, в-четвертых, подобный подход к определению прикладной математики позволяет выявить особенности применяемых в ней методов исследования и способов рассуждения.

2.1. Понятие модели

Существует достаточно большое число определений понятия «модель». Одни из них слишком абстрактны, другие – слишком конкретны. Ключевым для определения модели является представление объекта или процесса в упрощенном виде.

Модель – это форма представления и существования наших знаний.

Модель – это инструмент познания окружающего мира.

Модель – как аналог (образец) будущего изделия.

Модель – как аналог реального объекта.

Аналогия (от греч. *analogia* – соответствие, соразмерность) – это представление о каком-либо частном сходстве двух объектов, причем сходство может быть как существенным, так и несущественным. Существенность сходства или различия двух объектов условна и зависит от уровня абстрагирования (отвлечения), определяемого конечной целью исследования. Уровень абстрагирования зависит от набора учитываемых параметров объекта исследования.

В дальнейшем будем придерживаться следующего определения понятия модели, которое является более узким и более конкретным.

Объект **M** *является в определенных условиях* ***моделью системы*** **S** *(объекта, процесса, явления, ситуации), если* ***модель*** **M** *имитирует (воспроизводит) требуемые характеристики (свойства, признаки) системы* **S**.

Таким образом, модель и исходная система *эквивалентны* относительно множества *воспроизводимых* характеристик, в то время как полное множество характеристик самой системы, как правило, значительно шире подмножества характеристик, воспроизводимых моделью. Например, при моделировании броска камня под некоторым углом к горизонту, в простейшей модели принимается, что воздушная среда отсутствует (сопротивление воздуха и воздействие ветра отсутствуют).

Модель **M** по сравнению с оригиналом **S** имеет существенные преимущества: наглядность, простота, обозримость, легкость преобразований с ней, возможность проведения испытаний и получения с ее помощью новых информации и знаний. В той же модели броска камня в первом приближении получаем траекторию полета камня в виде параболы, что вполне согласуется с многочисленными опытными данными.

В свою очередь сама модель является системой. Модель имеет структуру, цель, является некоторой иерархически организованной целостностью.

Структура модели – это упорядоченное множество элементов и их отношений.

Понятие модели так же, как и понятие системы, претерпевало определенную эволюцию. Эволюция понятий моделей отражает эволюцию процесса познания. Так, на ранних этапах под моделью понимали некоторое физическое устройство (объект), которое в определенных условиях заменяет другой объект. Примерами таких устройств могут служить модели самолетов, кораблей, машин, различные макеты, шаблоны, протезы и т.д.

На следующем этапе под моделью объекта понимался объект-заменитель, который отражал лишь интересующие исследователя свойства и характеристики объекта-оригинала. При этом модель перед объектом обладала такими преимуществами, как наглядность, простота, доступность для эксперимента, возможность идентификации и т.д. Само понятие модели уже значительно расширилось и включало в себя чертежи, таблицы, характеристики, графики, рисунки, картографические изображения, различные формы описания устройств и т.д.

На третьем этапе в понятие модели включают модели реальных (физических, материальных) систем и абстрактные (идеальные) построения. Примером первых могут служить математические модели функционирования технических систем [Аюпов, 2017], последних – идеи, гипотезы, теории, математические и логические конструкции. Сам процесс мышления можно трактовать как процесс последовательного перехода от одних абстрактных моделей к другим. При этом модель выступает как форма существования и представления знаний об исследуемом объекте (явлении, процессе, системе). Таким образом, познание материального мира идет через модели, а целенаправленная деятельность человека невозможна без моделирования.

Модель создается для исследования объекта без воздействия на него. При этом к моделям предъявляются также требования по глубине и по времени. То есть, модель должна обладать необходимой глубиной описания, достаточной для решения актуальных проблем объекта, и дополнительными ограничениями по времени, необходимом для принятия решения.

Модель всегда тесно связана с проблемой, т. к. решение проблемы всегда начинается с моделирования проблемной ситуации объекта, а затем уже переходят к моделированию стратегических альтернатив и моделированию последствий принимаемого решения, куда, естественно, включаются такие элементы, как цель развития объекта управления, состояние внешней среды, функционирование объекта и др.

Модель может выступать в качестве эталона, идеализирующего собой различные формы деятельности: управление, планирование, принятие решений, прогнозирование и т.д. Например, в адаптивных (самонастраивающихся) системах управления реализуется принцип управления по эталонной модели.

Главный *недостаток* метода моделирования заключается в том, что при некорректном моделировании можно получить результаты, не имеющие отношения к исследуемым свойствам системы или неправильно отражающие свойства реальной системы. В этом есть объективная причина: модель отражает (не всегда точно) только определенные, но не все, свойства реального объекта.

Модели *экономичны,* так как они экономят время, сокращают издержки и затраты материальных ресурсов в процессе исследования или проектирования технического объекта.

Модели *практичны*, они всегда строятся так, чтобы были проще и удобнее для исследований, чем исходные объекты. На моделях можно ставить такие эксперименты, проведение которых на реальных объектах либо слишком дорого, либо опасно для персонала и окружающей среды.

Некоторые явления можно изучать *только* на их моделях. Например, ядерные взрывы, траектории космических аппаратов, электрические разряды молнии, полет самолета при развитии критической ситуации на борту в результате отказов отдельных функциональных подсистем и т.п.

Модели воспроизводят лишь основные, наиболее *важные для данного исследования* свойства изучаемой системы. Отсюда же следует, что у изучаемой системы (объекта) могут быть несколько (много) моделей, каждая из которых воспроизводит (имитирует) определенный набор свойств и характеристик. Так, например, проектируя новое техническое устройство, можно построить и использовать модель, описывающую динамические (упрощенно, скоростные) свойства и характеристики. В то же время для определения прочностных характеристик, изгибно-крутильных свойств потребуется совершенно другая модель.

Модели позволяют выявить механизм формирования исследуемых свойств системы, научиться прогнозировать эти свойства и целенаправленно их изменять в желаемую сторону.

Исследования, проведенные с применением моделей, могут послужить основанием для заключения о несостоятельности некоторых гипотез или идей.

При моделировании систем могут возникнуть и побочные эффекты. Например, модель может воспроизводить такие признаки системы, которые адекватны реальным свойствам, но данная модель не была предназначена для этого. Этот эффект следует рассматривать как исключение, а не как закономерность, хотя в истории науки есть случаи, когда подобным образом делались открытия в области тонких физических явлений.

Моделирование – это высший способ познания.

2.2. Классификация математических моделей

Первоначально дадим несколько различных определений математических моделей.

Математическая модель – это объект, который имеет с оригиналом следующее однозначное соответствие:

1) структуры, т.е. состава элементов и связей между ними;

2) уравнений, описывающих свойства этих элементов и их связей.

Учитывая, что система есть совокупность взаимосвязанных элементов (объектов) в определенном смысле обособленная от окружающей среды и взаимодействующая с ней как целое, можно сформулировать определение математической модели системы.

Математическая модель системы – это множество математических моделей элементов, взаимосвязанных и взаимодействующих друг с другом и адекватно отражающих свойства системы.

Практически любая математическая модель позволяет по заданным исходным данным найти значения интересующих исследователя параметров моделируемого объекта или явления. Поэтому можно полагать, что суть любой подобной модели заключается в отображении некоторого заданного множества значений входных параметров на множество значений выходных параметров. Данное обстоятельство позволяет рассматривать математическую модель как некоторый *математический оператор* и сформулировать следующее определение.

Математическая модель – это любой оператор $\boldsymbol{A}$, позволяющий по соответствующим значениям входных параметров $\boldsymbol{X}$ установить выходные значения параметров $\boldsymbol{Y}$ объекта моделирования: $\boldsymbol{Y}=\boldsymbol{AX}$.

В зависимости от природы моделируемого объекта элементами множеств $\boldsymbol{X}$ и $\boldsymbol{Y}$ могут являться любые математические объекты (числа, векторы, тензоры, функции, множества и т.п.). В то же время понятие оператора $\boldsymbol{A}$ в приведенном определении может трактоваться достаточно широко. Это может быть как некоторая функция, связывающая входные и выходные значения, так и отображение, представляющее символическую запись системы алгебраических, дифференциальных, интегро-дифференциальных или интегральных уравнений. Наконец, это может быть некоторый алгоритм, совокупность правил или таблиц, обеспечивающих нахождение выходных параметров по заданным исходным значениям.

Определение математической модели через понятие оператора является более конструктивным, с точки зрения построения классификации (табл.1) таких моделей, поскольку включает в себя все многообразие имеющихся в настоящее время математических моделей.

Классификация математических моделей Таблица 1

Признак классификации	Вид
Сфера применения	−прикладные −учебные
Способ получения	−теоретические −экспериментальные
Форма представления	−инвариантные −аналитические −графические −функциональные −структурные −алгоритмические
Вид оператора	−алгебраические −функциональные −дифференциальные −интегральные
Свойства параметров оператора модели	−линейные −нелинейные −сосредоточенные −распределенные −стационарные −нестационарные
Фактор времени	−статические −динамические
Количество входов/выходов	−скалярные −матричные (многосвязные)
Количество переменных состояния	−одномерные −многомерные
Характер переменных	−непрерывные −дискретные −логические −детерминированные −стохастические (вероятностные)

Прикладные модели применяются для получения новых знаний, для обоснования условий принятия решений и решения других прикладных задач;

Учебные модели применяются для передачи знаний об изучаемом объекте (процессе);

Теоретические модели получают на основе описания физических процессов функционирования объекта.

Экспериментальные модели формируются на основе поведения объекта во внешней среде, рассматривая его как ”черный ящик”. Эксперименты при этом могут быть физические (на техническом объекте или на его физической модели) или вычислительные (теоретической математической модели).

Инвариантная форма – это запись соотношений в математической модели в общем виде с помощью традиционного математического языка безотносительно (не учитывая) к методу решения.

Аналитические модели – модели в форме аналитических функциональных зависимостей, когда представление преобразования входного сигнала в выходной осуществляется с помощью некоторой функциональной зависимости или логического условия.

Графические (схемные) модели представляются в виде графов, эквивалентных схем, диаграмм и т.п.

Функциональные модели описывают процессы функционирования технических объектов и имеют форму систем уравнений. По способам получения функциональные математические модели делятся на *теоретические* и *экспериментальные*.

Структурные – модели, отображающие только структуру исследуемого объекта и использующиеся при решении задач структурного синтеза. Параметрами структурных моделей являются признаки функциональных или конструктивных элементов, из которых состоит технический объект и по которым один вариант структуры объекта отличается от другого. Эти параметры называют *морфологическими переменными*. Структурные модели имеют форму таблиц, матриц и графов.

Сложные явления и системы описываются множествами уравнений и соотношений. Получение требуемого результата моделирования в виде конечной формулы или численного значения является весьма сложной, а часто неразрешимой задачей. В этих случаях успешным является использование *алгоритмических* моделей.

Алгоритмические – модели в форме алгоритма получения требуемых результатов, реализуемого на компьютере с использованием методов вычислительной математики. Такие модели могут учитывать практически любое число существенных факторов, а потому используются для моделирования наиболее сложных объектов и процессов и чаще всего с помощью мощных и быстродействующих компьютеров.

Алгебраические – модели в форме алгебраических уравнений.

Дифференциальные – модели в форме дифференциальных уравнений (обыкновенных дифференциальных уравнений, систем обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, систем дифференциальных уравнений в частных производных).

Интегральные – модели в форме интегральных уравнений и систем интегральных уравнений.

Математическая модель называется *линейной,* если оператор модели обеспечивает линейную зависимость выходных величин от значений входных величин (выполняется принцип суперпозиции).

Математическая модель называется *нелинейной,* если оператор модели не обеспечивает линейную зависимость выходных величин от значений входных величин (не выполняется принцип суперпозиции).

В моделях с *сосредоточенными* параметрами предполагается, что все свойства оператора модели сосредоточены в фиксированных точках. Такое предположение приводит к использованию моделей в форме алгебраических и/или обыкновенных дифференциальных уравнений.

В моделях с *распределенными* параметрами предполагается, что свойства оператора модели распределены в пространстве, что приводит к тому, что оператор модели имеет вид дифференциальных уравнений в частных производных.

Стационарная (статическая) – модель, отображающая взаимосвязь между входным и выходным воздействиями объекта в его установившемся состоянии без учета времени. Математическая модель стационарна и в том случае, когда параметры оператора модели неизменны во времени. Математически это обстоятельство выражается в том, что параметры (коэффициенты) модели явно не зависят от времени.

Математическая модель называется *нестационарной* (неустановившейся) в том случае, когда параметры оператора модели изменяются с течением времени.

Статические математические – модели, которые описывают установившиеся (равновесные) режимы работы системы. По своей форме статические модели – алгебраические уравнения или функциональные зависимости, не содержащие в качестве аргумента время.

Динамические математические модели – модели, которые описывают неустановившиеся (неравновесные, переходные) режимы работы системы. Чаще всего динамические математические модели представляются в дифференциальной форме.

Разделение математических моделей на одномерные и многомерные, на скалярные и матричные не имеет строгих установившихся правил. Но наиболее часто используемыми являются следующие представления.

Модель называется *скалярной,* если в качестве входной переменной величины (входного сигнала) выступает одна единственная переменная величина и выходная переменная величина (выходной сигнал) также представлена в единственном числе. Число внутренних переменных (переменных состояния) при этом может быть произвольным.

Модель называется *матричной* (многосвязной), если число входных переменных и/или число выходных переменных величин не равно единице. Опять же число внутренних переменных (переменных состояния) при этом может быть произвольным.

Модель называется *одномерной*, если количество внутренних переменных (переменных состояния), обеспечивающих полное однозначное описание каждого состояния объекта моделирования равно единице. *Одномерная* математическая модель содержит *одну выходную величину.* Входных величин может быть несколько.

Модель называется *многомерной*, если количество внутренних переменных (переменных состояния), обеспечивающих полное однозначное описание каждого состояния объекта моделирования больше единицы. *Многомерная* математическая модель содержит *несколько выходных величин.* Для многомерного объекта число уравнений соответствует числу выходных величин.

Математические модели называются *непрерывными*, если все внутренние переменные модели являются непрерывными величинами.

Математические модели называются *дискретными*, если хотя бы одна переменная модели является дискретной величиной.

Логические – модели, в которых в качестве переменных величин используются логические величины или логические выражения.

Детерминированные – модели, переменные которых представляют собой детерминированные величины, а каждому параметру модели соответствует конкретное целое, вещественное или комплексное число либо соответствующая функция.

Стохастические (вероятностные) – модели, переменные которых представляют собой случайные величины, заданные плотностями вероятностей.

Классификация моделей по какому-либо одному признаку не может охватить всех видов моделей, ибо модель, как и исходная система, многогранна и отражает лишь те ее свойства, которые представляют интерес для исследователя.

В качестве примеров классификации математических моделей с помощью таблицы 1 будет рассмотрена классификация моделей систем и процессов, приведенных в разделах 4,5.

2.3. Свойства математических моделей

Рассмотрим некоторые свойства математических моделей, которые позволяют в той или иной степени либо различать, либо отождествлять модель с оригиналом (объектом, процессом). Принято выделять следующие свойства математических моделей: целенаправленность, адекватность, замкнутость, корректность, простота и сложность, мягкость и жесткость, конечность, приближенность, экономичность, истинность, информативность, полнота, адаптивность, управляемость, эволюционируемость.

Целенаправленность. Модель всегда отражает некоторую систему, то есть имеет *цель*. Остальные свойства модели зависят от постановки задачи.

Адекватность. Под *адекватностью* модели принято понимать правильное качественное и количественное описание объекта (процесса) по выбранному множеству характеристик с некоторой разумной степенью точности.

Адекватность является важнейшим требованием к модели, она требует соответствия модели ее реальному объекту (процессу, системе и т.д.) относительно выбранного множества его свойств и характеристик. При этом имеется в виду адекватность не вообще, а адекватность по тем свойствам модели, которые являются для исследователя существенными. Полная адекватность означает тождество между моделью и прототипом.

Математическая модель может быть адекватна относительно одного класса ситуаций (состояние системы + состояние внешней среды) и неадекватна относительно другого. Применение неадекватной модели может привести либо к существенному искажению реального процесса или свойств (характеристик) изучаемого объекта, либо к изучению несуществующих явлений, свойств и характеристик.

Адекватность модели должна проверяться, контролироваться, уточняться постоянно в процессе исследования на частных примерах, аналогиях, экспериментах и т.д. В результате проверки адекватности выясняют, к чему приводят сделанные допущения: то ли к допустимой потере точности, то ли к потере качества. При проверке адекватности также можно обосновать законность применения принятых рабочих гипотез при решении рассматриваемой задачи или проблемы.

Замкнутость. Математическая модель является *замкнутой*, если она учитывает и отображает замкнутую (полную) систему необходимых гипотез, связей и отношений.

Контроль математической замкнутости, состоящий в проверке того, что выписанная система математических соотношений дает возможность, притом однозначно, решить поставленную математическую задачу. Например, если задача свелась к отысканию $\boldsymbol{n}$ неизвестных из некоторой системы алгебраических или трансцендентных уравнений, то контроль замкнутости состоит в проверке того факта, что число независимых уравнений должно быть $\boldsymbol{n}$. Если их меньше $\boldsymbol{n}$, то надо установить недостающие уравнения, а если их больше $\boldsymbol{n}$, то либо уравнения зависимы, либо при их составлении допущена ошибка. Однако если уравнения получаются из эксперимента или в результате наблюдений, то возможна постановка задачи, при которой число уравнений превышает $\boldsymbol{n}$, но сами уравнения удовлетворяются лишь приближенно, а решение ищется, например, по методу наименьших квадратов. Неравенств среди условий также может быть любое число, как это бывает, например, в задачах линейного программирования. Свойство математической замкнутости системы математических соотношений тесно связано с введенным Ж. Адамаром [Адамар, 2002] понятием корректно поставленной математической задачи. При этом проверка математической замкнутости математической модели является менее сложной по сравнению с проверкой корректности математической модели.

Корректность. В практике математического моделирования используют такие понятия как «корректно поставленная задача» и «корректная математическая модель».

Математическая задача является *корректно поставленной*, если ее решение существует, оно единственно и непрерывно зависит от исходных данных. В этом случае решение считается непрерывным, если малому изменению исходных данных соответствует достаточно малое изменение решения. Доказательство корректности конкретной задачи часто является достаточно сложной математической проблемой. Математическая модель считается корректной, если для нее осуществлен и получен положительный результат всех вышеперечисленных контрольных проверок.

Понятие корректности задачи имеет большое значение в прикладной математике. Например, численные методы решения оправдано применять лишь к задачам, поставленным корректно. При этом далеко не все задачи, возникающие на практике, можно считать корректными (например, так называемые обратные задачи). Доказательство корректности конкретной математической задачи – достаточно сложная проблема, она решена только для некоторого класса математически поставленных задач. В настоящее время активно исследуются свойства некорректных задач, разрабатываются методы их решения.

Чтобы математическая модель была корректной, она должна пройти ряд проверок.

Математическая модель является *корректной*, если для нее осуществлен и получен положительный результат всех контрольных проверок: размерности, порядков, характера зависимостей, экстремальных ситуаций, начальных и граничных условий, физического смысла и математической замкнутости.

Проверка корректности математической модели

Во многих случаях оператор модели включает в себя систему обыкновенных дифференциальных уравнений (ОДУ), дифференциальных уравнений в частных производных (ДУЧП) и/или интегро-дифференциальных уравнений (ИДУ). Тогда для обеспечения корректности математической модели, представленной системой ОДУ (ДУЧП) добавляются начальные или граничные условия, которые могут быть алгебраическими или дифференциальными соотношениями различного порядка.

Можно выделить несколько наиболее распространенных типов задач для систем ОДУ или ДУЧП:

- задача Коши, или задача с начальными условиями, в которой по заданным в начальный момент времени переменным (начальным условиям) определяются значения этих искомых переменных для любого момента времени;
- начально-граничная, или краевая, задача, когда условия на искомую функцию выходного параметра задаются в начальный момент времени для всей пространственной и на границе последней в каждый момент времени (на исследуемом интервале);
- задачи на собственные значения, в формулировку которых входят параметры, определяемые из условия качественного изменения поведения системы (например, потеря устойчивости состояния равновесия или стационарного движения, появление периодического режима, резонанс и т.д.).

Для контроля корректности полученной системы математических соотношений проводят ряд проверок, в частности:

- контроль размерностей величин при использовании принятой системы единиц для значений всех параметров;
- контроль порядков, состоящий из грубой оценки сравнительных порядков складываемых величин и исключения малозначимых параметров (например, если при сложении трех величин одна из них много меньше других, то такой величиной

можно пренебречь);

- контроль характера зависимостей, который заключается в проверке того, что значения выходных параметров модели соответствуют, например, физическому или иному смыслу изучаемой модели;

- контроль экстремальных ситуаций – проверка того, какой вид принимают математические соотношения, а также результаты моделирования, если параметры модели или их комбинации приближаются к своим предельно допустимым значениям;

- контроль граничных условий, включающий проверку того, что граничные условия действительно наложены, что они использованы в процессе построения искомого решения и что значения выходных параметров модели на самом деле удовлетворяют данным условиям;

- контроль математической замкнутости, состоящий в проверке того, что выписанная система соотношений дает возможность получить однозначное решение задачи.

Простота и сложность. Одновременное требование простоты и адекватности модели является противоречивым. С точки зрения адекватности сложные модели являются предпочтительнее простых. В сложных моделях можно учесть большее число факторов, влияющих на изучаемые характеристики объектов. Хотя сложные модели и более точно отражают моделируемые свойства оригинала, но они более громоздки и неудобны в обращении. Поэтому исследователь стремится к упрощению модели, так как простыми моделями легче оперировать. При стремлении к построению простой модели должен соблюдаться основной *принцип упрощения модели: упрощать модель можно до тех пор, пока сохраняются основные свойства и характеристики, присущие оригиналу.*

Этот принцип указывает на предел упрощения. При этом понятие простоты (или сложности) модели является понятием относительным. Модель считается достаточно простой, если современные средства исследования (математические, информационные, физические) дают возможность провести качественный и количественный анализ с требуемой точностью. А поскольку возможности средств исследований непрерывно растут, то те задачи, которые раньше считались сложными, теперь могут быть отнесены к категории простых.

Более трудной задачей является обеспечение простоты/сложности модели сложной системы, состоящей из отдельных подсистем, соединенных друг с другом в некоторую иерархическую и многосвязную структуру. При этом каждая подсистема и каждый уровень имеют свои локальные критерии сложности и адекватности, отличные от глобальных критериев системы.

С целью меньшей потери адекватности упрощение моделей целесообразнее проводить:

1) на физическом уровне с сохранением основных физических соотношений,

2) на структурном уровне с сохранением основных системных свойств.

Упрощение же моделей на математическом уровне может привести к существенной потере степени адекватности. Например, усечение характеристического уравнения высокого порядка до 2 – 3-го порядка может привести к совершенно неверным выводам о динамических свойствах системы.

Жесткость и мягкость модели. Жесткие модели представляют собой модель начальной грубой оценки изучаемого объекта, явления или процесса. По существу, в грубой модели производятся прикидки и делаются самые простые и общие предположения, положенные в основу разработки модели. *Жесткие* модели всегда локальны, и увеличение масштаба прогнозов по этим моделям приводит на каком-то этапе к ошибочным предсказаниям. Примером жесткой модели является простейшая модель роста населения Земли: $\dot{x} = kx$. Она ведет, как известно, к экспоненциальному (т.е. очень быстрому) росту численности населения с течением времени. Применение такой жесткой системы имеет границы применимости, так как при постоянном положительном значении параметра k с некоторого момента эта модель перестает быть адекватной – расчетная численность населения неограниченно возрастает. Естественно, при больших значениях x конкуренция за ресурсы приводит к уменьшению параметра k, и жесткая модель должна быть заменена *мягкой* моделью: $\dot{x} = k(x)x$ [Арнольд, 2000]. Гармонический осциллятор – другой пример так называемой «жёсткой» модели. Она получена в результате сильной идеализации реальной физической системы. Для решения вопроса о её применимости необходимо понять, насколько существенными являются факторы, которыми мы пренебрегли. Иными словами, нужно исследовать «мягкую» модель, получающуюся малым возмущением «жёсткой». Если система сохраняет свое качественное поведение при малом возмущении, говорят, что она структурно устойчива. Гармонический осциллятор – пример структурно-неустойчивой (негрубой) системы. Тем не менее, эту модель можно применять для изучения процессов на ограниченных промежутках времени.

Конечность моделей. Известно, что мир бесконечен, как любой объект, не только в пространстве и во времени, но и в своей структуре (строении), свойствах, отношениях с другими объектами. Бесконечность проявляется в иерархическом строении систем различной физической природы. Однако при изучении объекта исследователь ограничивается конечным количеством его свойств, связей, используемых ресурсов и т.д. Он как бы «вырезает» из бесконечного мира некоторый конечный фрагмент в виде конкретного объекта, системы, процесса и т.д. и пытается познать бесконечный мир через конечную модель этого фрагмента.

Правомерен ли такой подход к исследованию бесконечного мира? Практика отвечает положительно на этот вопрос, основываясь на свойствах человеческого разума и законах Природы, хотя сам разум конечен, но зато бесконечны генерируемые им способы познания мира. Процесс познания идет через непрерывное расширение наших знаний. Это можно наблюдать на эволюции разума, на эволюции науки и техники.

Таким образом, конечность моделей систем заключается, во-первых, в том, что они отображают оригинал в конечном числе отношений, т.е. с конечным числом связей с другими объектами, с конечной структурой и конечным количеством свойств на данном уровне изучения, исследования, описания, располагаемых ресурсов. Во-вторых, в том, что ресурсы (информационные, финансовые, энергетические, временные, технические и т.д.) моделирования и наши знания как интеллектуальные ресурсы конечны, а потому объективно ограничивают возможности моделирования и сам процесс познания мира через модели. Поэтому исследователь (за редким исключением) имеет дело с конечномерными моделями.

Выбор размерности модели (ее степени свободы, переменных состояния) тесно связан с классом решаемых задач. Увеличение размерности модели связано с проблемами сложности и адекватности. При этом необходимо знать, какова функциональная зависимость между степенью сложности и размерностью модели. Если эта зависимость степенная, то проблема может быть решена за счет применения вычислительных систем. Если же эта зависимость экспоненциальная, то «проклятие размерности» [Калман, Фалб, Арбиб, 2010] неизбежно и избавиться от него пока не удается.

Как отмечалось выше, увеличение размерности модели приводит к повышению степени адекватности и одновременно к усложнению модели. При этом степень сложности ограничена возможностью оперирования с моделью, т.е. теми средствами моделирования, которыми располагает исследователь. Необходимость перехода от грубой простой модели к более точной реализуется за счет увеличения размерности модели путем привлечения новых переменных, качественно отличающихся от основных и которыми пренебрегли при построении грубой модели. Эти переменные могут быть отнесены к одному из следующих трех классов:

1) *быстропротекающие* переменные, протяженность которых во времени или в пространстве столь мала, что при грубом рассмотрении они принимались во внимание своими интегральными или осредненными характеристиками;

2) *медленнопротекающие* переменные, протяженность изменения которых столь велика, что в грубых моделях они считались постоянными;

3) *малые переменные* (малые параметры), значения и влияние которых на основные характеристики системы столь малы, что в грубых моделях они игнорировались.

Отметим, что разделение сложного движения системы по скорости на быстропротекающее и медленнопротекающее движения дает возможность изучать их в грубом приближении независимо друг от друга, что упрощает решение исходной задачи. Что касается малых переменных, то ими пренебрегают обычно при решении задачи синтеза, но стараются учесть их влияние на свойства системы при решении задачи анализа.

При моделировании стремятся по возможности выделить небольшое число основных факторов, влияние которых одного порядка и не слишком сложно описывается математически, а влияние других факторов оказывается возможным учесть с помощью осредненных, интегральных или "замороженных" характеристик.

Приближенность моделей. Конечность и простота (упрощенность) модели характеризуют *качественное* различие (на структурном уровне) между оригиналом и моделью. Тогда приближенность модели будет характеризовать *количественную* сторону этого различия.

Можно ввести количественную меру приближенности путем сравнения, например, грубой модели с более точной эталонной (полной, идеальной) моделью или с реальной моделью. Приближенность модели к оригиналу *неизбежна*, существует объективно, так как модель как другой объект отражает лишь отдельные свойства оригинала. Поэтому степень приближенности (близости, точности) модели к оригиналу определяется постановкой задачи, целью моделирования.

Чрезмерное стремление к повышенной точности модели приводит к ее значительному усложнению, и, следовательно, к снижению ее практической ценности. При моделировании сложных (человеко-машинных, организационных) систем точность и практический смысл несовместимы и исключают друг друга. Причина противоречивости и несовместимости требований точности и практичности модели кроется в неопределенности и нечеткости знаний о самом оригинале – его поведении, его свойствах и характеристиках, о поведении окружающей среды, о механизмах формирования цели, путей и средствах ее достижения и т.д.

Экономичность моделей. Данное свойство математических моделей определяется затратами ресурсов (человеческих, материальных, временных, вычислительных и др.) на ее реализацию и эксплуатацию.

*Истинность моделей***.** В каждой модели есть доля истины, т.е. любая модель в чем-то правильно отражает оригинал. Степень истинности модели выявляется только при практическом сравнении её с оригиналом. Что касается малых переменных, то ими пренебрегают обычно при решении задачи синтеза, но стараются учесть их влияние на свойства системы при решении задачи анализа.

При моделировании стремятся по возможности выделить небольшое число основных факторов, влияние которых одного порядка и не слишком сложно описывается математически, а влияние других факторов оказывается возможным учесть с помощью осредненных, интегральных или "замороженных" характеристик.

С одной стороны, в любой модели содержится безусловно истинное, т.е. определенно известное и правильное. С другой стороны, в модели содержится и условно истинное, т.е. верное лишь при определенных условиях. Типовая ошибка при моделировании заключается в том, что исследователи применяют те или иные модели *без проверки условий их истинности*, границ их применимости. Такой подход приводит заведомо к получению неверных результатов.

В любой модели также содержится нечто, могущее быть в условиях неопределенности либо верным, либо ложным. Только на практике устанавливается фактическое соотношение между истинным и ложным в конкретных условиях. Таким образом, при анализе уровня истинности модели необходимо выяснить:

1) точные, достоверные знания;

2) знания, достоверные при определенных условиях;

3) знания, оцениваемые с некоторой степенью неопределенности;

4) знания, не поддающиеся оценке даже с некоторой степенью неопределенности;

5) незнания, т.е. то, что неизвестно.

Таким образом, оценка истинности модели как формы знаний сводится к выявлению содержания в ней как объективных достоверных знаний, так и знаний, приближенно оценивающих оригинал, а также то, что составляет незнание.

Информативность. Модель должна содержать достаточную информацию о системе (в рамках гипотез, принятых при построении модели) и должна давать возможность получить новую информацию.

Полнота. В модели должны быть учтены все основные связи и отношения, необходимые для обеспечения достижения цели моделирования.

Адаптивность. Модель должна быть приспособлена к различным входным параметрам, воздействиям среды.

Управляемость. Модель должна иметь хотя бы один параметр, изменениями которого можно имитировать поведение моделируемой системы в различных условиях.

Эволюционируемость. Возможность развития моделей исследуемого объекта (системы) предыдущего уровня к моделям того же объекта (системы) последующего, более высокого уровня. В модели более высокого уровня учитывается большее число факторов, описывающих сам объект и окружающую его среду, и большая глубина (детализация) учитываемых в исследовании факторов [Аюпов, 2012].

2.4. Общие требования и рекомендации по математическому моделированию

При построении математических моделей объектов, систем, процессов целесообразно придерживаться следующих требований и рекомендаций, которые имеют характер *постоянного контроля* за процессом математического моделирования:

1. Моделирование следует начинать с построения самых грубых моделей на основе выделения самых существенных факторов. При этом необходимо четко представлять как цель моделирования, так и цель познания с помощью данных моделей.

2. Желательно привлекать к работе следующие гипотезы:

- опирающиеся на реальность (подтверждающие актуальность теоретической модели);
- научно-экспериментальные (устанавливающих детерминацию различных закономерностей);
- эмпирические (сформулированные лишь для некоторого класса объектов или явлений);
- экспериментальные (необходимые для проведения эксперимента, подтверждающего или опровергающего выдвинутое научное утверждение);
- статистические (необходимые для сравнения параметров, определяющих и влияющих на достоверность).

3.Необходимо контролировать:

-размерность переменных величин, придерживаясь правила: складываться и приравниваться могут только величины одинаковой размерности;

-порядок складываемых друг с другом величин с тем, чтобы выделить основные слагаемые (переменные, факторы) и отбросить малозначительные. При этом должно сохраняться свойство «грубости» модели: отбрасывание малых величин приводит к малому изменению количественных выводов и к сохранению качественных результатов;

-характер функциональных зависимостей, следуя правилу, проверять сохранность зависимости изменения направления и скорости одних переменных от изменения других. Это правило позволяет глубже понять физический смысл и правильность выведенных соотношений;

-поведение переменных при приближении параметров модели к особым, экстремальным значениям. Обычно в экстремальной точке модель упрощается или вырождается, а соотношения приобретают более наглядный смысл и могут быть проще проверены;

-поведение модели в известных условиях: удовлетворение модели поставленным начальным и граничным условиям; поведение системы как модели при действии на нее типовых входных сигналов;

-получение побочных эффектов и результатов, анализ которых может дать новые направления в исследованиях или потребовать перестройки самой модели.

Постоянный контроль правильности функционирования моделей в процессе исследования позволяет избежать грубых ошибок в конечном результате. При этом выявленные недостатки модели исправляются в ходе моделирования, а не вычисляются заранее.

2.5. Этапы построения и применения математических моделей

Построение математической модели – это центральный этап исследования объекта или процесса. От качества разработанной модели зависит весь последующий анализ объекта исследования. Построение математической модели – это процедура не формальная. Она существенно зависит от исследователя, его опыта и вкуса, и всегда опирается на определенный эмпирический материал.

В общем случае процесс разработки математических моделей состоит из следующих этапов.

1) Обследование объекта моделирования и формулировка технического задания на разработку модели (содержательная постановка задачи)

Этап обследования включает следующие работы:

- выявление основных факторов, механизмов, влияющих на поведение объекта моделирования, определение параметров, подлежащих отражению в модели;

- сбор и проверка имеющихся экспериментальных данных об объектах-аналогах, проведение при необходимости дополнительных экспериментов;

- обзор литературных источников, анализ и сравнение между собой построенных ранее моделей данного объекта (или подобных рассматриваемому объекту);

- анализ и обобщение всего накопленного материала, разработка общего плана создания математической модели.

Содержательная постановка задачи моделирования может уточняться и конкретизироваться в процессе дальнейшей разработки модели. Если объектом моделирования является технологический процесс, машина, конструкция или деталь, то содержательную постановку задачи моделирования называют технической постановкой задачи. Вместе с дополнительными требованиями к реализации модели и представлению результатов содержательная постановка задачи моделирования оформляется в виде *технического задания* на проектирование и разработку модели.

2) Концептуальная и математическая постановка задачи

На данном этапе формулируется совокупность гипотез о поведении объекта, его взаимодействии с окружающей средой, изменении внутренних параметров. Для обоснования принятых гипотез, как правило, используются некоторые теоретические положения и/или экспериментальные данные об объекте. Законченная концептуальная постановка позволяет сформулировать *расчетную схему объекта* (*процесса*) и ее математическое описание.

Совокупность математических соотношений определяет вид оператора модели. Наиболее простые операторы модели получают, используя различные методы аппроксимации экспериментальных данных (интерполяция, метод наименьших квадратов и др.). Более сложные теоретические модели получают на основе каких-либо законов, справедливых для объектов исследования в рассматриваемой области знаний, например, на основе уравнений законов сохранения. В ряде случаев математические соотношения, описывающие поведения объекта, приходится устанавливать самому исследователю.

3) Качественный анализ и проверка корректности модели

В общем случае оператор математической модели может включать в себя различные математические выражения и конструкции.

Для *контроля правильности* и *корректности* полученной системы математических соотношений проводят их качественный анализ и делают ряд проверок, о которых говорилось в разделе 2.3.

4) Выбор и обоснование выбора методов решения задачи

При выборе или разработке метода решения задачи, прежде всего, устанавливается область его применения. Чем шире круг задач, которые объявлены как допустимые для решения данным методом, тем этот метод более универсален. В большинстве случаев четкая и однозначная формулировка ограничений на применение метода затруднительна. Возможны ситуации, когда оговоренные заранее условия применения метода выполняются, однако удовлетворительное решение задачи не получается. Успешное применение удачно выбранного метода в оговоренном заранее классе задач повышает так называемую *надежность алгоритма*. Аналитические методы более надежны, но не всегда применимы. Отказы в решении задач алгоритмическими методами могут проявляться, например, в несходимости итерационного процесса (итерация – последовательное приближение), в превышении погрешностями предельно допустимых значений и т.п.

К наиболее важным машинным (численным) методам относятся:

-интерполяция и численное дифференцирование;

-численное интегрирование;

-определение корней линейных и нелинейных уравнений;

-решение систем линейных уравнений (подразделяют на прямые и итерационные методы);

- решение систем нелинейных уравнений;

- решение задачи Коши для обыкновенных дифференциальных уравнений;

- решение краевых задач для обыкновенных дифференциальных уравнений;

- решение уравнений в частных производных;

- решение интегральных уравнений.

Говоря о численных вычислениях, важно осознавать, что они по своей природе являются приближенными и получаемое численное решение – это не всегда точное математическое решение.

Рассмотрим, например, квадратное уравнение $x^2 - 20x + 1 = 0$, которое имеет меньший корень $x \approx 0.05$. Предположим, что вычисления квадратных корней производятся с точностью до одного знака после запятой, тогда:

$$x_{min} = \frac{20 - \sqrt{20^2 - 4 \cdot 1 \cdot 1}}{2} = 10 - \sqrt{99} \approx 10 - 9{,}9 = 0{,}1.$$

Результат отличается от точного решения на 100 %. Однако его можно улучшить, если перевести вычисляемый квадратный корень в знаменатель:

$$x_{min} = 10 - \sqrt{99} = \frac{(10 - \sqrt{99})(10 + \sqrt{99})}{10 + \sqrt{99}} = \frac{100 - 99}{10 + \sqrt{99}} = \frac{1}{19{,}9} \approx 0{,}05.$$

Т.е. в данном случае удалось найти способ исправить ошибку, обусловленную неудачно выбранным методом.

Вообще, выделяют четыре основных типа ошибок, характерных для приближенных вычислений.

1. *Ошибки исходных данных* имеют место, когда исходные данные носят приближенный характер, например, получены путем физических измерений (любое средство измерения имеет ограниченную точность). Этот вид ошибки рассматривается как шум (говорят, что данные зашумлены). Улучшение точности при наличии таких ошибок не достигается даже при правильной организации процесса вычислений.

2. *Ошибки представления данных (округления, переполнения)*.

3. *Ошибки метода (алгоритма)* возникают вследствие отклонения алгоритмического процесса вычислений от точного (аналитического). Если при неограниченном увеличении числа шагов алгоритма решение дискретной задачи стремится к решению исходной задачи, то говорят, что вычислительный метод *сходится*. Для повышения надежности алгоритмов часто применяют комбинирование различных методов, автоматическую параметрическую настройку методов и т.п.

4. *Ошибки интерпретации*.

5) Поиск решения, разработка алгоритма решения и исследование его свойств, реализация алгоритма в виде программы для ЭВМ.

В случаях, когда решение можно найти аналитическим методом, потребности в разработке специального программного обеспечения, как правило, не возникает. Численный, или приближенный, метод реализуется всегда в виде вычислительного алгоритма. Требования, предъявляемые к алгоритму, указываются в следующем определении. *Алгоритм* – это упорядоченный набор недвусмысленных и выполнимых этапов, определяющий некоторый конечный процесс. Это определение содержит несколько важных требований: 1) требование *упорядоченности* указывает, что этапы алгоритма должны выполняться в некотором определенном порядке, но необязательно один за другим; 2) требование *выполнимости* этапа означает принципиальную возможность его осуществления; 3) требование *недвусмысленности* означает, что во время выполнения алгоритма при любом состоянии процесса информации должно быть достаточно, чтобы полностью определить действия, которые требуется осуществить на каждом этапе; 4) требование *конечности* процесса означает, что алгоритм должен быть *результативен*, т.е. выполнение алгоритма должно приводить к его завершению. Кроме того, к методам и алгоритмам, как и к математическим моделям, предъявляют требования точности и экономичности.

Точность характеризуется степенью совпадения точного решения уравнений заданной модели и приближенного решения, полученного с помощью оцениваемого метода, а *экономичность* – затратами вычислительных ресурсов на реализацию метода (алгоритма).Оценки точности и экономичности бывают теоретическими и экспериментальными. Теоретические оценки обычно характеризуют эффективность применения исследуемого метода не к одной конкретной модели, а к некоторому классу моделей и являются предметом изучения в вычислительной математике. Экспериментальные оценки основаны на определении показателей эффективности решения с помощью набора специально составленных тестовых задач. Процесс создания программного обеспечения обычно идет в следующей последовательности: – составление технического задания на разработку программного обеспечения; –проектирование структуры программного комплекса; – кодирование алгоритма; – тестирование и отладка; – сопровождение и эксплуатация. *Техническое задание* на разработку программного обеспечения оформляют в виде спецификации. Примерная форма спецификации включает следующие семь разделов:

Название задачи – дается краткое определение решаемой задачи, название программного комплекса, указывается система программирования для его реализации и требования к аппаратному обеспечению (компьютеру, внешним устройствам и т.д.).

Описание – подробно излагается математическая постановка задачи, описываются применяемая математическая модель для задач вычислительного характера, метод обработки входных данных для задач не вычислительного (логического) характера и т.д.

Управление режимами работы программы – формируются основные требования к способу взаимодействия пользователя с программой (интерфейс «пользователь–компьютер»).

Входные данные – описываются входные данные, указываются пределы, в которых они могут изменяться, значения, которые они не могут принимать, и т.д.

Выходные данные – описываются выходные данные, указывается, в каком виде они должны быть представлены (в числовом, графическом или текстовом), приводятся сведения о точности и объеме выходных данных, способах их сохранения и т.д.

Ошибки – перечисляются возможные ошибки пользователя при работе с программой (например, ошибки при вводе входных данных), указываются способы диагностики (обнаружения ошибок при работе программного комплекса) и защиты от этих ошибок на этапе проектирования, а также возможная реакция пользователя при совершении им ошибочных действий и реакция программного комплекса (компьютера) на эти действия.

Тестовые задачи – приводятся один или несколько тестовых примеров, на которых в простейших случаях проводится отладка и тестирование программного комплекса.

На этапе *проектирования* формируется общая структура программного комплекса. Вся программа разбивается на программные модули. Для каждого программного модуля формулируются требования по реализуемым функциям и разрабатывается алгоритм, выполняющий эти функции. Определяется схема взаимодействия программных модулей, называемая *схемой потоков данных* программного комплекса. Разрабатывается план, и задаются исходные данные для тестирования отдельных модулей и программного комплекса в целом. Большинство профессиональных программных средств, реализующих математические модели, состоят из трех основных частей:

– *препроцессора* (подготовка и проверка исходных данных модели);

– *процессора* (решение задачи, реализация вычислительного эксперимента);

– *постпроцессора* (отображение полученных результатов) [Дьяконов, 2001].

Возможности пре– и постпроцессора наиболее широко реализуются в современных системах автоматизированного проектирования (САПР), где они в значительной степени сокращают время на получение данных и оценку результатов моделирования.

6) Проверка адекватности модели

Проверка адекватности модели преследует две цели:

– убедиться в справедливости совокупности гипотез, сформулированных на этапах концептуальной и математической постановок;

– установить, что точность полученных результатов соответствует точности, оговоренной в техническом задании.

Проверка разработанной математической модели выполняется путем сравнения с имеющимися экспериментальными данными о реальном объекте или с результатами других, созданных ранее и хорошо себя зарекомендовавших моделей. Как правило, различают качественное и количественное совпадение результатов сравнения. При качественном сравнении требуется лишь совпадение вида функции распределения выходных параметров (убывающая или возрастающая, с одним экстремумом или с несколькими). При количественном сравнении оценивают точность вычисления параметров. В моделях, предназначенных для выполнения различных расчетов, точность зависит от цели исследования и от постановки задачи.

Неадекватность результатов моделирования возможна, по крайней мере, по трем причинам: а) значения задаваемых входных параметров модели не соответствуют допустимой области этих параметров, определяемой принятой системой гипотез; б) принятая система гипотез верна, но константы и параметры в использованных определяющих соотношениях установлены неточно; в) неверна исходная совокупность гипотез.

Все три случая требуют дополнительного исследования как моделируемого объекта (с целью накопления новой дополнительной информации о его поведении), так и самой модели (с целью уточнения границ ее применимости).

7) Практическое использование модели

Практическое использование и анализ результатов моделирования позволяет: выполнить модификацию рассматриваемого объекта, найти его оптимальные характеристики или, по крайней мере, лучшим образом учесть его поведение и свойства; обозначить область применения модели; проверить обоснованность гипотез, принятых на этапе математической постановки, оценить возможность упрощения модели с целью повышения ее эффективности при сохранении требуемой точности; показать, в каком направлении следует развивать модель в дальнейшем.

2.6. Классификация математических моделей, применяемых в землеустройстве

В землеустройстве типовыми являются следующие модели [Волков, 2007]:

1) корреляционные модели, устанавливающие степень влияния факторов;

2) производственные функции, как зависимости результатов производства от производственных факторов;

3) балансовые модели, обосновывающие пропорции воспроизводства;

4) модели оптимизации.

Контрольные вопросы по теме: «Модели и моделирование»

1. Что такое модель?
2. Какие типы моделей существуют?
3. Основные свойства моделей.
4. Что такое моделирование в широком и узком смысле?
5. Что является предметом и объектом моделирования?
6. Модель. Моделирование. Классификация методов моделирования.
7. Математическое моделирование.
8. Математическая модель.
9. Свойства математических моделей.
10. Классификация математических моделей.
11. Требования и рекомендации по математическому моделированию.
12. Этапы построения и применения математических моделей.
13. Классификация математических моделей, применяемых в землеустройстве.

3. Основы компьютерной математики

В этой главе описывается компьютерная математика, возникшая на стыке классической математики и информатики, и ее средства для решения математических задач с помощью компьютеров.

На протяжении многих веков математика делилась на фундаментальную и прикладную. Основы фундаментальной математики достаточно полно изучаются в школе и развиваются в курсах высшей математики в ВУЗах. Здесь учащиеся знакомятся с основными понятиями, аксиомами и теоремами математики, а также с основами доказательства теорем.

Но для подавляющего большинства людей более важным является прикладной аспект математики. В его основе лежит предположение о том, что в рамках принятых математических понятий аксиомы корректны, а все выводы и положения математики верны. Это позволяет широко пользоваться ими на практике и отражает общепринятую в цивилизованном мире технологию разделения труда – математики формулируют первичные понятия, аксиомы и теоремы, а другие специалисты применяют их в меру необходимости.

К примеру, в наше время нет необходимости в совершенстве знать логарифмы для того, чтобы вычислять произведение ряда чисел. С помощью даже простейшего калькулятора выполнять любые арифметические действия намного проще и надежнее, чем в уме или даже на бумаге. Для вычисления специальных математических функций или интегралов достаточно воспользоваться научным (инженерным) микрокалькулятором или подходящей для таких вычислений программой.

К сожалению, такой подход имеет серьезные недостатки. Число первичных понятий, аксиом, теорем и всевозможных следствий из них ныне уже столь велико, что есть риск ошибиться в применении тех или иных из них в конкретных областях науки и техники. Становится очевидной необходимость автоматизации вычислений, как простых, так и самых сложных. И не только численных, но и символьных (алгебраических).

Вот почему все большую роль в выполнении математических вычислений, начиная от самых простых и кончая самыми сложными, стали играть ЭВМ и позже ПК. Стало зарождаться новое понятие и направление на стыке математики и новых информационных технологий – *компьютерная математика*.

Термин «компьютерная математика» является обобщением ряда терминов, таких как символьная математика, компьютерная алгебра, вычислительная математика, конкретная математика, математическое моделирование и компьютерное моделирование. Во всех этих разделах математики речь идет об *автоматизации решения математических задач*, включая моделирование на компьютерах, т.е. о компьютерной математике и о компьютерном моделировании.

Важно отметить, что компьютерная математика является частью прикладной и классической математики.

Нынешний этап развития компьютерной математики характерен поддержкой и продвижением новых направлений в математике, таких, как методы решения некорректных задач, нечеткая логика, нейронные сети, вейвлеты и так далее. Широкое распространение получили системы математического моделирования природных и общественных явлений, систем и устройств.

Современные СКМ – это, прежде всего, мощные электронные справочники и базы данных по всем современным направлениям математики, эффективные средства решения подавляющего большинства математических задач.

3.1. Классификация и обзор средств компьютерной математики

Компьютерная математика, как новое научное направление на стыке математики и информатики, зародилась совсем недавно - в начале 80-х годов XX века. Она включает в себя программные средства для автоматизированного решения широкого класса математических задач.

Программное обеспечение компьютерной математики можно разделить на два широких класса, пересекающихся друг с другом. В классификации ниже в качестве примеров представлены наиболее известные продукты.

1. Языки программирования:

1) специализированные математические языки (Fortran, Octave, Julia, Haskell, R);

2) универсальные языки (C, C++, Java, Perl, Python).

2. Системы и библиотеки для теоретических и прикладных расчётов:

1) системы численных расчётов (Scilab, GNU Octave,

TK Solver);

2) табличные процессоры (Excel, Calc, Google Spreadsheets);

3) матричные системы (Mathcad);

4) статистические пакеты (SPSS, Statistica);

5) системы компьютерной алгебры (WolframAlpha, Mathlab);

6) САПР – системы автоматизированного проектирования (семейство Cax-систем, FEM-системы, класс GIS);

7) прочие специализированные программные пакеты;

8) универсальные системы (Mathematica, Maple, Magma, MATLAB, SageMath).

Исторически первым классом появились универсальные языки программирования и языки, нацеленные на решение прикладных задач. В то же самое время программное обеспечение второго класса имеет свои встроенные языки программирования или позволяет использовать библиотеки, написанные на языках первого класса.

Универсальные системы компьютерной математики (СКМ) претендуют на охват широчайшего спектра математических задач, что достигается расширяемостью их функционала с помощью встроенных языков программирования.

Благодаря реализации СКМ на ПК они доступны педагогам и ученым, студентам и школьникам, причем не только в коллективном, но и в индивидуальном их применении. СКМ используются в университетах и вузах, школах и колледжах (особенно с математическим уклоном). Велика роль таких систем и в автоматизации научно-технических расчетов, и в математическом моделировании различных явлений, систем и устройств.

В рамках данного учебного пособия мы остановимся на СКМ универсального типа Wolfram Mathematica.

3.2. Структура СКМ Mathematica

Каждая СКМ, с точки зрения архитектуры, имеет свои особенности. Рассматриваемая нами универсальная СКМ Mathematica имеет структуру, в состав которой входят следующие основные элементы: *ядро*, *интерфейс, справочная система*, *пакеты математических расширений*. Центральное место занимает *ядро* системы. Оно представляет собой набор встроенных функций и процедур, обеспечивающих минимальную заявленную работоспособность системы: аналитические преобразования, алгебраические преобразования, символьные вычисления, поддержку внутреннего языка WolframLanguage, функции элементарной и высшей математики и пр. В Mathematica ядро может использоваться для распределённых вычислений. *Интерфейс* обеспечивает взаимодействие пользователя с ядром, позволяет загружать исходные данные и выгружать результаты вычислений. Mathematica может работать под управлением различных ОС, несмотря на это элементы её интерфейса остаются схожими (меню, палитры, перемещаемые и масштабируемые окна и т. д.). В 2009 году фирмаWolfram выпустила отдельный сетевой продукт WolframAlpha, использующий другой интерфейс для доступа к вычислениям. Функциональность ядра расширяется с помощью *пакетов расширений*. Пакеты расширений используются для хранения кода, который может быть динамически загружен в рабочую сессию Mathematica. Пакет расширения предоставляет пользователю одну или несколько функций, скрывая от пользователя их реализацию. *Справочная система* обеспечивает получение оперативных справок по любым вопросам работы с системами компьютерной математики с примерами такой работы. Она содержит и многочисленный гипертекстовый справочный материал – математические и физические таблицы, формулы для нахождения производных и интегралов, алгебраические преобразования и т. д.

Описание основных классов данных, функций, математических выражений и других математических конструкций системы Mathematica, используемых при решении задач классической и прикладной математики, приведено в главе 7.

4. Методы и модели решения прикладных задач

4.1. Основные методы решения вычислительных задач

На этапе программной реализации модели и реализации плана экспериментов необходим выбор методов решения *вычислительных задач*. При этом используются три основные группы методов:

Графические – оценочные приближенные методы, основанные на построении и анализе графиков.

Аналитические – решения, строго полученные в виде аналитических выражений (пригодны для узкого круга задач).

Численные – основной инструмент для решения сложных математических задач, основанный на применении различных численных методов.

Аналитическое решение удается получить редко и чаще лишь при упрощенной формулировке задачи в линейном приближении. Основным средством решения является алгоритмический подход, реализующий вычислительный эксперимент на ЭВМ. Получаемое на ЭВМ решение почти всегда содержит некоторую *погрешность*. Напомним, что есть абсолютная погрешность

$$\Delta = x - x_u$$

в виде разности между приближенным x_u и точным x и значениями результата и относительная погрешность

$$\delta = \Delta / x_u .$$

Наличие погрешности решения обусловлено рядом причин. Перечислим основные источники погрешности:

- математическая модель является лишь приближенным описанием реального процесса (погрешность модели);

- исходные данные, как правило, содержат погрешности, так как либо являются результатами экспериментов (измерений), или решениями вспомогательных задач (погрешность данных);

- применяемые для решения задачи методы в большинстве случаев являются приближенными (погрешность метода);

- при вводе исходных данных в ЭВМ, выполнении операций производятся округления (вычислительная погрешность).

Погрешности 1 и 2 – неустранимые на данном этапе решения, для их уменьшения приходится возвращаться вновь к построению математической, а и иногда и концептуальной модели, проводить дополнительное экспериментальное уточнение условий задачи.

Оценка *обусловленности* вычислительной задачи – еще одно обязательное требование при выборе метода решения и построении математической модели.

Пусть вычислительная задача корректна. Теоретически *устойчивость* задачи означает, что ее решение может быть найдено со сколь угодно малой погрешностью, если только гарантировать достаточно малую погрешность входных данных. Однако на практике их точность ограничена.

Как влияют малые, но конечные погрешности входных данных на решение? Как сильно они искажают результат? Ответ на это дает понятие обусловленности задачи, то есть чувствительность решения вычислительной задачи к малым погрешностям входных данных.

Задачу называют *хорошо обусловленной*, если малым погрешностям входных данных $\Delta(x^*)$ отвечают малые погрешности решения $\Delta(y^*)$, и *плохо обусловленной*, если возможны сильные изменения решения.

Введем количественную оценку *степени обусловленности* (число обусловленности ν), которое можно интерпретировать как коэффициент возможного возрастания погрешности в решении по отношению к вызвавшей их погрешности входных данных. Если установлено неравенство между этими погрешностями

$\Delta(y^*) \le \nu_\Delta \Delta(x^*)$,

то ν_Δ – абсолютное число обусловленности или неравенство
$\delta(y^*) \le \nu_\delta \delta(x^*)$,
то ν_δ – относительное число обусловленности (вместо погрешности могут фигурировать их границы).

Вычислительные методы преобразуются к виду, удобному для программной (численной) реализации.

Можно выделить следующие классы *методов*:

– *метод эквивалентных преобразований* – исходную задачу заменяют другой, имеющей то же решение: нахождение корня нелинейного уравнения $f(x) = 0$ сводят к поиску точек глобального минимума $\Phi(x) = (f(x))^2$;

– *методы аппроксимации* – заменяют исходную задачу другой, решение которой близко к решению исходной задачи;

– *методы конечно-разностные*, основанные на замене производных конечными разностями;

– *прямые* (*точные*) *методы* – решение может быть получено за конечное число элементарных операций (арифметические и извлечение корня);

– *итерационные методы*– методы последовательных приближений к решению задачи. Задается начальное приближение решения, строится итерационная последовательность приближений к решению. Если эта последовательность сходится к решению, то говорят, что итерационный процесс сходится. Множество начальных приближений, для которых метод сходится, называются *областью сходимости метода*;

– *метод статистических испытаний* – основан на моделировании случайных величин и построении статистических оценок решений задач (для моделирования больших систем).

Численные методы группируются вокруг типичных математических задач: задач анализа, алгебры, оптимизации, решения дифференциальных и интегральных уравнений, обратных задач (синтез). Этот этап решения заканчивается выбором и обоснованием конкретных численных методов решения, разработкой алгоритма, которые могут быть программно реализованы средствами компьютерной техники.

4.2. Контроль правильности вычислительной модели

Для контроля правильности полученной модели может использоваться ряд приемов:

- анализ размерности – величины в левой и правой части выражения, отдельные слагаемые в каждой из частей должны иметь одинаковую размерность;
- проверка порядков и характеров зависимостей – параметры и переменные, которые в данной задаче выражены величинами большего порядка малости, могут быть исключены из рассмотрения как несущественные, что часто позволяет значительно упростить модель и ее анализ. Характер изменения значений моделируемых величин должен соответствовать их реальному смыслу, не противоречить наблюдаемым данным;
- исследование предельных случаев. Результаты моделирования при крайних значениях параметров модели равных, как правило, нулю или бесконечности, не должны противоречить смыслу (например, энергия реальной физической системы не может оказаться бесконечно большой, время протекания процесса – отрицательным и т.п.). Модель в этом случае существенно упрощается и легче для понимания;
- проверка замкнутости и корректности математической задачи – система математических соотношений должна иметь единственное решение.

Вычислительная задача называется *корректной*, если она удовлетворяет трем требованиям:

- ее решение существует при любых допустимых входных данных;
- это решение единственно (однозначно определено);
- решение непрерывно зависит от данных задачи – устойчиво по отношению к малым возмущениям входных данных.

Решение вычислительной задачи называется *устойчивым* по входным данным X, если оно зависит от входных данных непрерывным образом.

Далеко не все практические задачи являются корректными. К ним, например, не относятся обратные задачи геофизики, астрофизики, спектрографии, распознавания образов, синтез и многие другие важные прикладные проблемы. Свойство корректности задачи имеет большое значение для выбора метода решения. К некорректным задачам неприменимы обычные численные методы вычислительной математики. Строгий анализ корректности во многих случаях математически сложен, и ограничиваются проверкой соответствия количества неизвестных и связывающих их уравнений в модели.

4.3. Задача моделирования полета камня

4.3.1. Постановка задачи моделирования

Началу моделирования предшествует постановка содержательной задачи моделирования, переход от когнитивной модели к формулировке в словесной форме основных вопросов об объекте моделирования. Правильная постановка задачи очень важна, так как ошибка здесь потребует вернуться к построению модели с самого начала. Содержательная постановка задачи, называемая в технических дисциплинах техническим заданием, в дальнейшем уточняется и конкретизируется, однако принципиальные, основные положения остаются неизменными [Советов, Яковлев, 2001, Андрейченко, 2000, Асанов, 2007, Ашихмин, 2005, Голубева, 2013, Мышкис, 2016 и др.].

Пример 4.1. Дать постановку задачи «Полет камня», позволяющую описать полет камня, брошенного под некоторым углом к горизонту.

Модель должна позволять:

Вычислять положение камня в любой момент времени.

Исходные данные: масса камня, начальные координаты, начальная скорость и начальный угол бросания камня.

4.3.2. Концептуальная формулировка задачи

На основе содержательной модели разрабатывается концептуальная формулировка задачи моделирования.

Пример 4.2. Концептуальная постановка задачи «Полет камня». Движение камня может быть описано в соответствии с законами классической механики Ньютона.

Гипотезы, принятые для модели:

- камень будем считать материальной точкой массой m, положение которой совпадает с центром масс камня;
- движение происходит в поле силы тяжести Земли с постоянным ускорением свободного падения g и описывается уравнениями классической механики Ньютона;
- движение камня происходит в одной плоскости, перпендикулярной поверхности Земли;
- сопротивлением воздуха на первых порах пренебрегаем;
- в месте падения камня земля абсолютно не упругая (нет отскока камня).

В качестве параметров движения будем использовать координаты (x,y) и скорость $v(v_x,v_y)$ центра масс камня.

Концептуальная постановка задачи на основе принятых гипотез может быть следующей:

Определить закон движения материальной точки массой m под действием силы тяжести, если в начальный момент времени t_0=0 известны начальные координаты точки x_0 и y_0, ее начальная скорость v_0 и угол броска α_0.

Таким образом, модель является простой – объект материальная точка не имеет внутренней структуры. Учитывая типичные скорости и высоту броска камня, можно считать постоянным ускорение свободного падения. Переход от трехмерных координат к плоскости значительно упрощает решение задачи. Он вполне допустим, если камень не подкручивается при броске и нет ветрового воздействия на камень. Пренебрежение сопротивлением воздуха, как будет показано далее, приводит к значительной систематической ошибке результатов моделирования.

4.3.3. Построение математической модели

Теперь перейдем к составлению *математической модели* полета камня с учетом принятых выше гипотез и заданных условиях движения.

Пример 4.3. Математическая постановка задачи «Полет камня».

Исходя из законов теоретической механики и принятых предположений, сформируем уравнения модели полета камня.

По оси x на камень не действуют никакие силы, по оси y – действует сила тяжести. Согласно законам Ньютона имеем уравнения движения по оси $\boldsymbol{x}$ и оси $\boldsymbol{y}$:

$$m \cdot \frac{d^2x}{dt^2} = 0,\ m \cdot \frac{d^2y}{dt^2} = -m \cdot g,\ v_x = \frac{dx}{dt},\ \ v_y = \frac{dy}{dt} \tag{4.1}$$

при следующих начальных условиях:

$t_0 = 0,\ x(t_0) = x_0,\ y(t_0) = y_0,\ v_x(t_0) = v_0 \cdot cos\alpha_0,\ \ v_y(t_0) = v_0 \cdot sin\alpha_0.$

Найти зависимости $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$.

Математическая постановка соответствует решению задачи Коши для системы обыкновенных дифференциальных уравнений (ОДУ) с заданными начальными условиями. Известно, что решение задачи Коши существует и единственно. Количество искомых переменных равно количеству дифференциальных уравнений. Таким образом, математическая модель корректна. Решение этой задачи можно найти, например, в учебниках по физике и теоретической механике [Бурсиан, 1991, Мещерский, 1986].

На основании классификационной таблицы 1 охарактеризуем полученную математическую модель следующим образом:

1) по сфере применения эта математическая модель учебная;

2) по способу получения – теоретическая;

3) по форме представления – аналитическая и графическая;

4) вид оператора модели – дифференциальный;

5) по свойствам параметров оператора − это модель линейная с сосредоточенными, стационарными параметрами;

6) по фактору времени − динамическая;

7) по количеству входов/выходов − скалярная;

8) по количеству переменных состояния − одномерная;

9) по характеру переменных – непрерывная, детерминированная.

4.3.4. Выбор метода решения

Пример 4.4. Выбор метода решения задачи «Полет камня». Задача может быть решена как аналитически, так и численно. Рассмотрим оба варианта.

Аналитическое решение.

Из соотношений (4.1) запишем систему ОДУ первого порядка:

$$\frac{dv_x}{dt} = 0, \quad \frac{dv_y}{dt} = -g, \ v_x = \frac{dx}{dt}, \ v_y = \frac{dy}{dt}.$$

После интегрирования получим:

$$v_x(t) = C_1, v_y(t) = C_2 - g \cdot t, x(t) = C_3 + C_2 \cdot t, y(t) = C_4 + C_2 \cdot t - \frac{gt^2}{2}.$$

Определив константы интегрирования из начальных условий, окончательно запишем:

$$x(t) = x_0 + v_0 \cdot cos\alpha_0 \cdot t, y(t) = y_0 + v_0 \cdot sin\alpha_0 \cdot t - \frac{gt^2}{2},$$

$$v_x(t) = v_0 \cdot cos\alpha_0, , \ v_y(t) = v_0 \cdot sin\alpha_0 - g \cdot t.$$

Из аналитического решения вытекает, что полет камня при отсутствии сопротивления воздуха происходит строго по параболической траектории, причем она на участках полета камня вверх и вниз симметрична.

Численное решение.

Численное решение может быть найдено только для конкретных значений параметров модели, например m=200 г, α_0=45°, v_0=20 м/с, g=9.8 м/с^2, x_0=0, y_0=1 м. Существует большое количество численных методов решения систем ОДУ [Марчук, 1989 и др.]. Для данной задачи можно использовать любой из них, простейшим из которых является явный метод Эйлера.

Очевидно, что вычисления нужно вести до момента времени t_k, когда $y(t_k)$ станет равным 0, т.е. камень упадет на землю (при оговоренных условиях в концептуальной постановке задачи). В более сложных случаях выбор численного метода решения является ответственным этапом, необходимо учитывать жесткость системы ОДУ, скорость работы процессора, сходимость и точность метода.

Пример 4.5. Программная реализация математической модели «Полет камня».

Будем использовать программную среду компьютерной системы Mathematica. Особенность ее работы – максимальное приближение записи алгоритма решения к естественной математической форме.

Система Mathematica имеет несколько встроенных функций для решения систем ОДУ (см. глава 7). При их использовании достаточно корректно записать условие задачи и вызвать соответствующую команду – см. рис. 1, на котором показано решение задачи моделирования полета брошенного камня в условиях отсутствия сопротивления воздуха.

Моделирование полета камня без учета сопротивления воздуха

```
x0 = 0; y0 = 1; g = 9.8; v0 = 20; α = 30.*Degree; ds = DSolve[{x''[t] == 0,
   y''[t] == -g, x'[0] == v0*Cos[α], y'[0] == v0*Sin[α],
   x[0] == x0, y[0] == y0}, {x[t], y[t]}, t];
xt[t] := x[t] /. First[First[ds]]; yt[t] := y[t] /. Last[First[ds]];
y1 = ∂t yt[t]; s = Solve[y1 == 0, t];
tm = t /. First[s]; xm = xt[t] /. t → tm; ym = yt[t] /. t → tm;
sy = Solve[yt[t] == 0, t];
tk = t /. First[Last[sy]]; sk = xt[t] /. t → tk; hm = ym; p1 = ParametricPlot
[Evaluate[{xt[t], yt[t]}, {t, 0, tk}], PlotStyle → {RGBColor[0, 0, 1],
   Thickness[0.0011]}, Frame → True, GridLines → Automatic,
 PlotRange → {{0, 40}, {0, 8}}]
```

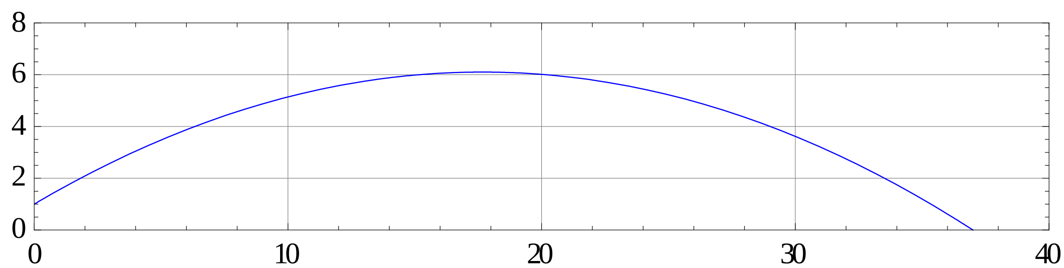


Рис.1. Программная реализация решения в среде Mathematica

задачи «Полет камня» без учета сопротивления воздуха

Расчеты показывают, что траектория полета камня является квадратичной параболой.

4.3.6. Проверка адекватности модели

Необходимым требованием, которому должна отвечать каждая модель является адекватность – соответствие результатов, полученных при моделировании, данным эксперимента, теоретическим положениям или тестовым примерам. Ограничимся нашим примером.

Пример 4.6. Анализ адекватности модели «Полет камня».

Результаты, полученные по приведенной выше модели, будут существенно отличаться от действительных. Особенно это будет заметно, если уменьшать вес камня, например, взяв вместо камня аналогичных размеров кусок пенопласта. Очевидно, это можно объяснить только грубым упрощением в принятой системе гипотез – пренебрежением сопротивления воздуха. Учет этого фактора требует изменения концептуальной модели и всех последующих этапов решения. А именно - требуется учитывать сопротивление воздуха.

Теперь уточним математическую модель. Сила сопротивления воздуха направлена против направления движения камня:

$$m \cdot \frac{d^2x}{dt^2} = -Fmp_x,\ m \cdot \frac{d^2y}{dt^2} = -m \cdot g - Fmp_y,\ v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt} \qquad (4.2)$$

Сопротивление воздуха зависит от скорости движения тела и может быть описано следующей эмпирической формулой

$$Fmp = Av + Bv^3\ ,$$

где A=0.1 Н·с/м, B=10^{-3} Н·с^3/м^3.

Поскольку нелинейные задачи в аналитическом виде чаще всего не решаются, выберем численный метод, который достаточно просто реализуется функциями Mathematica.

4.3.7. Анализ результатов моделирования

Анализ результатов моделирования – необходимый этап грамотного решения любой задачи. Такой анализ позволяет:

- получить представление о поведении объекта в различных условиях, найти оптимальные характеристики процесса;
- определить область применения модели;
- оценить обоснованность принятых при построении модели гипотез, определить пути ее совершенствования.

Пример 4.7. Анализ результатов решения задачи численным методом – рис. 2. Из приведенного примера явно видно, что при учете сопротивления воздуха траектория полета камня заметно отличается от параболической, она заметно круче на спаде. И камень пролетает меньшее расстояние.

Моделирование полета камня с учетом сопротивления воздуха

```
x0 = 0; y0 = 1; g = 9.8; v0 = 20; α = 30.*Degree; m = 0.2; a = 0.1; b = 0.001;
tk = 2; ds = NDSolve[{vx'[t] == -(1/m (a*vx[t]*Cos[α] + b*(vx[t]*Cos[α])^3)),
   vy'[t] == -g - (1/m (a*vy[t]*Sin[α] + b*(vy[t]*Sin[α])^3)), x'[t] == vx[t],
   y'[t] == vy[t], vx[0] == v0*Cos[α], vy[0] == v0*Sin[α], x[0] == x0, y[0] == y0},
  {vx, vy, x, y}, {t, 2}, Method → "BDF"]

{{vx → InterpolatingFunction[{{0.`, 2.`}}, "<>"],
  vy → InterpolatingFunction[{{0.`, 2.`}}, "<>"],
  x → InterpolatingFunction[{{0.`, 2.`}}, "<>"],
  y → InterpolatingFunction[{{0.`, 2.`}}, "<>"]}}

p4 = ParametricPlot[Evaluate[{x[t], y[t]} /. ds], {t, 0, 2}, PlotStyle →
   {RGBColor[0, 0, 1], Thickness[0.0011]}, Frame → True, GridLines →
   Automatic, PlotRange → {{0, 40}, {0, 8}}]
```

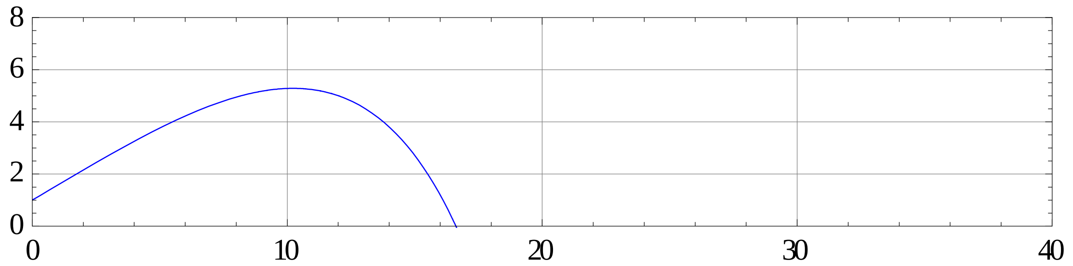


Рис. 2. Программная реализация решения в среде Mathematica задачи «Полет камня» с учетом сопротивления воздуха.

Таким образом, рассмотрена классическая задача моделирования полета камня, которая свелась к решению системы дифференциальных уравнений, описывающих такой полет. Другие примеры моделирования, основанные на решении систем дифференциальных уравнений приведены в разделах 4.4, 4.5.

Методические указания

Внимательно изучить определения моделей и моделирования. Разобрать пример исследования полета камня. Ответить на вопросы.

Вопросы

1. Что такое модель и какие виды моделей вы знаете?
2. Какие виды моделирования существуют?
3. От чего зависит точность моделирования?
4. Какие виды погрешности моделирования вы знаете?
5. Какие программные продукты используют для моделирования?
6. Как осуществляется подготовка к моделированию?
7. Какими возможностями в моделировании обладают СКМ?
8. Чем отличается траектория полета камня без учета сопротивления воздуха и с его учетом?

4.4. Хаос и моделирование аттрактора Лоренца

Броуновское движение частиц и колебания в системе Даффинга являются проявлениями *хаоса* в природе (рис. 3). Наблюдая за изменениями курса акций, сходами ледников и снежных лавин или за колебаниями температуры, можно убедиться в том, что наряду с вполне предсказуемыми изменениями некоторого параметра (например, повышением температуры летом и понижением зимой) нередко наблюдаются хаотические изменения, которые трудно или невозможно заранее предвидеть. Иногда «развал», казалось бы, устойчивой системы приводит к резким изменениям ее поведения – наш «черный вторник» или обвал рубля в 1988 году тому наглядные примеры. Эти и множество других примеров показывают, что хаотическое поведение систем более характерно для природы, чем стационарное. Так что хаос становится одним из важных объектов изучения современной наукой. Его моделирование осуществляется на основе численных методов.

Решение дифференциального уравнения Даффинга

```
c = 0.2; ω = 1.0; a = 0.25; x0 = 1; v0 = 0.6;
daf = NDSolve[{x1'[t] + c*x1[t] - x[t] + x[t]^3 == a*Cos[ω*t],
   x'[t] == x1[t], x1[0] == v0, x[0] == x0}, {x, x1}, {t, 200}]

{{x → InterpolatingFunction[{{0.`, 200.`}}, "<>"],
  x1 → InterpolatingFunction[{{0.`, 200.`}}, "<>"]}}

p3 = ParametricPlot[Evaluate[{x[t], x1[t]} /. daf], {t, 0, 200},
  PlotStyle → {RGBColor[0, 0, 1], Thickness[0.0011]},
  Frame → True, GridLines → Automatic, PlotRange → {{-2, 2}, {-1, 1}}]
```

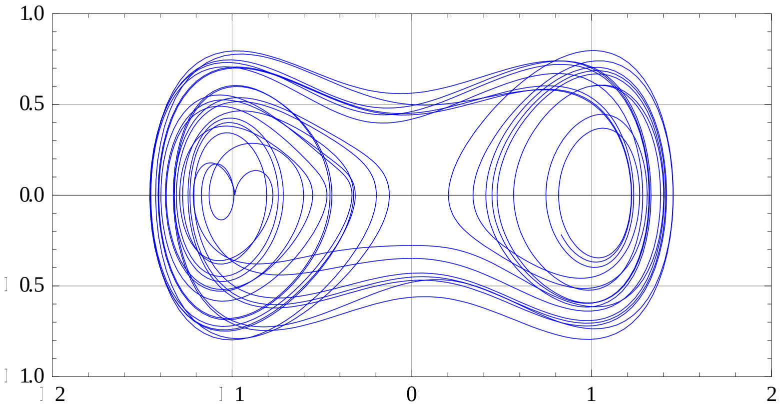


Рис. 3. Программная реализация решения в среде Mathematica уравнения Даффинга.

Чем сложнее система и чем большим количеством дифференциальных уравнений она описывается, тем больше вероятность возникновения в системе хаотических режимов – даже если она автономна. Изучение этого вопроса показало, что уже в системах из трех дифференциальных уравнений возможно возникновение хаотических режимов. Наглядным примером этого является аттрактор Лоренца, пример моделирования которого представлен на рис. 4. При определенных значениях параметров r и b и начальных параметров переменных поведение аттрактора (он в этом случае называется *странным аттрактором*) очень напоминает хаотические колебания в системе Даффинга.

Аттрактором в теории колебаний называется притягивающая область в фазовом пространстве. Причины неустойчивости аттракторов связаны с экспоненциальной неустойчивостью системы в малых областях фазового пространства. При этом наблюдаются хаотические переходы из одной области фазового пространства в другие области, но при этом колебания могут не выходить из некоторой более обширной области фазового пространства. «Обвал» системы означает переход в некоторое состояние, резко отличающееся от других состояний, т.е. выход за пределы ограниченного фазового состояния системы. Такое состояние может оказаться устойчивым и приводит к переходу системы в статическое состояние, при котором изменения ее параметров отсутствуют.

Аттрактор Лоренца

```
σ = 10; r = 35; b = 2.7; x0 = 10; y0 = 20; z0 = 10;

d5 = NDSolve[{x'[t] == -σ * (x[t] - y[t]), y'[t] == -x[t] * z[t]
     + r * x[t] - y[t], z'[t] == x[t] * y[t] - b * z[t], x[0] == x0,
   y[0] == y0, z[0] == z0}, {x[t], y[t], z[t]}, {t, 0, 50}]

{{x[t] → InterpolatingFunction[{{0.`, 50.`}}, "<>"][t],
  y[t] → InterpolatingFunction[{{0.`, 50.`}}, "<>"][t],
  z[t] → InterpolatingFunction[{{0.`, 50.`}}, "<>"][t]}}

ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /. d5, {t, 0, 30}],
 PlotPoints → 10 000]
```

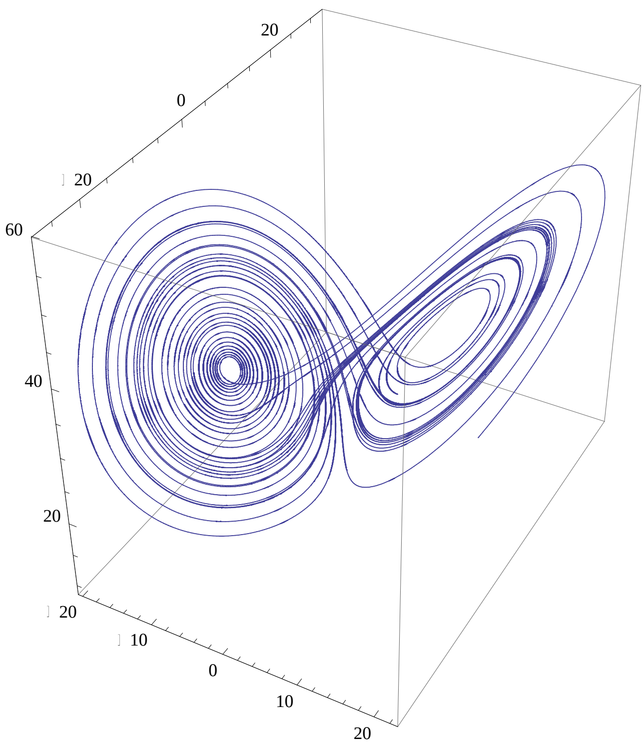


Рис. 4. Программная реализация численного решения в среде Mathematica модели аттрактора Лоренца.

Рассмотренная математическая модель Лоренца на основании классификационной таблицы 1 может быть охарактеризована так:

1) по сфере применения эта математическая модель прикладная;

2) по способу получения математической модели – теоретическая;

3) по форме представления – аналитическая и графическая;

4) вид оператора модели – дифференциальный;

5) по свойствам параметров оператора – это модель нелинейная с сосредоточенными, нестационарными параметрами;

6) по фактору времени – динамическая;

7) по количеству входов/выходов – матричная;

8) по количеству переменных состояния – многомерная;

9) по характеру переменных – непрерывная, детерминированная.

4.5. Моделирование биологических систем

4.5.1. Модель системы «хищник-жертва» Лотки-Вольтера

Рассмотрим типичную земную задачу о совместном проживании хищников и их жертв. Поскольку жертвы поедаются хищниками, число жертв начинает сокращаться, а число хищников – расти. Однако так не может продолжаться долго. Через некоторое время хищникам начинает не хватать пищи, и их популяция перестает расти и даже уменьшается. В итоге жертвы начинают размножаться более интенсивно и их число растет. Далее эти процессы повторяются, и в них обнаруживается периодичность.

Одной из первых моделей такой системы «хищник-жертва» стала модель Лотки и Вольтера [Вольтера, 1976] (рис.5). Пусть y_0 и y_1 – число жертв и хищников. Предположим, что относительный прирост жертв y_0'/y_0 равен $a–by_1$, где $a>0$– скорость размножения жертв в отсутствие хищников, $–by_1$ ($b>0$) – потери от хищников. Развитие популяции хищников зависит от количества пищи (жертв), при отсутствии пищи ($y_0=0$) относительная скорость изменения популяции хищников равна $y_1'/y_1=c$, где $c>0$. Наличие пищи компенсирует убывание хищников, и при $y_0>0$ имеем $y_1'/y_1=(–c+dy_0)$, где $d>0$.

Модель системы "хищник-жертва" Лотки - Вольтера

```
a = 1; b = 3; c = 1; d = 1; vnds1 = NDSolve[{x'[t] ==
    (a - b*y[t]) * x[t], y'[t] == (-c + d*x[t]) * y[t],
   x[0] == 3.00, y[0] == 1.00}, {x, y}, {t, 100}]

{{x → InterpolatingFunction[{{0.`, 100.`}}, "<>"],
  y → InterpolatingFunction[{{0.`, 100.`}}, "<>"]}}

q1 = ParametricPlot[Evaluate[{x[t], y[t]} /. vnds1],
  {t, 0, 100}, AspectRatio → 1.1, PlotRange →
   {{0, 5}, {0, 1.5}}, Frame → True, Ticks → {{0, 2}, {0, 1, 1, 5}}]
```

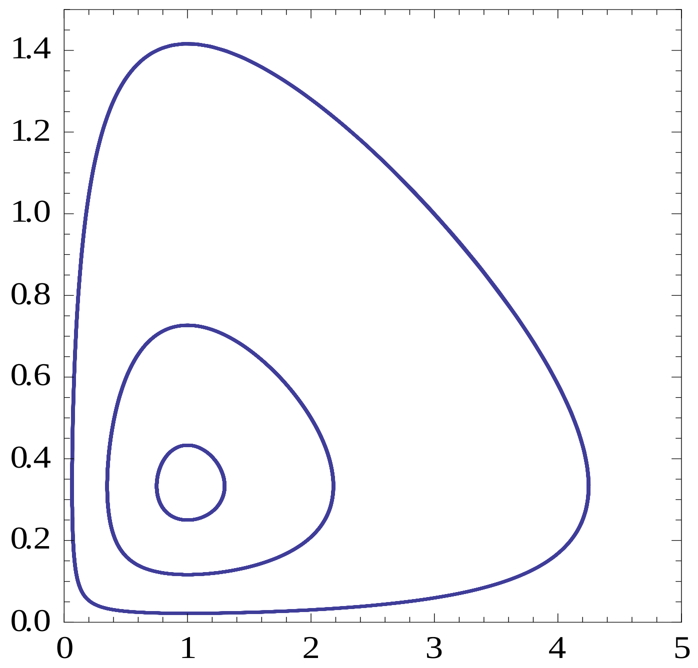


Рис. 5. Численный расчет модели системы «хищник-жертва» Лотки-Вольтера в трех вариантах в среде Mathematica.

Рассмотренная модель достаточно универсальна. Она может описывать не только изменение популяций хищников и жертв, но и поведение конкурирующих фирм, рост народонаселения, численность воюющих армий, изменение экологической обстановки, развитие науки и пр. Рекомендуется поэкспериментировать с этой моделью и убедиться, что моделируемые процессы могут иметь не только колебательный, но и апериодический характер.

4.5.2. Модель системы «хищник-жертва» с логистической поправкой

Колебания популяций хищников и жертв на самом деле наблюдаются не всегда. Нередко мы наблюдаем стабильное количество тех и других, хотя процесс съедения жертв хищниками идет постоянно. Такой случай требует введения некоторой логистической поправки, которая учитывается в несколько иной модели системы «хищник-жертва», представленной на рис. 6.

Модель Лотки-Вольтера с логистической поправкой

```
a = 3; b = 2; c = 1.8; d = 1; α = 0.08; vnds4 = NDSolve[{x'[t]
    == (a - b * y[t]) * y[t] - α * x[t]^2, y'[t] == (-c + d * x[t]) * y[t]
     - α * y[t]^2, x[0] == 3.00, y[0] == 1.00}, {x, y}, {t, 100}]

{{x → InterpolatingFunction[{{0.`, 100.`}}, "<>"],
  y → InterpolatingFunction[{{0.`, 100.`}}, "<>"]}}

q4 = ParametricPlot[Evaluate[{x[t], y[t]} /. vnds4],
  {t, 0, 100}, AspectRatio → 1.1, PlotRange → {{1, 3.5}, {0.5, 2.5}},
  Frame → True, Ticks → {{0, 2}, {0, 1, 1, 5}}]
```

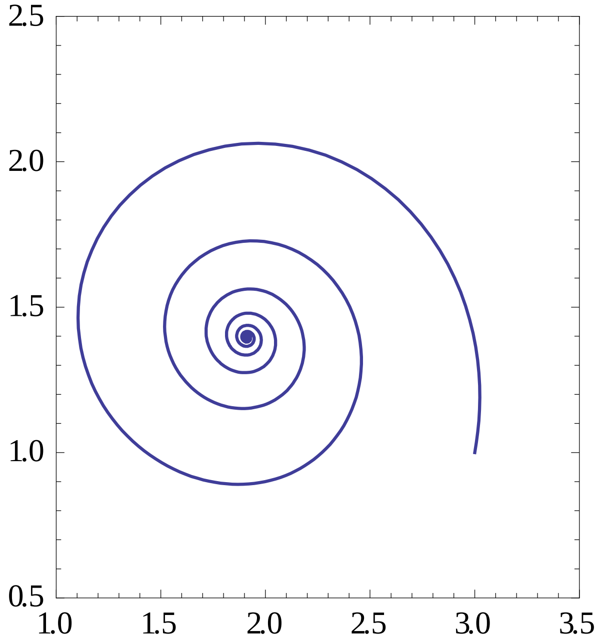


Рис. 6. Моделирование в среде Mathematica системы «хищник-жертва» Лотки-Вольтера с логистической поправкой.

Дополнительный параметр α в этой модели позволяет управлять затуханием осцилляций (колебаний) модели. Как нетрудно заметить, при указанных параметрах модели колебательный процесс в ней явно затухает и устанавливается длительное равновесие между числом хищников и жертв. Фазовый портрет приобретает устойчивый *фокус*. Форма фазового портрета свидетельствует о довольно малой нелинейности этой системы. Поэтому колебания напоминают затухающую синусоиду. Однако при $a < 0$ образуется неустойчивый фокус, и колебания начинают нарастать.

Методические указания

Материал этой главы в основном предназначен для углубленного изучения темы математического и компьютерного моделирования. Разделы ориентированы на преподавателей и выборочно могут использоваться студентами.

Контрольные вопросы

1. Какие примеры моделирования с использование готовых математических моделей вы знаете?
2. Почему в биологических моделях наблюдаются автоколебания, могут ли при этом существовать апериодические режимы работы?
3. Какие режимы работы возможны в линейных системах второго порядка?
4. Как влияет на форму колебаний нелинейность систем?
5. Что такое фазовый портрет и как он строится?
6. В каких системах наступают случайные (хаотические) колебания?
7. В чем суть линейного программирования?
8. Какие функции в системе Mathematica обеспечивают поиски экстремумов и какие виды экстремумов они могут находить?
9. Для чего предназначен комплекс Mathematica?
10. Основные виды математических методов, применяемых в математических расчетах.
11. Какие виды программного математического обеспечения существуют?
12. Что делает необходимым внедрение математических методов и моделирования в землеустроительное производство?

5. Постановка и решение оптимизационных задач

Целью высшего профессионального образования является подготовка квалифицированного специалиста соответствующего уровня и профиля, конкурентно-способного на рынке труда, свободно владеющего профессиональными знаниями и навыками, ориентирующегося в смежных областях знаний и деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту.

Будущим специалистам различных профилей в практической деятельности придется постоянно сталкиваться с задачами поиска и принятия наилучших возможных (оптимальных) решений, большое число которых возникает в экономике, менеджменте, технике. Множество аналогичных проблем придется решать и выпускникам факультета землеустройства и кадастра.

Для решения землеустроительных задач различных классов используются разнообразные математические модели, позволяющие проводить анализ использования земельных, трудовых и материальных ресурсов, выявлять тенденции развития производства, находить оптимальные варианты устройства территории, определять оптимальные варианты проектов землеустройства и т.д. Например, решение задачи по определению размеров крестьянского (фермерского) хозяйства (искомые переменные – общая земельная площадь, площадь пашни, состав земельных угодий и отраслей), которые, исходя из специализации хозяйства, его трудоопеспеченности и фондообеспеченности (основные ограничения), давали бы максимальную прибыль (максимальное значение целевой функции) [Волков, 2007].

Оптимизационные модели в землеустройстве делятся на две разновидности: *комбинированные* и *дифференцированные*.

При *комбинированном моделировании* все вопросы землеустроительного проекта решаются комплексно по всем составным частям и элементам проекта. Этот вид моделирования является более правильным, однако он приводит к громоздким задачам, решение которых затруднено.

Дифференцированное моделирование заключается в последовательном решении частных задач проекта в сочетании с традиционными методами. Модели при этом получаются значительно меньшего объема и их решение существенно облегчается. Применение дифференцированного моделирования в землеустройстве объясняется сложностью и многообразием решаемых вопросов.

Методами математического программирования решается широкий класс экономико-математических задач, позволяющих найти экстремальное (max, min) значение целевой функции при ограниченных ресурсах.

Основу экономико-математического моделирования составляет математическое моделирование экономических систем. Как правило, все землеустроительные экономико-математические задачи имеют многовариантный, альтернативный характер, и основной вопрос заключается в том, как из множества допустимых вариантов выбрать наилучший, оптимальный вариант по заданному критерию. Математически такие задачи сводятся к отыскиванию максимумов или минимумов различных функций, т. е. к решению задач на экстремум.

При решении задач на экстремум применяют так называемые *методы математического программирования*, которые находят широкое применение при решении различных инженерно-экономических задач. Термин «программирование» указывает на тот факт, что эти методы позволяют последовательно находить программы действий, начиная от исходного допустимого плана до наилучшего решения.

Формулировка задачи математического программирования включает целевую функцию вида $f(x_1, x_2, \cdots, x_n) \rightarrow max(min)$ которая является зависимостью критерия оптимизации, а именно какого либо обобщенного показателя, в качестве которого может выступать, например, доход, издержки, себестоимость от параметров модели $x_1, x_2, \cdots, x_n$ (искомых переменных величин, которые могут принимать различные численные значения). На эти неизвестные налагаются определенные условия, образующие так называемую *систему ограничений*. Ограничениями служат уравнения или неравенства, построенные в соответствии с логическим содержанием задачи: $q_i(x_1, x_2, \dots, x_n) \leq=\geq 0; i = 1,2, \dots, m$. Иногда данная система ограничений дополняется другими условиями, например, условиями неотрицательности переменных: $x_j \geq 0$. В задаче требуется найти такой набор значений неизвестных, который удовлетворяет системе ограничений и дает целевой функции наибольшее или наименьшее значение.

В зависимости от характера функции и системы ограничений различают линейные и нелинейные задачи математического программирования.

Если система ограничений и целевая функция линейны относительно искомых величин $x_1, x_2, \cdots, x_n$, то имеется задача *линейного программирования*.

Линейное программирование выражает совокупность приемов, в которых для решения задач количественные зависимости могут быть выражены с помощью линейных уравнений и неравенств с неизвестными в первой степени.

Если же в задаче фигурирует хотя бы одно нелинейное выражение, программирование будет *нелинейным*.

Нелинейное программирование применяют для решения задач, зависимости в которых выражаются нелинейными целевой функцией и ограничениями и результаты (кривые – гипербола, парабола и др.) при этом возрастают или убывают непропорционально изменению масштабов использования ресурсов.

Математическое программирование объединяет задачи обоих типов.

Имеются и другие классификации задач математического программирования.

Целочисленное программирование используется для решения задач, требующих ответа в целых числах.

К задачам, в которых исходные параметры выражены вполне определенными числами, применимы методы, разработанные для условий полной информации; если же эти параметры случайные величины, - используют *методы стохастического программирования*.

Задачи, для которых необходимо вычислить экстремум на одном этапе, являются *одноэтапными или статическими*; многоэтапные задачи требуют применения методов *динамического программирования,* которое используется для решения задач, в которых переменные рассматриваются в динамике и решение их определяют в зависимости от изменения целевой функции во времени.

Если исходные параметры оптимизационных задач изменяются в некоторых пределах, то их исследуют с помощью методов *параметрического программирования.*

Если же параметры задач могут принимать лишь ограниченное число дискретных значений (при использовании некоторых стандартов), то применяют методы *дискретного программирования.*

Кроме названных задач математического программирования в экономических исследованиях широкое применение находят и другие методы количественного анализа (*корреляционно-регрессионного, дисперсного* и др.), а также *методы межотраслевого баланса*, основанные на выявлении и количественной оценке взаимосвязей, сложившихся между различными отраслями производства в регионе.

Постановка и решение задач оптимизации в конечномерных пространствах подробно представлены в учебном пособии [Лутманов, 2007]. В последующих разделах рассмотрим лишь теоретические основы постановки и решения задач одномерной и двумерной оптимизации.

5.1.Постановка задачи одномерной оптимизации

Пусть задана функция

$y = f(x)$,

определенная на множестве $X \subset R^1 = (-\infty, +\infty)$, которую требуется исследовать на наибольшее и наименьшее значения. При этом, задача максимизации функции $f(x)$ на множестве X может быть сведена к задаче минимизации функции $-f(x)$ на том же множестве X, поэтому в дальнейшем ограничимся лишь задачами минимизации функции.

Пусть для точек $x_1, x_2 \in X$ имеет место неравенство $f(x_1) \leq f(x_2)$. Тогда очевидно, что точка x_1 «лучше», чем точка x_2. В случае, когда существует такая точка $x_* \in X$, что

$$f(x_*) \leq f(x) \text{ , } \forall x \in X, \qquad (5.1)$$

то x_* является «наилучшим» элементом из множества X и он доставляет решение так поставленной задачи оптимизации. В математическом анализе точка x_*, удовлетворяющая условию (5.1), называется точкой минимума функции $f(x)$ на множестве X. Для этого случая применим следующие обозначения:

$f(x_*) = \min_{x \in X} f(x) = f_*$.

Нетрудно заметить, что задача оптимизации имеет два аспекта решения, каждый из которых формализуется в виде отдельной задачи.

Задача А. Найти $x_* \in X$, при котором

$f(x_*) = \min\limits_{x \in X} f(x)$

Задача Б. Найти $f_* \in R^1$, при котором

$f_* = \min_{x \in X} f(x)$.

Возникают вопросы: всегда ли разрешима *задача А* (следовательно, и *задача Б*), единственно ли это решение, если оно существует? Обозначим

$X_* = \{x_* | f(x_*) = \min_{x \in X} f(x)\}$.

В зависимости от условий конкретных задач, возможны следующие исходы:

1. Решение задачи существует и единственно.
2. Решения нет (и *задачи А и задачи Б).*
3. Существует бесконечное множество решений *задачи А.*

Заметим, что если решение *задачи Б* существует, то оно всегда единственно.

Возможна более широкая постановка *задачи А,* когда ищутся точки так называемого локального минимума.

Определение. Точка $x_* \in X$ называется точкой локального минимума функции $f(x)$ на множестве X, если существует такое $\alpha > 0$, что $f(x_*) \leq f(x)$ для всех $x \in X \cap O(x_*, \alpha)$, где $O(x_*, \alpha)$={t$\in R^1 | \, |t - x_*| \leq \alpha\}$.

При этом значение $f(x_*)$ называется *локальным минимумом* функции $f(x)$.

Введем обозначение

$f_{min} = f(x_*) = \underset{x \in X}{locmin} f(x)$.

Аналогично вводится понятие *локального максимума* функции $f(x)$, который обозначим

$f_{max} = f(x_*) = \underset{x \in X}{locmax} f(x)$.

Локальный минимум и локальный максимум функции объединяются общим названием *локальный экстремум* функции $f(x)$, а те значения аргумента, при которых достигаются экстремумы функции, называются *точками локального экстремума* функции. Необходимые условия локального экстремума функции формулируются следующей *теоремой (теорема Ферма):*

Если функция $f(x)$ определена и дифференцируема на множестве X и во внутренней точке x_* этого множества имеет локальный экстремум, то в этой точке ее производная равна нулю: $f'(x_*) = 0$.

Точки, в которых первая производная $f'(x)$ от функции $y = f(x)$ равна нулю, называются *критическими*.

Теорема Ферма выражает лишь необходимые, но не достаточные условия экстремума функции.

Достаточные условия экстремума функции могут быть сформулированы в виде теоремы:

Пусть функция $f(x)$ дважды дифференцируема в точке x_* и $f'(x_*) = 0$, $f''(x_*) \neq 0$. Тогда в точке x_* функция имеет локальный экстремум, причем точка x_* будет точкой локального минимума, если $f''(x_*) > 0$ и максимума, если $f''(x_*) < 0$.

5.2. Наибольшее и наименьшее значения функции на отрезке

Функция, имеющая непрерывную производную на всём множестве определения, называется *гладкой* или *непрерывно дифференцируемой функцией*.

Кусочно-заданной называется функция, которая на разных промежутках области ее определения задана разными формулами.

Пусть заданы $x_1, x_2, ..., x_n$ $(a < x_1 < x_2 < ... < x_n < b)$ – точки смены формул. Как и все *кусочно-заданные* функции, *кусочно-гладкую* функцию можно записывать на каждом из интервалов $(a; x_1), (x_1; x_2), \ldots, (x_n; b)$ отдельной формулой:

$$f(x)=\begin{cases} f_0(x), & a<x<x_1 \\ f_1(x), & x_1<x<x_2 \\ & \dots \\ f_n(x), & x_n<x<b \end{cases}$$

Здесь $f_i(x)$ – гладкие функции. Если к тому же выполнены *условия согласования*
$f_{i-1}(x_i)=f_i(x_i)=f(x_i)$ при $i=1,2,3,\dots,n$,
то кусочно-гладкая функция будет непрерывной.

Пусть $y=f(x)$ непрерывная на отрезке $[a,b]$ функция. Тогда по теореме Вейерштрасса на этом отрезке функция достигает своих наибольшего и наименьшего значений. Если наибольшее значение функции достигается внутри отрезка $[a,b]$, то очевидно, что это значение будет наибольшим из максимумов (если имеется несколько локальных максимумов). Но может случиться, что наибольшее значение будет достигаться на одном из концов отрезка.

Таким образом, функция на отрезке $[a,b]$ достигает своего наибольшего значения либо на одном из концов этого отрезка, либо в такой внутренней точке этого отрезка, которая является точкой локального максимума.

То же самое можно сказать и о наименьшем значении функции.

Из изложенного выше следует *правило*: если требуется найти наибольшее и наименьшее значения кусочно-гладкой непрерывной функции на отрезке $[a,b]$, то надо:

1) найти все критические точки исследуемой функции;

2) вычислить значения функции в критических точках, граничных точках a и b отрезка $[a,b]$ и точках, в которых производная функции не существует;

3) выбрать из этих значений функции наименьшее и наибольшее.

*Пример 5.1.*Найти наибольшее и наименьшее значения

функции
$f(x) = |x^2 - x - 3|$, заданной на отрезке $X = [-2,2]$.
Аналитическое решение. Определим точки, в которых производная заданной функции не существует:
$x^2 - x - 3 = 0 \Rightarrow x_{1,2} = \frac{1 \mp \sqrt{13}}{2} \Rightarrow x_1 = \frac{1-\sqrt{13}}{2} \approx -1.30278, x_2 = \frac{1+\sqrt{13}}{2} \approx$ 2.30278.

Заметим, что $x_2 \notin [-2, 2]$. Далее, следуя сформулированному выше правилу, находим

1) $(x^2 - x - 3)' = 0 \Rightarrow 2x - 1 = 0 \Rightarrow x_0 = \frac{1}{2}$;
2) $f(-2) = 3\,, f\left(\frac{1-\sqrt{13}}{2}\right) = 0\,, f\left(\frac{1}{2}\right) = 3.25\,,\ f(2) = 1\,;$
3) $x_{min} = \frac{1-\sqrt{13}}{2} \approx -1.30278, f_{min} = 0\,;\ x_{max} = \frac{1}{2}\,, f_{max} = 3.25\,.$

Приведем подтверждающий полученные результаты график исследуемой функции, построенный с помощью системы Mathematica(см. рис. 7), и аналитическое решение этой задачи в системе Mathematica.

```
Plot[Abs[x^2 - x - 3], {x, -2, 2}]
```

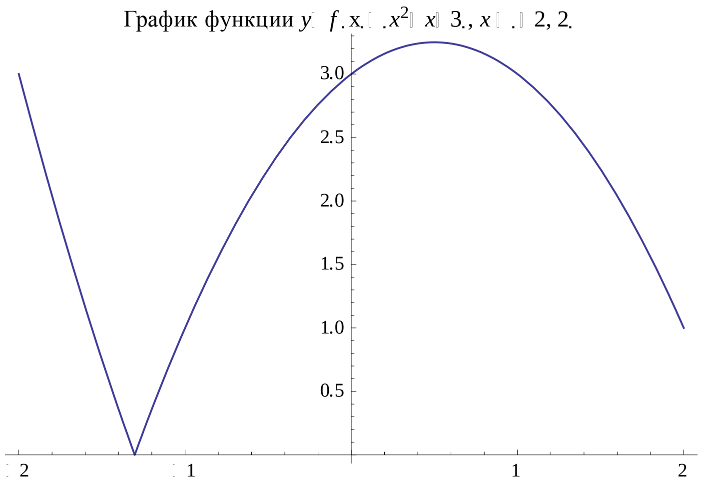


Рис. 7. График функции $y = f(x) = |x^2 - x - 3|$, построенный в Mathematica

Аналитическое решение, реализованное в СКМ Mathematica.

```
a = -2; b = 2; f[x_] = Abs[x^2 - x - 3];

s1 = Solve[x^2 - x - 3 == 0 && x ≥ a && x ≤ b, x]
```

$\left\{\left\{x \to \frac{1}{2}\left(1-\sqrt{13}\right)\right\}\right\}$

```
x1 = x /. First[First[s1]] // N

-1.30278

f1 = ∂x (x^2 - x - 3); sd = Solve[f1 == 0, x]; x2 = x /. First[First[sd]]
```

$\frac{1}{2}$

```
fk1 = f[a]

3

fk2 = f[x2] // N

3.25

fk3 = f[x1] // Chop

0

fk4 = f[b]

1

fmin = Min[fk1, fk2, fk3, fk4]

0

fmax = Max[fk1, fk2, fk3, fk4]

3.25
```

5.3. Безусловный экстремум функций нескольких переменных

Понятие экстремума и локального экстремума функции одной переменной можно обобщить на случай функции двух и более переменных. При этом для функции нескольких переменных следует различать понятия условного и безусловного экстремума функции. Первое из них предполагает наличие дополнительных соотношений, связывающих между собой значения аргументов исследуемой функции. Эта задача является существенно более сложной, чем задача отыскания безусловного экстремума функции нескольких переменных.

Определим понятие безусловного локального минимума (максимума) функции нескольких переменных.

Определение 1.

Точка $u_* \in R^n$ называется точкой безусловного локального минимума (максимума) функции $f(u)$, если существует такое α > 0, что неравенство $f(u_*) \leq f(u)$ $(f(u_*) \geq f(u))$выполняется для всех $u\in$ $O(u_*, \alpha)$, где

$O(u_*, \alpha)$=$\{u \in R^n |\ \|u - u_*\| < \alpha\}$.

Необходимые и достаточные условия безусловного экстремума функции

Необходимые условия безусловного локального экстремума, как и в случае функции одной переменной, выражает теорема Ферма.

Теорема 1 (Необходимые условия)

Пусть дана дифференцируемая функция $f(u)$, для которой точка $u_* \in R^n$ является локального экстремума. Тогда выполняется равенство

$$\frac{\partial f}{\partial u}(u_*) = 0.$$

Это условие эквивалентно n скалярным равенствам:

$$\frac{\partial f}{\partial u^1}(u_*) = 0,\ \cdots \frac{\partial f}{\partial u^n}(u_*) = 0. \qquad (5.2)$$

Число уравнений в системе (5.2) совпадает с числом неизвестных $u_* = \begin{pmatrix} u_*^1 \\ \cdots \\ u_*^n \end{pmatrix}$, поэтому необходимые условия теоремы Ферма являются *эффективными*, т.е. позволяют в принципе определить эти неизвестные. Теорема Ферма выражает лишь необходимые условия экстремума. Следовательно, среди точек

$u_0 \in R^n$, удовлетворяющих равенству $\frac{\partial f}{\partial u}(u_0) = 0$, могут оказаться и такие, которые не доставляют локальный экстремум функции $f(u)$. Для выявления точек локальных экстремумов функции необходимо провести дополнительные исследования, связанные с привлечением достаточных условий.

Определение 2.

Точка $u_0 = \begin{pmatrix} u_0^1 \\ \cdots \\ u_0^n \end{pmatrix} \in R^n$, удовлетворяющая условиям (5.2), называется критической точкой $f(u)$.

Теорема 2 (Достаточные условия)

Пусть $u_0 = \begin{pmatrix} u_0^1 \\ \cdots \\ u_0^n \end{pmatrix} \in R^n$ – критическая точка дважды дифференцируемой функции $f(u)$. Тогда для того чтобы в этой точке достигался локальный минимум (максимум) функции, достаточно, чтобы главные миноры матрицы

$$\frac{\partial^2 f}{\partial u \partial u}(u_0) = \begin{pmatrix} \frac{\partial^2 f}{\partial u^1 \partial u^1}(u_0) & \cdots & \frac{\partial^2 f}{\partial u^1 \partial u^n}(u_0) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial u^n \partial u^1}(u_0) & \cdots & \frac{\partial^2 f}{\partial u^n \partial u^n}(u_0) \end{pmatrix}$$

были строго положительны (знакочередующиеся).

Для случая n=2 достаточные условия локального экстремума имеют следующий вид.

Обозначим

$A_0 = \frac{\partial^2 f}{\partial u^1 \partial u^1}(u_0^1, u_0^2)$, $B_0 = \frac{\partial^2 f}{\partial u^1 \partial u^2}(u_0^1, u_0^2) = \frac{\partial^2 f}{\partial u^2 \partial u^1}(u_0^1, u_0^2)$, $C_0 = \frac{\partial^2 f}{\partial u^2 \partial u^2}(u_0^1, u_0^2)$,

$\Delta_0 = \begin{vmatrix} A_0 & B_0 \\ B_0 & C_0 \end{vmatrix}$.

Тогда, если:

1) $\Delta_0 > 0$, то в точке u_0 имеет место локальный экстремум, и

$\begin{cases} A_0 < 0 \Rightarrow f(u_0^1, u_0^2) - loc\max, \\ A_0 > 0 \Rightarrow f(u_0^1, u_0^2) - loc\min; \end{cases}$

2) $\Delta_0 < 0$, то в точке u_0 локального экстремума нет.

Пример 5.2. Найти локальный экстремум функции

$z = f(x, y) = -x^2 - x\,y - y^2 + 8x + 7y - 3.$

Решение.

Необходимые условия.

$$\begin{cases} \dfrac{\partial z}{\partial x} = -2x - y + 8 = 0, \\ \dfrac{\partial z}{\partial y} = -x - 2y + 7 = 0. \end{cases}$$

Решением этой системы будут числа $x_0 = 3,\ y_0 = 2$.

Достаточные условия.

$A_0 = \frac{\partial^2 f}{\partial x^2}(x_0, y_0) = -2$, $B_0 = \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) = -1$,

$C_0 = \frac{\partial^2 f}{\partial y^2}(x_0, y_0) = -2$,

$\Delta_0 = \begin{vmatrix} -2 & -1 \\ -1 & -2 \end{vmatrix} = 4 - 1 = 3 > 0$, $A_0 = -2 < 0 \Rightarrow$ случай локального максимума.

Вычислим максимальное значение исследуемой функции:

$f_{max} = f(x_0, y_0) = -x_0{}^2 - x_0 y_0 - y_0{}^2 + 8x_0 + 7y_0 - 3 = 16.$

Компьютерная реализация решения этого примера:

```
z = -x^2 - x*y - y^2 + 8 x + 7 y - 3;

FindMaximum[z, {x, 3}, {y, 1}]

{16., {x → 3., y → 2.}}
```

5.4. Условный экстремум функций нескольких переменных

5.4.1. Условный экстремум функции нескольких переменных с ограничениями типа равенства

В задаче поиска экстремума функции нескольких переменных дополнительно потребуем, чтобы ее переменные $u = \begin{pmatrix} u^1 \\ \cdots \\ u^n \end{pmatrix}$ удовлетворяли условиям

$g_1(u) = 0, \cdots , g_m(u) = 0,\ g_i\colon R^n \to R^1,\ i = 1,2, \cdots,m,$ (5.3)

которые при этом называются *ограничениями*, а функция $f(u)$ – целевой функцией. Математическая формулировка задачи оптимизации имеет вид

Задача 1.

$f(u) \to min(max),\ u \in R^n, g_1(u) = 0, \cdots , g_m(u) = 0.$

Поставленная так задача называется *задачей на условный экстремум с ограничениями типа равенств*. При решении задачи 1 обычно предметом поиска служат точки локального экстремума.

Определение 3.

Точка $u_* \in R^n$ называется точкой условного (с ограничениями типа равенств) локального минимума (максимума) функции $f(u)$, если существует такое $\alpha > 0$, что неравенство $f(u_*) \le f(u)$ $(f(u_*) \ge f(u))$ выполняется для всех точек $u \in O(u_*, \alpha)$, удовлетворяющих равенствам (5.3).

Важным инструментом исследования задач на условный экстремум является функция Лагранжа.

Определение 4.

Функция $L\colon R^{n+m+1} \to R^1$, определенная равенством
$L(u,\lambda) = \lambda_0 f(u) + \lambda_1 g_1(u) + \cdots + \lambda_m g_m(u) =$
$\lambda_0 f(u) + \sum_{i=1}^{m} \lambda_i g_i(u)$, $\lambda = \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \cdots \\ \lambda_m \end{pmatrix} \in R^{m+1}$ называется функцией Ла-

гранжа для *задачи 1*. Переменные $\lambda_0, \lambda_1, \cdots, \lambda_m$ называются *множителями Лагранжа*.

В предположении, что функции $f, g_1, \cdots, g_m$ дифференцируемы в точке локального экстремума $u_* = \begin{pmatrix} u_*^1 \\ \cdots \\ u_*^n \end{pmatrix}$, приведем условия, которым эта точка должна удовлетворять.

Теорема 3. Пусть точка $u_* = \begin{pmatrix} u_*^1 \\ \cdots \\ u_*^n \end{pmatrix}$ является точкой локального экстремума функции f с ограничениями (1). Тогда существует вектор $\lambda^* = \begin{pmatrix} \lambda_0^* \\ \lambda_1^* \\ \cdots \\ \lambda_m^* \end{pmatrix} \neq 0$, такой, что $\lambda_0^* \in \{0,1\}, \frac{\partial L}{\partial u}(u_*, \lambda^*) = 0$.

Заметим, что равенство $\frac{\partial L}{\partial u}(u_*, \lambda^*) = 0$ эквивалентно n скалярным равенствам

$$\frac{\partial L}{\partial u^1}(u_*, \lambda^*) = 0, \cdots, \frac{\partial L}{\partial u^n}(u_*, \lambda^*) = 0. \qquad (5.4)$$

Из анализа включения $\lambda_0^* \in \{0,1\}$ устанавливается конкретное значение величины λ_0^*. После этого число неизвестных в условиях (5.4) становится равным $n+m$. Оно совпадает с числом объединенных в систему уравнений в условиях (5.3) и (5.4). Таким образом, необходимые условия экстремума, приведенные в теореме являются эффективными.

Среди точек u_0, удовлетворяющих условиям (5.3) и (5.4), могут оказаться и такие, которые не доставляют локальный экстремум функции f. Для выявления локальных экстремумов функции следует провести дополнительные исследования, связанные с привлечением достаточных условий.

Определение 5.

Пара $(u_0, \lambda^0) \in R^{n+m+1}$, удовлетворяющая условиям (5.3), (5.4), называется *критической парой для задачи 1.*

Для случая $n = 2$, $m = 1$ достаточные условия локального экстремума принимают следующий вид. Пусть $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} 1 \\ \lambda_1^0 \end{pmatrix}$ – стационарная пара.

Положим

$$\Delta_0 = \begin{vmatrix} 0 & \frac{\partial g}{\partial x}(x_0, y_0) & \frac{\partial g}{\partial y}(x_0, y_0) \\ \frac{\partial g}{\partial x}(x_0, y_0) & \frac{\partial^2 L}{\partial x^2}(x_0, y_0, 1, \lambda_1^0) & \frac{\partial^2 L}{\partial x \partial y}(x_0, y_0, 1, \lambda_1^0) \\ \frac{\partial g}{\partial y}(x_0, y_0) & \frac{\partial^2 L}{\partial y \partial x}(x_0, y_0, 1, \lambda_1^0) & \frac{\partial^2 L}{\partial y^2}(x_0, y_0, 1, \lambda_1^0) \end{vmatrix} \quad (5.5)$$

Из неравенства $\Delta_0 > 0$ следует, что точка $u_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ является точкой условного локального максимума, а из неравенства $\Delta_0 < 0$ –точкой условного локального минимума.

Пример 5.3. Найти экстремум функции

$f(x, y) = xy$

при условии, что

$g(x, y) = x^2 + y^2 - 2 = 0.$

Решение.

Запишем функцию Лагранжа для задачи

$L(x, y, \lambda_0, \lambda_1) = \lambda_0 xy + \lambda_1 (x^2 + y^2 - 2)$

и необходимые условия экстремума

$$x^2 + y^2 = 2, \qquad \frac{\partial L}{\partial x}(x, y, \lambda_0, \lambda_1) = \lambda_0 y + 2\lambda_1 x = 0,$$

$$\frac{\partial L}{\partial y}(x, y, \lambda_0, \lambda_1) = \lambda_0\, x + 2\,\lambda_1 y = 0, \lambda_0 \in \{0,1\},\ {\lambda_0}^2 + {\lambda_1}^2 > 0. \qquad (5.6)$$

Пусть $\lambda_0 = 0$. Тогда условия (5.6) принимают вид

$$x^2 + y^2 = 2, \qquad 2\lambda_1\, x = 0,\ 2\lambda_1\, y = 0 \ \Rightarrow x = y = 0 \ \Rightarrow 0 = 2.$$

Получили противоречие. Тогда принимаем, что $\lambda_0 = 1$. Перепишем уравнение (5.4) с учетом последнего равенства в виде

$$x^2 + y^2 = 2,\ y + 2\lambda_1 x = 0,\ x + 2\,\lambda_1\, y = 0. \qquad (5.7)$$

Решая систему (7), находим

$$y = -2\lambda_1 x, \qquad x = -2\lambda_1 y \Rightarrow 4{\lambda_1}^2 x^2 + 4{\lambda_1}^2 y^2 = 2 \Rightarrow 4{\lambda_1}^2 (x^2 + y^2)$$
$$= 2 \ \Rightarrow {\lambda_1}^2 = \frac{1}{4}.$$

Отсюда получаем следующие стационарные наборы при $\lambda_0 = 1$.

1) ${\lambda_1}^{(1)} = -\frac{1}{2}$, $\begin{cases} x_1 = y_1 = \ \ 1 \\ x_2 = y_2 = -1 \end{cases}$, 2)${\lambda_1}^{(2)} = \frac{1}{2}$, $\begin{cases} x_3 = -y_3 = \ \ 1 \\ x_4 = -y_4 = -1 \end{cases}$.

Достаточные условия экстремума.

$$\frac{\partial g}{\partial x} = y, \qquad \frac{\partial g}{\partial y} = x, \qquad \frac{\partial^2 L}{\partial x^2} = 2\lambda_1, \qquad \frac{\partial^2 L}{\partial x \partial y} = \frac{\partial^2 L}{\partial y \partial x} = \lambda_0, \frac{\partial^2 L}{\partial y^2} = 2\lambda_1 \ .$$

Строим определитель Δ:

$$\Delta = \begin{vmatrix} 0 & y & x \\ y & 2\lambda_1 & \lambda_0 \\ x & \lambda_0 & 2\lambda_1 \end{vmatrix} \Rightarrow \Delta_1 = \begin{vmatrix} 0 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{vmatrix} = 4, \ \Delta_2$$
$$= \begin{vmatrix} 0 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \end{vmatrix} = 4,$$

$$\Delta_3 = \begin{vmatrix} 0 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} = -4, \ \ \Delta_4 = \begin{vmatrix} 0 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{vmatrix} = -4.$$

По теореме 2 заключаем, что точки $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, \begin{pmatrix} x_4 \\ y_4 \end{pmatrix}$ являются точками локального минимума, а точки $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ – локального максимума. При этом $f_{min} = -1, \ f_{max} = 1$.

Компьютерная реализация решения задачи в примере 5.3 приведена ниже.

Задача. Найти экстремум функции при условии типа равенство.

```
z = f[x, y] = x * y
```

$x\,y$

```
g = x^2 + y^2 - 2 == 0
```

$-2 + x^2 + y^2 == 0$

```
sy = Solve[g, y]
```

$\left\{\left\{y \to -\sqrt{2 - x^2}\right\}, \left\{y \to \sqrt{2 - x^2}\right\}\right\}$

```
y1 = y /. First[First[sy]]
```

$-\sqrt{2 - x^2}$

```
y2 = y /. First[Last[sy]]
```

$\sqrt{2 - x^2}$

```
h1 = x * y1
```

$-x\sqrt{2 - x^2}$

```
h2 = x * y2
```

$x\sqrt{2 - x^2}$

```
fmax1 = FindMaximum[h1, {x, -0.8}]
```

$\{1., \{x \to -1.\}\}$

```
fmin1 = FindMinimum[h1, {x, +0.6}]
```

$\{-1., \{x \to 1.\}\}$

```
fmax2 = FindMaximum[h2, {x, +0.7}]
```

$\{1., \{x \to 1.\}\}$

```
fmin2 = FindMinimum[h2, {x, -0.5}]
```

$\{-1., \{x \to -1.\}\}$

5.4.2. Изопериметрическая задача

Классическая изопериметрическая задача ставится следующим образом.

Задача.

Среди всех плоских фигур заданного вида и одинакового периметра найти ту, которая имеет максимальную площадь.

Фигурами заданного вида могут быть, например, треугольники, прямоугольники, n–угольники, произвольные фигуры и др. Решением изопериметрической задачи в классе произвольных плоских замкнутых фигур является *круг*.

Определение 6. Плоский n–угольник, имеющий наибольшую площадь среди всех изопериметрических с ним n–угольников, называется *максимальным* n–угольником.

Теорема 4 (Зенодора). Максимальный n–угольник должен быть правильным, т.е.:

1) равносторонним; 2) равноугольным; 3) выпуклым.

Пример 5.4. Решеткой длиной 120 метров нужно огородить прямоугольную площадку наибольшей площади. Определить размеры прямоугольной площадки.

Решение.

Пусть x, y – стороны искомого прямоугольника. Тогда

$S = xy \to max,$

при условии, что

$2(x + y) = 120.$

Выпишем функцию Лагранжа

$L(x, y, \lambda_0, \lambda_1) = \lambda_0 xy + \lambda_1(x + y - 60)$

и необходимые условия экстремума для данной задачи

$x + y = 60,$

$\frac{\partial L}{\partial x}(x, y, \lambda_0, \lambda_1) = \lambda_0 y + 2\lambda_1 = 0,\ \frac{\partial L}{\partial y}(x, y, \lambda_0, \lambda_1) = \lambda_0\, x + 2\,\lambda_1 = 0.$

Предположение ${\lambda_0}^0 = 0$ противоречит условию $\lambda^0 = \begin{pmatrix} \lambda_0^0 \\ \lambda_1^0 \end{pmatrix} \neq 0.$

Тогда $\lambda_0^0 = 1$, и соотношения (1) принимают вид:

$$x + y = 60, \qquad y + 2\lambda_1 = 0, \qquad x + 2\lambda_1 = 0.$$

Решением полученной системы линейных алгебраических уравнений являются $y = -2\lambda_1,\ x = -2\lambda_1 \Rightarrow -4\lambda_1 = 60 \Rightarrow \lambda_1 = -15 \Rightarrow x_0 = y_0 = 30$. Заметим, что здесь

$$\Delta = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = 2 > 0.$$

Точка $u_0 = \begin{pmatrix} 30 \\ 30 \end{pmatrix}$ действительно доставляет условный максимум функции S.

Данный результат подтверждает справедливость теоремы Зенодора, поскольку квадрат является правильным четырехугольником. Среди треугольников максимальным будет равносторонний треугольник, среди пятиугольников – правильный пятиугольник, и т.д. С ростом числа сторон в пределе получим круг. Таким образом, среди изопериметрических фигур круг имеет максимальную площадь.

5.5. Линейное программирование

Задачи *управления и планирования* обычно сводятся к выбору некоторой системы параметров и системы функций, которые математически приводят к экстремальным задачам следующего вида: требуется найти переменные задачи $x_1, x_2, \ldots, x_n$, их обычно записывают в виде вектора $X = (x_1, x_2, \ldots, x_n)$, которые обеспечивают *экстремум целевой функции*

$$Z(X) = f(x_1, x_2, \ldots, x_n) \to max(min) \tag{5.8}$$

и удовлетворяют системе *ограничений*

$$\begin{cases} \varphi_i(x_1, x_2, \ldots, x_n) = 0, i = 1, 2, \ldots, l; \\ \varphi_i(x_1, x_2, \ldots, x_n) \le 0 (\ge 0),\ \ i = l + 1, l + 2, \ldots, m. \end{cases} \tag{5.9}$$

Это *общая задача математического программирования*. В зависимости от вида целевой функции (5.8) и системы ограничений (5.9) математическое программирование делится на *линейное* и *нелинейное*.

Под *линейным программированием* понимается раздел теории экстремальных задач, в котором изучаются задачи минимизации (или максимизации)*линейных функций*, на множествах (ограничениях), задаваемых системами *линейных равенств и неравенств*.

Допустимым решением (*планом*) задачи линейного программирования называется любой n-мерный вектор X=(x_1, x_2, ..., x_n), удовлетворяющий системе ограничений и условиям неотрицательности ($x_j \geq 0, j = 1, 2, \ldots, n$). Множество допустимых решений образует область допустимых решений (ОДР).

Оптимальным решением (планом) задачи линейного программирования называется такое допустимое решение (план) задачи, при котором целевая функция достигает экстремума.

Рождение линейного программирования связано с работой нашего отечественного математика Л.В. Канторовича, написанной им в 1939 году и посвященной оптимизации раскроя листов фанеры для самолетов. В этой же работе было показано, что многие задачи экономики формулируются как задачи об экстремуме линейной функции при линейных ограничениях. В 1947 году появились работы, посвященные разработке той же тематики, американского экономиста Т. Купманса [Купманс, 1957]. Именно он ввел термин «линейное программирование». Премия памяти Нобеля за 1975 г. в области экономики была присуждена Т. Купмансу совместно с Л.В. Канторовичем [Канторович, 1939] «за вклад в теорию оптимального распределения ресурсов».

5.5.1. Задачи линейного программирования

Математической моделью экономической задачи называется совокупность математических соотношений, описывающих рассматриваемый экономический процесс. Для составления математической модели необходимо: 1) выбрать переменные задачи; 2) составить систему ограничений; 3) задать целевую функцию.

Рассмотрим задачи, математические модели которых являются задачами линейного программирования.

Задача использования ресурсов. При производстве n видов продукции используются m видов ресурсов (сырья, энергии, комплектующих). Известны: запасы ресурсов: $в_1$, $в_2$, …, $в_m$; *расход* каждого i-го вида ресурса на изготовление единицы j-й продукции, который будем обозначать a_{ij} (i=1,2, ...,m; j=1,2,...,n); c_j–прибыль, получаемая при реализации единицы j-й продукции (j=1,2,...,n). Составить план выпуска продукции, обеспечивающий *максимальную* прибыль.

Математическая модель. Пусть $X=(x_1, x_2,\dots, x_n)$ – искомый вектор задачи, где x_j (j=1,2,…,n) – объем выпуска j–той продукции. Тогда c_jx_j –прибыль от реализации всего объема j-той продукции, $a_{ij}x_j$ – затраты i-го вида ресурса на весь объем выпускаj-той продукции. Так как общее количество затраченных i-тых ресурсов на производство всех видов продукции равно $a_{i1}x_1+a_{i2}x_2+\dots+a_{in}x_n$ и оно не может превосходить их запаса $в_i$, то должно выполняться неравенство

$$a_{i1}x_1+a_{i2}x_2+\dots+a_{in}x_n \le в_i \; (i=1,2,\dots,m). \qquad (*)$$

Кроме того, необходимо учитывать неотрицательность величин x_j, так как объем выпуска продукции не может быть отрицательным. Таким образом, для обеспечения максимальной прибыли объема продукции каждого вида, т.е. величиныx_1, x_2, …,x_n, должны быть таковы, чтобы функция $Z(X) = c_1x_1+c_2x_2+\dots+c_nx_n$ принимала наибольшее значение при неотрицательных x_j и при условии выполнения соотношений (*) для любого i=1,2,…,m. Следовательно, математическая модель имеет вид:

$$Z(X) = c_1x_1+c_2x_2+\dots+c_nx_n \to \max,$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \le b_2; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..; \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \le b_m; \\ x_j \ge 0, j = 1,2,\dots,n \;. \end{cases}$$

Задача о составлении рациона питания. Животные должны получать ежедневно m питательных веществ в количестве не менее b_1, b_2, …, b_m. В рацион животных входят корма n видов. Известно: a_{ij} (i=1,2,…,m; j=1,2,…,n) - содержание i-го питательного вещества в единице j-го вида корма; c_j (j=1,2,…,n) – стоимость единицы j-го вида корма. Составить суточный рацион кормления животных, обеспечивающий минимальные затраты.

Математическая модель. Введем переменные задачи $X = (x_1, x_2,\ldots, x_n)$, где x_j (j=1,2,…,n) – объем j - го вида корма, входящего в суточный рацион. Так как $a_{ij}x_j$ – количество i -го питательного вещества, содержащегося в j -ом виде корма, входящего в суточный рацион, c_jx_j – стоимость j - го корма, то математическая модель имеет вид:

$Z(X) = c_1x_1+c_2x_2+\ldots+c_nx_n \rightarrow \min$,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geq b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geq b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..; \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m; \\ x_j \geq 0, j = 1,2,\ldots,n\ . \end{cases}$$

5.5.2. Приведение общей задачи линейного программирования к канонической форме

Каноническая задача линейного программирования имеет вид:

$$Z(X)=c_1x_1+c_2x_2+\ldots+c_nx_n \rightarrow \min (\max), \qquad (5.10)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..; \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m; \\ x_j \geq 0, j = 1,2,\ldots,n\ . \end{cases} \qquad (5.11)$$

Она отличается от других задач тем, что ее система ограничений является системой уравнений и все переменные неотрицательные.

При необходимости перехода от неравенства к уравнению вводят дополнительные переменные. Неравенство $a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n \leq b_i$ заменяется уравнением $a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n+x_{n+1}= b_i$ и условием неотрицательности дополнительной переменной $x_{n+1} \geq 0$, а неравенство $a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n \geq b_i$ – уравнением $a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n$-

$x_{n+1} = b_i$ и условием неотрицательности $x_{n+1} \geq 0$. Дополнительные переменные вводят в целевую функцию с коэффициентом, равным нулю.

Любая переменная x_j, на которую не наложено условие неотрицательности, заменяется разностью двух других неотрицательных переменных $x_j = x_j' - x_j''$, $x_j' \geq 0$, $x_j'' \geq 0$.

В канонической задаче целевая функция может, как минимизироваться, так и максимизироваться. Для того, чтобы перейти от нахождения min к нахождению max или наоборот, достаточно изменить знаки коэффициентов целевой функции. Полученная в результате этого задача и исходная задача имеют одно и то же оптимальное решение, а значения целевых функций на этом решении отличаются только знаком.

Пример. Привести к каноническому виду задачу линейного программирования

$Z(X) = 3x_1 + x_2 + x_3 \to \min$

$$\begin{cases} x_1 + 2x_2 + x_3 = 7; & \\ x_1 - x_2 + 2x_3 \leq 1; & \quad +x_4 \\ 2x_1 + x_2 - 2x_3 \geq 5; & \quad -x_5 \\ x_2 \geq 0,\ x_3 \geq 0. & \end{cases}$$

в целях превращения в уравнения.

Решение. Перейдем к задаче на отыскание max целевой функции. Для этого изменим знаки коэффициентов целевой функции. Переменную x_1, на которую не наложено условие неотрицательности, заменим разностью

$$x_1 = x_1' - x_1'',\ x_1' \geq 0,\ x_1'' \geq 0.$$

Записываем задачу в каноническом виде:

$Z(X) = -3x_1' + 3x_1'' - x_2 - x_3 + 0 \cdot x_4 + 0 \cdot x_5 \to \max$

$$\begin{cases} x_1' - x_1'' + 2x_2 + x_3 = 7; \\ x_1' - x_1'' - x_2 + 2x_3 + x_4 = 1; \\ 2x_1' - 2x_1'' + x_2 - 2x_3 - x_5 = 5; \\ x_1' \geq 0, x_2' \geq 0, x_j \geq 0, j = 2,3,4,5. \end{cases}$$

Замечание. Матрица A, составленная из коэффициентов системы (5.11), называется *матрицей условий канонической задачи линейного программирования*. Будем считать, что rang $A = m$. Значит $n \geq m$.

Если это не так, т.е. rang $A < m$, то находим ранг расширенной матрицы B, составленной из коэффициентов и свободных членов системы уравнений (5.11), который обозначим rang B.

Если окажется rang $B >$ rang A, то система (5.11) несовместна, и рассматриваемая каноническая задача линейного программирования решения не имеет. Если же rang B = rang A = $m_1 < m$, то в системе (5.11) оставим только те m_1 соотношений, которые определяют ее ранг, а остальные соотношения опускаем, понимая, что они являются следствием оставшихся. Во всех дальнейших рассуждениях надо под числом m подразумевать число m_1.

Если $n = m$, то (5.11) имеет единственное решение, которое и будет решением рассматриваемой задачи линейного программирования. Поэтому в дальнейшем будем считать, что параметры удовлетворяют соотношениям $n > m$ = rang A.

5.5.3. Симметрическая или стандартная форма задач линейного программирования

При решении некоторых задач возникает необходимость перехода от канонической задачи к *симметричной*, которая в матричной записи имеет вид:

$$\begin{cases} Z(X) = CX \to max; \\ AX \le A_0, X \ge 0 \end{cases}$$

или

$$\begin{cases} Z(X) = CX \to min; \\ AX \ge A_0, X \ge 0, \end{cases}$$

где $C=(c_1, c_2, \ldots, c_n)$; $A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$; $X=\begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ x_n \end{pmatrix}$; $A_0=\begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ . \\ b_m \end{pmatrix}$.

5.5.4. Применение графического метода

в задаче линейного программирования

Графический метод решения задач линейного программирования эффективен лишь в случае, когда разность между числом переменных n и числом условий на эти переменные m не превышает 2. Если же это условие не выполняется, используется общий алгоритм решения задачи линейного программирования [Лутманов, 2007].

В *экономико-производственной* задаче (6) [Аксенов, 2016]:

$Z(X) = 60x_1 + 140x_3 \to max,$

$$\begin{cases} -x_1 - 7x_3 \leq -8; \\ 5x_1 + 7x_3 \leq 30; \\ x_1 \geq 0, x_3 \geq 0, \end{cases} \qquad (5.12)$$

возможна следующая графическая иллюстрация решения: изобразив на плоскости x_1, x_3 область, удовлетворяющую условиям (5.12), получим треугольник ABC с вершинами:

$A(0;\frac{30}{7})$; $B(\frac{11}{2};\frac{5}{14})$; $C(0;\frac{8}{7})$.

Если на той же плоскости изобразить множество точек, где оптимизируемая функция $Z(X)$ принимает постоянное значение p, то получится прямая, уравнение которой $60x_1+140x_3=p$. В зависимости от величины p эта прямая, нормальный вектор которой $(60;140)\|(3;7)$, перемещается перпендикулярно ему.

На рис.8 эта прямая обозначена как прямая EE. Поскольку нормальный вектор этой прямой есть вектор $\vec{n}=\{3;7\}$, а нормальные векторы прямых AB и CB будут, соответственно, $\vec{n}_1=\{5;7\}$и $\vec{n}_2=\{1;7\}$, то очевидно, что наибольшее значение функция $Z(X)$ примет при прохождении EE через точку $A(0;\frac{30}{7})$. Таким образом, наибольшим значением этой функции будет число $M = \max Z(X) = 600\times0+140\times\frac{30}{7} = 600$.

При этом оптимизация процесса утилизации отходов будет обеспечена при отсутствии реагента P_1 (x_1=0), при наличии $\frac{30}{7}$ условных единиц реагента P_3 ($x_3=\frac{30}{7}$) и $\frac{50}{7}$ условных единиц реагента P_2 ($x_2 = \frac{50}{7}$) (выполнение равенства $2x_1+x_2+3x_3=20$ при $x_1=0$, $x_2 = \frac{50}{7}$ и $x_3=\frac{30}{7}$).

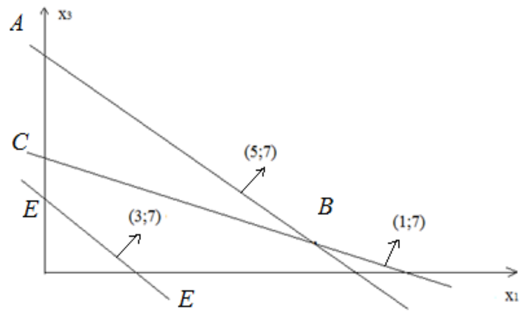


Рис.8. Графическая иллюстрация решения задачи (6).

Графическим методом можно решать любые задачи линейного программирования, у которых соотношение между параметрами n, m удовлетворяет неравенству $n–m \leq 2$. В этом случае вначале задача преобразовывается к такому виду, что на некоторой плоскости можно изобразить область возможного изменения параметров задачи. Затем проводятся операции, аналогичные тем, что были сделаны в рассмотренном примере.

Точный алгоритм решения канонической задачи линейного программирования графическим методом в этом случае таков:

– выбираем среди n переменных x_i, i=1,2,…,n две переменные (обозначим их z_1, z_2), которые будем называть *свободными* и через которые можно однозначно выразить остальные (n–2) переменных x_i, которые будем называть *базисными*. Очевидно, для этого достаточно, чтобы определитель, составленный из коэффициентов при выбранных базисных переменных в уравнениях, составляющих условия-ограничения решаемой задачи, был бы отличен от нуля;

– разрешаем относительно базисных переменных уравнения, составляющие условия – ограничения рассматриваемой задачи;

– подставляя полученные выражения для базисных переменных в оптимизируемую функцию и приведя подобные члены, найдем значения коэффициентов при свободных переменных. Пусть эти коэффициенты будут c_1^*, c_2^*;

– пользуясь неотрицательностью всех базисных переменных, заменяем каждое из полученных (n–2) уравнений, составляющих условия-ограничения решаемой задачи, неравенствами, связывающими свободные переменные;

– на плоскости двух свободных переменных (z_1, z_2) изображаем области допустимых решений, отвечающих полученным неравенствам. Пусть G – пересечение всех полученных областей;

– если G – пустое множество, то задача не имеет решения ввиду несовместимости ее системы ограничений;

– если G не является пустым множеством, строим нормаль *линии уровня* решаемой задачи и одну из линий уровня, имеющую общую точку с G.

Линией уровня решаемой задачи называется прямая, на которой целевая функция задачи принимает постоянное значение.

Если, например, свободными переменными задачи являются x_1, x_2 то линия уровня имеет вид $c_1^* \cdot x_1 + c_2^* \cdot x_2 = p$, где p = const. Все линии уровня параллельны между собой. Их нормаль $\overline{n}=\{c_1^*, c_2^*\}$.Перемещаем линию уровня в направлении её нормали до опорной прямой области G в задаче на максимум и в противоположном направлении – в задаче на минимум.

Опорной прямой области G называется такая линия уровня, которая имеет с G хотя бы одну общую точку и по отношению к которой сама G находится в одной из её полуплоскостей.

Если область G неограничена и при перемещении линии уровня в направлении, соответствующем приближению к экстремуму целевой функции, эта линия, неограниченно удаляясь от начала координат, имеет с областью G общие точки, рассматриваемая задача не имеет конечного решения ввиду не-ограниченности целевой функции.

Если рассматриваемая задача имеет оптимальное решение, то для его нахождения необходимо рассмотреть взаимное расположение опорной прямой области G и прямых, являющихся границами области G.

Возможны следующие случаи:

– если целевая функция задачи достигает экстремума в одной точке $M(z_1^*, z_2^*)$, то координаты этой точки являются оптимальными координатами тех исходных переменных рассматриваемой задачи, которые были выбраны свободными. Значения остальных переменных, обеспечивающих оптимум целевой функции, определяются по полученным ранее формулам, определяющим зависимость $(n–2)$ базисных координат через свободные;

– если целевая функция задачи достигает экстремума в каких-либо двух точках $\{P, Q\}$, то задача имеет бесконечное множество решений. Оптимальным решением в этом случае будет любая общая точка области G и прямой PQ. Сама оптимальная величина целевой функции равна ее значению в любой из этих точек.

Пример решения двумерной задачи линейного программирования в системе Mathematica приведен в главе 7

5.5.5. Целочисленные задачи линейного программирования

При рассмотрении целого ряда задач линейного программирования (в том числе транспортной задачи, многих задач финансового менеджмента и бизнеса, задач экономического планирования и т.д.) необходимо учитывать требования целочисленности используемых переменных. Рассмотрим, например, следующую задачу.

Задача. Из пункта А в пункт В ежедневно отправляются скорые и пассажирские поезда. Наличный парк вагонов разных типов, из которых ежедневно можно комплектовать данные поезда, и число пассажиров, вмещающихся в каждом из вагонов, приведены в таблице 2:

Пример комплектации поезда Таблица 2

Вагоны	Число вагоновв поезде		Число пассажиров	Парк вагонов
	скором	пассажирском		
плацкартный	5	8	58	92
купейный	6	4	40	80
мягкий	3	1	32	30

Определить количество скорых и пассажирских поездов, при которых число перевозимых пассажиров достигает максимума.

Вычислим сначала число пассажиров, перевозимых в одном скором ($N_с$) и в одном пассажирском ($N_п$) поезде. Из условий задачи находим:

$N_с$=5·58+6·40+3·32=626; $N_п$=8·58+4·40+1·32=656.

Если за n_1обозначить количество скорых поездов, заn_2 количество пассажирских поездов, которые можно сформировать из имеющегося парка вагонов, то максимальное число перевозимых пассажиров будет получено в результате решения следующей задачи линейного программирования:

$$\begin{cases} Z(\mathrm{X}) = 626n_1 + 656n_2 \to max; \\ 5n_1 + 8n_2 \le 92; \\ 6n_1 + 4n_2 \le 80; \\ 3n_1 + n_2 \le 30; \\ n_1 \ge 0, \; n_2 \ge 0. \end{cases}$$

Получилась обычная задача линейного программирования. Но из её постановки очевидно, что нас могут интересовать только такие решения, которые представляются натуральными числами. В данном случае мы таких чисел не получим. Возникает вопрос, как же найти целочисленные решения, оптимизирующие функционал в этой и аналогичной ей задачах.

В линейном программировании разработан метод решения таких задач. В честь итальянского математика, разработавшего этот метод, он называется *методом Гомори*.

Если число неизвестных решаемой задачи невелико, точнее, если решаемая задача допускает использование графического метода исследования, то можно предложить следующую процедуру нахождения оптимальных целочисленных решений таких задач:

Надо изобразить на плоскости фазовых переменных X область допустимых значений задачи и определить в этой области точки с целочисленными координатами. Затем вычислить значения оптимизируемой функции в этих точках и из этих значений выбрать оптимальное.

Так, например, для рассмотренной выше задачи, решенной графическим методом, множество допустимых значений переменных X является внутренностью треугольника ABC, изображенного на рис.8. Из этого рисунка легко найти, что этому треугольнику могут принадлежать лишь следующие точки с целочисленными координатами
$A_1(0;2)$, $A_2(0;3)$, $A_3(1;1)$, $A_4(1;2)$, $A_5(1;3)$, $A_6(2;2)$, $A_7(3;2)$.

Вычислив значения оптимизируемой функции в этих точках, находим, что наибольшее значение эта функция принимает в точке $A_5(1;3)$. Это значение равно 480.

Итак, величины (x_1=1, x_3=3) будут целочисленным решением задачи; при этих значениях оптимизируемая функция принимает значение 480.

Рассмотренная выше математическая модель *экономико-производственной задачи* на основании классификационной таблицы 1 может быть охарактеризована следующим образом:

1) по сфере применения эта модель прикладная;
2) по способу получения математической модели − теоретическая;
3) по форме представления – аналитическая и графическая;
4) вид оператора модели – алгебраический;
5) по свойствам параметров оператора − это модель линейная с сосредоточенными, стационарными параметрами;
6) по фактору времени − стационарная;
7) по количеству входов/выходов – матричная;
8) по количеству переменных состояния − многомерная;
9) по характеру переменных – дискретная, детерминированная.

6. Задачи на экстремум функции

6.1. Текстовые задачи на экстремум функции одной переменной

1.01. Требуется изготовить открытый сверху цилиндрический сосуд максимальной вместимости. Каковы должны быть размеры сосуда (радиус R и высота H), если на его изготовление имеется S=84.82 дм^2 материала?

1.02. Требуется вырыть яму конической формы (воронку) с образующей a=3 м. При какой глубине объем воронки будет наибольший?

1.03. Требуется изготовить закрытый цилиндрический бак максимальной вместимости. Каковы должны быть размеры бака (радиус R и высота H), если на его изготовление имеется S=18.84 м^2 материала?

1.04. В прямоугольной системе координат через точку $M(2;3)$ проведена прямая, которая вместе с осями координат образует треугольник, расположенный в первом квадранте. Определить длины отрезков, отсекаемых прямой на осях координат, чтобы площадь треугольника была наименьшей?

1.05. Резервуар, открытый сверху, имеет форму прямоугольного параллелепипеда с квадратным основанием. Каковы должны быть размеры резервуара, чтобы на его изготовление пошло наименьшее количество материала, если он должен вмещать 256 л. воды?

1.06. Требуется вырыть яму цилиндрической формы с круглым основанием и вертикальной боковой поверхностью заданного объема V=25 м^3. Определить линейные размеры ямы (радиус R и высота H), чтобы на ее облицовку (дна и боковой поверхности) пошло min количество материала?

1.07. Из круглого бревна радиуса r требуется вырезать балку прямоугольного сечения с основанием b и высотой h. Прочность балки пропорциональна bh^2. При каких значениях b и h прочность балки будет наибольшей?

1.08. Требуется изготовить закрытый цилиндрический бак заданного объема V=50 м^3. Каковы должны быть размеры бака (радиус R и высота H), чтобы на его изготовление пошло наименьшее количество материала.

1.09. Из бревна, имеющего форму усеченного конуса, вырезать балку с поперечным сечением в виде квадрата и осью, совпадающей с осью бревна. Найти размеры балки (сторону квадрата a и длину балки b), при которых объем балки будет наибольшим. Диаметр большего основания бревна равен 2 м, меньшего - 1м, а длина бревна (по оси) равна 18 м.

1.10. Требуется поставить палатку в форме правильной четырехугольной пирамиды заданной боковой поверхности $S=4\sqrt{3}$ м^2. Каковы должны быть размеры палатки (сторона основания a и высота H), чтобы вместимость палатки была наибольшей?

1.11. Цистерна имеет форму прямого кругового цилиндра, завершенного с одной стороны полу-шаром. Вместимость цистерны V=41.89 м^3. Найти радиус цилиндра R, при котором цистерна будет иметь наименьшую полную поверхность.

1.12. Сечение оросительного канала имеет форму равнобочной трапеции, боковые стороны которой равны меньшему основанию. При каком угле наклона боковых сторон сечение канала будет иметь наибольшую площадь?

1.13. Требуется изготовить полотняный шатер, имеющий форму прямого кругового конуса заданной вместимости V=14.14 м^3 . Каковы должны быть размеры конуса (радиус основания R и высота H), чтобы на шатер ушло наименьшее количество полотна?

1.14. Сечение тоннеля имеет форму прямоугольника, завершенного сверху полукругом. Периметр сечения P=35,7 м. При каком радиусе полукруга площадь сечения будет наибольшей?

1.15. Из прямоугольного листа жести размером 24×9 см требуется изготовить открытую сверху коробку, вырезая по углам листа равные квадраты и загибая оставшиеся боковые полосы под прямым углом. Каковы должны быть стороны вырезаемых квадратов, чтобы вместимость коробки была наибольшей?

1.16. Найти min расстояния от прямой $y=2x-4$ до параболы $y=x^2$.

1.17. Из круга вырезан сектор с центральным углом α. Из оставшейся части круга свернута воронка. При каком начальном значении угла α вместимость воронки будет наибольшей?

1.18. Полоса жести шириной 60 см должна быть согнута в виде открытого желоба так, чтобы поперечный разрез имел форму равнобочной трапеции с меньшим основанием равным боковой стороне. Определить ширину желоба (большее основание трапеции) при которой вместимость его была бы наибольшей.

1.19. От реки шириной a отходит под прямым углом канал шириной b. Суда какой наибольшей длины могут входить в этот канал?

1.20. Турист идет из пункта A, находящегося на шоссейной дороге, в пункт B, расположенный в 8 км от шоссе. Расстояние от A до B по прямой составляет 17 км. В каком месте туристу следует свернуть с шоссе, чтобы в кратчайшее время прийти в пункт B, если скорость его по шоссе 5 км/ч, а по бездорожью – 3 км/ч?

6.2. Задачи на наибольшее и наименьшее значения функции одной переменной

Исследовать функцию $f(x) = |x^3 + a_2x^2 + a_1x + a_0|$, заданную на отрезке$[c, d]$,на наибольшее и наименьшее значения.

№ вар.	a_2	a_1	a_0	c	d
2.01.	2	1	-4	-1	3
2.02.	5	8	0	-2	2
2.03.	8	21	14	-2	2
2.04.	11	40	44	- 4	0
2.05.	- 4	5	- 6	1	5
2.06.	-7	16	-16	1	5
2.07.	-10	33	-40	2	6
2.08.	-13	56	-84	0	7
2.09.	-11	40	-50	4	6
2.10.	1	0	-2	-2	2
2.11.	-2	1	-2	-1	3
2.12.	-5	8	-6	0	4
2.13.	-8	21	-20	1	5
2.14.	7	16	10	-2	0
2.15.	10	33	34	-3	-1
2.16.	2	0	-3	-2	2
2.17.	13	56	78	- 4	-2
2.18.	8	20	13	-2	2
2.19.	- 4	4	-3	1	4
2.20.	-7	15	-12	2	6

6.3. Задачи на локальный экстремум функции двух переменных

В задаче (варианты 3.01.–3.20.) исследовать на безусловный экстремум функцию двух переменных.

3.01. $z = x^2 + xy + y^2 - 3x - 6y - 2$;

3.02. $z = 2x^2 - xy + y^2 - 3x - y + 1$;

3.03. $z = 3x^2 - 2xy + y^2 - 2x - 2y + 3$;

3.04. $z = 2x^2 + xy - y^2 - 7x + 5y + 2$;

3.05. $z = x^2 - 3xy - y^2 - 2x + 6y + 1$;

3.06. $z = 3x^2 + xy - 6y^2 - 6x - y + 9$;

3.07. $z = x^2 - 3xy + 2y^2 - 4x + 6y - 2$;

3.08. $z = 4x^2 - 2xy + y^2 - 2x - 4y + 1$;

3.09. $z = 0.5x^2 + xy + y^2 - x - 2y + 8$;

3.10. $z = 8x^2 - xy + 2y^2 - 16x + y - 1$;

3.11. $z = 2x^2 - 2xy + 3y^2 + 2x - 16y + 3$;

3.12. $z = -2x^2 + 6xy - y^2 - 14x + 5$;

3.13. $z = 2x^2 + 3xy - y^2 - 2x + 7y + 6$;

3.14. $z = -3x^2 + 10xy - 2y^2 - 26x + 18y - 1$;

3.15. $z = 3x^2 - 2xy + 2y^2 + 18x + 8y - 1$;

3.16. $z = -3x^2 - 8xy + 5y^2 + 4x + 26y + 3$;

3.17. $z = 2x^2 - 2xy - 3y^2 + 8x + 10y - 6$;

3.18. $z = 5x^2 + 2xy - 3y^2 - 18x - 10y + 4$;

3.19. $z = -7x^2 + 2xy - 5y^2 - 34x + 34y + 5$;

3.20. $z = 2x^2 + 2xy - 3y^2 - 10x + 16y - 7$.

6.4. Задачи на условный экстремум функции двух переменных

Решить задачу (варианты 4.01.– 4.20.) на условный экстремум:

$f(x, y) = x^a y^b \to \min, g(x, y) = \alpha x + \beta y = \gamma,$

$$(a > 0,\ b > 0,\ \alpha > 0,\ \beta > 0,\ \gamma > 0,\ a + b = 1)$$

№ варианта	a	b	α	β	γ
4.01.	0,1	0,9	2	7	140
4.02.	0,2	0,8	4	6	120
4.03.	0,3	0,7	2	4	80
4.04.	0,4	0,6	5	2	50
4.05.	0,6	0,4	5	3	75
4.06.	0,7	0,3	6	1	60
4.07.	0,8	0,2	6	3	90
4.08.	0,9	0,1	5	2	100
4.09.	0,1	0,9	3	4	120
4.10.	0,2	0,8	4	7	140
4.11.	0,3	0,7	1	5	50
4.12.	0,4	0,6	3	6	90
4.13.	0,6	0,4	4	8	160
4.14.	0,7	0,3	3	2	60
4.15.	0,8	0,2	5	3	75
4.16.	0,9	0,1	8	1	80
4.17.	0,1	0,9	2	5	100
4.18.	0,2	0,8	5	6	150
4.19.	0,3	0,7	2	3	60
4.20.	0,4	0,6	4	5	100

6.5. Текстовые задачи на отыскание наибольших и наименьших значений функции двух переменных

5.01.Положительное число aтребуется разбить на три неотрицательных слагаемых так, чтобы их произведение было наибольшим.

5.02.Из всех прямоугольных параллелепипедов, имеющих данный объем V найти тот, полная поверхность которого наименьшая.

5.03.Из всех прямоугольных параллелепипедов, имеющих заданную полную поверхность S найти тот, объем которого наибольший.

5.04.Бункер-параллелепипед, имеющий заданную диагональ L, заполнен зерном. Найти такие размеры бункера, чтобы в него вместилось максимальное количество зерна.

5.05.Рассчитать размеры закрытого бака цилиндрической формы так, чтобы при заданном объеме V=8м^3 на его изготовление было израсходовано наименьшее количество материала.

5.06.Из всех треугольников, вписанных в круг, найти треугольник наибольшей площади.

5.07.Из всех прямоугольников с заданной площадью S, найти такой, периметр которого имеет наибольшее значение.

5.08.Из всех треугольников, имеющих данный периметр, найти такой, площадь S которого имеет наибольшее значение.

5.09. Цилиндр вписан шар радиуса R. Найти размеры цилиндра, имеющего наибольшую полную поверхность.

5.10.Определить наружные размеры открытого прямоугольного ящика с заданной толщиной стенок δ и ёмкостью (внутренней) V так, чтобы на его изготовление было затрачено наименьшее количество материала.

5.11. На плоскости Oxy найти точку $M(x,y)$ сумма квадратов расстояний которой от трех прямых: $L_1{:}x= 0$; $L_2{:}y= 0$; $L_3{:}x-y+ 1 = 0$ была бы наименьшей.

5.12.При каких размерах открытая прямоугольная ванна данной вместимости V имеет наименьшую поверхность?

5.13.Мощность цеха сборки составляет 100 изделий типа A или 300 изделий типа B в сутки. Отдел технического контроля в сутки может проверить не более 150 изделий. Изделие типа A стоит вдвое дороже изделия типа B. Сколько изделий обоих типов следует выпускать в сутки, чтобы общая стоимость продукции была максимальной?

5.14. Найти кратчайшее расстояние от прямой: $x+ y= 4$ до эллипса: $4y^2+ x^2= 4$.

5.15.Скорость изготовления изделий типа A составляет 20 штук в час, а скорость изготовления изделий типа Bсоставляет 60 штук в час. На станке можно изготовить в час не более 30 изделий. Изделие типа A стоит вдвое дороже изделия типа B. Сколько изделий обоих типов следует изготовлять в час, чтобы общая стоимость продукции была максимальной?

5.16. Найти *min* расстояния от прямой: $x+ y= -5$ до эллипса: $9y^2+ 4x^2= 36$.

5.17.Параллелепипед имеет заданную диагональ d. Найти размеры этого параллелепипеда, чтобы его объем был наибольший.

5.18.На плоскости Oxy найти точку $M(x,y)$ сумма квадратов расстояний которой от трех прямых: $L1$: $x = 0$; $L2$: $y = 0$; $L3$: $y - x = 2$ была бы наименьшей.

5.19.При каких размерах открытой прямоугольной коробки объема V на ее изготовление потребуется минимум материала?

5.20.Шар радиуса R вписан в круговой конус и касается боковой поверхности и основания конуса. Найти размеры конуса, имеющего наименьшую полную поверхность.

6.6. Задачи по линейному программированию

Для изготовления двух видов продукции имеются три вида ресурсов, объемы которых ограничены величинами b_1, b_2, b_3 соответственно. Расход i–го вида ресурса на изготовление одной единицы j–го вида продукции равен a_{ij}, $i = 1,2,3$, $j = 1,2$. Объем выпуска каждого из видов продукции ограничен числом x_1^* и x_2^* единиц, а прибыль, получаемая от реализации одной единицы изготовленной продукции, равна c_1 и c_2, соответственно. Данные задачи могут быть представлены в матрично-векторном виде

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, x^* = \begin{pmatrix} x_1^* & x_2^* \end{pmatrix},\ c = \begin{pmatrix} c_1 & c_2 \end{pmatrix}$$,или в виде (табл. 3):

Исходные данные задачи. Таблица 3

Номер ресурса	Объем ресурса	Номер продукции	
		1	2
1	b_1	a_{11}	a_{12}
2	b_2	a_{21}	a_{22}
3	b_3	a_{31}	a_{32}
Ограничения по выпуску		x_1^*	x_2^*
Прибыль		c_1	c_2

Требуется сверстать план выпуска продукции (число единиц продукции по каждому виду), удовлетворяющий принятым ограничениям и приносящий максимум прибыли после реализации выпущенной продукции.

Исходные данные:

6.01. $A=\begin{pmatrix}1 & 4\\ 1 & 1\\ 4 & 1\end{pmatrix}$, $b=\begin{pmatrix}60\\ 24\\ 72\end{pmatrix}$, $x^*=(17 \quad 14)$, $c=(1 \quad 1)$,

6.02. $A=\begin{pmatrix}2 & 9\\ 5 & 6\\ 11 & 1\end{pmatrix}$, $b=\begin{pmatrix}162\\ 141\\ 176\end{pmatrix}$, $x^*=(15 \quad 17)$, $c=(4 \quad 3)$,

6.03. $A=\begin{pmatrix}1 & 9\\ 4 & 5\\ 11 & 1\end{pmatrix}$, $b=\begin{pmatrix}135\\ 106\\ 150\end{pmatrix}$, $x^*=(9 \quad 14)$, $c=(1 \quad 4)$,

6.04. $A=\begin{pmatrix}2 & 9\\ 1 & 1\\ 10 & 1\end{pmatrix}$, $b=\begin{pmatrix}162\\ 25\\ 160\end{pmatrix}$, $x^*=(15 \quad 17)$, $c=(7 \quad 2)$,

6.05. $A=\begin{pmatrix}2 & 11\\ 1 & 1\\ 9 & 1\end{pmatrix}$, $b=\begin{pmatrix}187\\ 26\\ 162\end{pmatrix}$, $x^*=(17 \quad 16)$, $c=(3 \quad 1)$,

6.06. $A=\begin{pmatrix}2 & 13\\ 1 & 1\\ 11 & 1\end{pmatrix}$, $b=\begin{pmatrix}221\\ 28\\ 198\end{pmatrix}$, $x^*=(17 \quad 16)$, $c=(6 \quad 2)$,

6.07. $A=\begin{pmatrix}3 & 7\\ 5 & 6\\ 6 & 1\end{pmatrix}$, $b=\begin{pmatrix}140\\ 137\\ 90\end{pmatrix}$, $x^*=(14 \quad 19)$, $c=(8 \quad 2)$,

6.08. $A=\begin{pmatrix}2 & 11\\ 1 & 1\\ 10 & 3\end{pmatrix}$, $b=\begin{pmatrix}176\\ 25\\ 180\end{pmatrix}$, $x^*=(17 \quad 15)$, $c=(5 \quad 5)$,

6.09. $A=\begin{pmatrix}2 & 9\\ 3 & 2\\ 11 & 3\end{pmatrix}$, $b=\begin{pmatrix}171\\ 61\\ 176\end{pmatrix}$, $x^*=(15 \quad 18)$, $c=(5 \quad 3)$,

6.10. $A=\begin{pmatrix}3 & 13\\ 3 & 2\\ 3 & 1\end{pmatrix}$, $b=\begin{pmatrix}234\\ 69\\ 20\end{pmatrix}$, $x^*=(19 \quad 17)$, $c=(3 \quad 5)$, 6.11. $A=\begin{pmatrix}1 & 14\\ 4 & 5\\ 14 & 1\end{pmatrix}$, $b=\begin{pmatrix}266\\ 146\\ 280\end{pmatrix}$, $x^*=\begin{pmatrix}19 & 18\end{pmatrix}$, $c=\begin{pmatrix}3 & 3\end{pmatrix}$,

6.12. $A=\begin{pmatrix}1 & 4\\ 1 & 1\\ 10 & 1\end{pmatrix}$, $b=\begin{pmatrix}72\\ 27\\ 180\end{pmatrix}$, $x^*=\begin{pmatrix}17 & 17\end{pmatrix}$, $c=\begin{pmatrix}6 & 5\end{pmatrix}$,

6.13. $A=\begin{pmatrix}1 & 9\\ 15 & 18\\ 11 & 2\end{pmatrix}$, $b=\begin{pmatrix}153\\ 423\\ 187\end{pmatrix}$, $x^*=\begin{pmatrix}16 & 16\end{pmatrix}$, $c=\begin{pmatrix}1 & 9\end{pmatrix}$,

6.14. $A=\begin{pmatrix}1 & 8\\ 4 & 5\\ 15 & 2\end{pmatrix}$, $b=\begin{pmatrix}160\\ 127\\ 225\end{pmatrix}$, $x^*=\begin{pmatrix}14 & 19\end{pmatrix}$, $c=\begin{pmatrix}6 & 7\end{pmatrix}$,

6.15. $A=\begin{pmatrix}1 & 10\\ 1 & 1\\ 6 & 1\end{pmatrix}$, $b=\begin{pmatrix}170\\ 26\\ 96\end{pmatrix}$, $x^*=\begin{pmatrix}15 & 16\end{pmatrix}$, $c=\begin{pmatrix}9 & 1\end{pmatrix}$,

6.16. $A=\begin{pmatrix}1 & 3\\ 5 & 4\\ 1 & 3\end{pmatrix}$, $b=\begin{pmatrix}57\\ 109\\ 176\end{pmatrix}$, $x^*=\begin{pmatrix}15 & 18\end{pmatrix}$, $c=\begin{pmatrix}3 & 7\end{pmatrix}$,

6.17. $A=\begin{pmatrix}3 & 13\\ 1 & 1\\ 6 & 1\end{pmatrix}$, $b=\begin{pmatrix}260\\ 30\\ 120\end{pmatrix}$, $x^*=\begin{pmatrix}19 & 19\end{pmatrix}$, $c=\begin{pmatrix}9 & 1\end{pmatrix}$,

6.18. $A=\begin{pmatrix}1 & 6\\ 2 & 3\\ 13 & 3\end{pmatrix}$, $b=\begin{pmatrix}108\\ 63\\ 195\end{pmatrix}$, $x^*=\begin{pmatrix}14 & 17\end{pmatrix}$, $c=\begin{pmatrix}8 & 6\end{pmatrix}$,

6.19. $A=\begin{pmatrix}1 & 3\\ 3 & 2\\ 11 & 3\end{pmatrix}$, $b=\begin{pmatrix}60\\ 61\\ 176\end{pmatrix}$, $x^*=\begin{pmatrix}15 & 19\end{pmatrix}$, $c=\begin{pmatrix}3 & 9\end{pmatrix}$,

6.20. $A=\begin{pmatrix}1 & 5\\ 1 & 1\\ 9 & 2\end{pmatrix}$, $b=\begin{pmatrix}85\\ 25\\ 162\end{pmatrix}$, $x^*=\begin{pmatrix}17 & 16\end{pmatrix}$, $c=\begin{pmatrix}5 & 3\end{pmatrix}$.

7. Универсальная система компьютерной математики «Mathematica»

Целью данного раздела учебного пособия является ознакомление обучающихся с УСКМ **Mathematica-5** и применение приобретенных знаний для решения задач курса высшей математики [Аюпов, 2015].

Mathematica-5 (и все ее предыдущие и последующие версии) создана фирмой **WolframResearch** во главе с ее президентом и главным разработчиком программ – Стивеном Вольфрамом.

В дальнейшем, ссылаясь на систему **Mathematica-5**, мы будем говорить **Mathematica**. Данная система совместима с любым современным компьютером, работающим под управлением операционной системы **Windows**. При этом большинство команд и функций системы **Mathematica** не зависят от типа компьютера.

Система **Mathematica** является профессиональной по своим возможностям. При этом она открыта и для неопытных пользователей. Пользовательский интерфейс системы **Mathematica** таков, что пользователь, имеющий элементарные навыки работы с **Windows**-приложениями, может сразу начать работу в системе **Mathematica.**

Работа в **Mathematica** происходит в режиме сессии **(session).** Назовем сессией работу с **Mathematica** в промежутке времени от вхождения в программу **Mathematica** до выхода из нее. Во время сессии можно работать с одним документом или попеременно с несколькими; при этом окна всех документов присутствуют на экране, но активным является только одно из них. Решение любой задачи с помощью **Mathematica** начинается с того, что нужно набрать с клавиатуры выражение, содержащее символы, числа, строки. После набора выражения следует запустить его вычисление нажатием клавиш **Shift + Enter** или **Enter** на цифровой клавиатуре справа. Если выражение набрано без ошибок **Mathematica** вычислит его, и последует вывод; если же в выражении есть синтаксические ошибки, **Mathematica** выдаст сообщение о них, которое поможет вам их исправить. Одна из составляющих успеха в работе с **Mathematica** – научиться безошибочно (в соответствии с правилами синтаксиса) составлять выражения. Этому может помочь справочная система **Help** и панели кнопочного набора (палитры **–Palettes),** которые значительно облегчат процесс набора. Допустим, что вы правильно набрали выражение, и **Mathematica** вычислила его. Тогда одновременно с появлением ответа набранное выражение будет помечено ремаркой **In[1] –,** а появившийся ответ – ремаркой **Out[1]**–; это входная и выходная ячейки. Под выходной ячейкой имеется горизонтальная черта, ниже которой ничего нет. Это означает, что **Mathematica** готова принять новое выражение. Как только вы начнёте печатать первый знак, горизонтальная черта исчезнет, а ваше новое выражение будет располагаться ниже исчезнувшей черты. После вычисления этого выражения вместе с ответом на него возникнут пометки **In[2]–, Out[2]** – и новая горизонтальная черта и т.д. Ячейки ввода и вывода помечаются справа отдельными квадратными скобками с треугольничками в верхней части (в скобке ячейки вывода есть ещё дополнительная горизонтальная чёрточка), а вместе они ограничены в правой части экрана общей квадратной скобкой; таким образом, сформирована группа ячеек. В группу могут входить также ячейки третьего типа – текстовые, их используют для заголовков и различного рода комментариев. Свойства входной и выходной ячеек различаются. Во входную ячейку легко помещается курсор, что позволяет как угодно редактировать ее стандартным образом. Чтобы поместить курсор в выходную ячейку (например, для того чтобы скопировать содержимое), предварительно нужно перевести ее во входной формат. Для этого выделяем выходную ячейку (щелчком мыши, подводя её указатель к скобке выходной ячейки), затем входим в меню **Format** и выбираем опцию **Style**, а в ней, переходя вправо к иконке, содержащей список форматов, — стиль **Input.** Скобка, окаймляющая группу ячеек, используется для того, чтобы свернуть (сделать невидимым) содержимое выходной ячейки. Для этого подводят справа к внешней скобке указатель мыши до того момента, когда он примет вид стрелки, направленной влево и ограниченной у острия вертикальным отрезком. После двукратного щелчка левой кнопкой мыши выходная ячейка и горизонтальная черта исчезают. Вы не сможете продолжать вычисления, пока не восстановите горизонтальную линию, расположив указатель мыши ниже измененной ячейки и щелкнув левой кнопкой. Развернуть содержимое спрятанной выходной ячейки можно тем же приёмом. Результаты работы можно сохранить в виде файлов с расширением .**nb.**

7.1. Основные классы данных

К основным классам данных относятся *числовые данные и константы, символьные данные, списки*.

К *числовым данным* относятся: двоичные числа, десятичные числа и числовые константы. Десятичные числа представлены целыми **(Integer),** рациональными **(Rational),** действительными **(Real)**и комплексными *(Complex*) числами. Примеры представлений чисел:1) 27 – целое число, 2) 1⁄3 – рациональное, 3) 3.29745 – действительное, 4) 8+i 6 – комплексное.

К константам в системе **Mathematica** отнесены: Pi– число π, имеющее значение 3,141593…,
E– число **e**=2.71828… – основание натурального логарифма, **Degree** – число радиан в одном градусе, равное $\frac{\pi}{180}$, **GoldenRatio** – константа, равная $(1+\sqrt{5})/2$, определяющая деление отрезка по правилу золотого сечения, **I** – мнимая единица, равная $i=\sqrt{-1}$, **Infinity** – бесконечность ∞ и другие.

Символьные данные могут быть представлены в виде одного или нескольких идущих подряд символов, например, **a**, **b**, **c**, … или **a1**, **bb**, **xyz24**, и т.д. Символьные строки задаются цепочкой символов, заключенных в кавычках, например, "**Math-**

ematica", "**computer\tsystem\nMathematica**". В последнем примере символы **\t** и **\n** –являются опциями, первая из которых определяет табуляцию, вторая – новую строку.

Наиболее общим видом сложных данных в системе Математика являются списки (**Lists**). Списки представляют собой совокупности однородных или разнородных данных, сгруппированных с помощью фигурных скобок или с помощью функции **List**[, , …].Примеры списков:{**a,b,c**} –список из трех символьных данных,{**1,2,3**} – список из трех целых чисел,{**a/b,x+y, x**^{2}**, Sin[*x*]**} – список из четырех математических выражений,{2.5,"**abc", x**^{2}} – список из трех разнотипных данных,{{**a,b**},{**c,d**}} – список, состоящий из двух списков, то есть список, эквивалентный матрице$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.Таким образом, списки могут быть простыми (одноуровневыми) и сложными (многоуровневыми). С помощью списков представляются множественные данные – массивы.

7.2. Объекты и идентификаторы

В общем случае система **Mathematica** оперирует с объектами. Под ними подразумеваются числа, константы, символы, строки, математические выражения, графические и звуковые объекты и другие. Каждый объект характеризуется своим именем – идентификатором. Это имя должно быть уникальным, т.е. единственным.

Правила задания идентификаторов: **ssssss** – имя объекта, заданного пользователем, **Ssssss**– имя объекта, входящего в ядро системы, **$Sssss** – имя системного объекта. Здесь **s** –любая буква или цифра. При этом первый символ – всегда буква.

7.3. Функции, опции и атрибуты

Функция – это объект, имеющий имя и список параметров, перечисленных через запятые и заключенных в квадратные скобки.

Формат записи функции:

Имя_функции[**o1**,**o2**,**o3**,…], где **o1**,**o2**,**o3**,… – объекты (параметры, опции, математические выражения и т.д.).

Опция – это параметр функции, задающий дополнительные условия выполнения этой функции. Опция задается так:

Имя_опции→Значение _опции

Значением опции обычно является **слово**.

Например, **Plot**[**Sin**[**x**],{**x, 0, 20**},**Axes→None**].

В этом примере **Axes** – опция, определяющая наличие осей, **None** – ее значение, смысл которого в том, что оси не нужно выводить. Чтобы узнать, какие опции используются в данной функции, нужно выполнить функцию **Options**[**Имя_функции**].

Каждый объект может характеризоваться своими свойствами и признаками, которые называются *атрибутами*.

Чтобы определить какие атрибуты имеет конкретная функция, например, **Sin**, нужно сформировать функцию-запрос: **Attributes**[**Sin**], на который система выдаст список атрибутов этой функции:

{**Listable**, **NumericFunction**, **Protected**}, которые означают соответственно, что функция **Sin**является дистрибутивной, числовой и что сочетание символов **Sin**защищено от использования в качестве идентификатора.

7.4. Арифметические функции и выражения

К арифметическим функциям относятся следующие функции:

Plus[**a,b**, ...] – сумма $a+b+\cdots$;

Times[**a,b**, ...] – произведение $a\cdot b\cdot\cdots$;

Divide[a,b] – деление a на b;

Power[a,n] – возведение a в степень n;

Sqrt[a] – извлечение из a квадратного корня;

Exp[**x**] – экспонента e^x;

Log[**x**] – натуральный логарифм $\ln x$;

Log[**a, *x***] – логарифм по основанию a от x: $\log_a x$;

n! – функция эн факториал ($n! = 1 \cdot 2 \cdot 3 \cdots n$);

Sin[x], Cos[x], Tan[x], Cot[x] –тригонометрические функции $\sin x, \cos x, tgx, ctgx$;

ArcSin[x], ArcCos[x], ArcTan[x], ArcCot[x] – обратные тригонометрические функции $\arcsin x, \arccos x, arctgx, arcctgx$;

Abs[x] – модульх: |x|;

Max[x1,x2, …, xn] – максимальное число из чисел $x1, x2, \cdots, xn$;

Min[x1,x2, …, xn] – минимальное число из чисел $x1, x2, \cdots, xn$;

FactorInteger[n] – разложение целого числа на простые сомножители;

N[expr, n] – вычисление выражения **expr** с точностью до **n** знаков после десятичной точки.

Любое выражение в системе **Mathematica** строится из атомарных выражений по правилу

h[$e_1, e_2, \ldots, e_n$],

где **h** – заголовок выражения, $e_1, e_2, \ldots, e_n$ – элементы выражения. Атомарные выражения могут быть соединены знаками действий. В системе **Mathematica** выражения подразделяются на арифметические, алгебраические, тригонометрические и смешанные (общего вида) выражения. Следующие записи представляют собой различные выражения:1) **Sin[x]**,2) **Integrate[Cos[x],x]**,3) **2*Log**[**x**]**+3**,
4**)** **(a+b^2+c^3)/(4*x-3*y)**.

Для простоты выражения будем называть <u>формулами</u>. Таким образом, для записи выражений используются как операторы: +;−;*;/;^ и т.д., круглые скобки, так и функции. При составлении выражений придерживаются следующих соглашений: 1) круглые скобки используются для выделения части выражения и для задания последовательности действий; 2) знак умножения может быть заменен пробелом; 3) встроенные функции начинаются с большой буквы; 4) параметры функции задаются в квадратных скобках.

7.5. Работа со списками

Формирование списков

В системе **Mathematica** имеется четыре функции формирования списков: **List, Table, Range, Array**.

<u>Формат функции **List**.</u>

$\mathbf{List}[\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_n}]$, где $\mathbf{x_i}$ – объекты системы **Mathematica**. Например, $\mathbf{List}[\mathbf{1}, \mathbf{Cos}[\mathrm{x}], \mathbf{List}\ [\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}]]$ формирует неоднородный двухуровневый список, $\mathbf{List}[\{\mathbf{1}, \mathbf{4}, \mathbf{9}\}, \{\mathbf{0}, \mathbf{1}, \mathbf{4}\}, \{\mathbf{0}, \mathbf{0}, \mathbf{1}\}]//\mathbf{MatrixForm} \Rightarrow \begin{pmatrix} 1 & 4 & 9 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}$ формирует однородный двухуровневый список, который представлен в виде матрицы, $\mathbf{List}[\{\mathbf{1}, \mathbf{4}, \mathbf{9}\}, \{\mathbf{0}, \mathbf{1}, \mathbf{4}\}, \{\mathbf{0}, \mathbf{0}, \mathbf{1}\}]//$ $\mathbf{TableForm} \Rightarrow \begin{matrix} 1 & 4 & 9 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{matrix}$ тот же список, представленный в виде таблицы.

<u>Формат функции **Table**.</u>

$\mathbf{Table}[\mathrm{f}, \{\boldsymbol{n}\}]$ – формирует список $\{\mathrm{f}, \mathrm{f}, \dots, \mathrm{f}\}$, состоящий из **n** знаков (выражений) f;

$\mathbf{Table}[\mathrm{f}[\boldsymbol{i}], \{\boldsymbol{i}, \boldsymbol{n}\}]$ – формирует список $\{\mathrm{f}[\mathbf{1}], \mathrm{f}[\mathbf{2}], \dots, \mathrm{f}[\boldsymbol{n}]\}$;

$\mathbf{Table}[\mathrm{f}[\boldsymbol{i}, \boldsymbol{j}], \{\boldsymbol{i}, \boldsymbol{n}\}, \{\boldsymbol{j}, \boldsymbol{m}\}]$ – формирует двухуровневый вложенный список и т.д. Например,

$\mathbf{Table}\left[\boldsymbol{e}^{\boldsymbol{i}}, \{\mathbf{5}\}\right] \Rightarrow \left\{\boldsymbol{e}^{\boldsymbol{i}}, \boldsymbol{e}^{\boldsymbol{i}}, \boldsymbol{e}^{\boldsymbol{i}}, \boldsymbol{e}^{\boldsymbol{i}}, \boldsymbol{e}^{\boldsymbol{i}}\right\}$;

$\mathbf{Table}\left[\boldsymbol{e}^{\boldsymbol{i}}, \{\boldsymbol{i}, \mathbf{5}\}\right] \Rightarrow \left\{\boldsymbol{e}^{\mathbf{1}}, \boldsymbol{e}^{\mathbf{2}}, \boldsymbol{e}^{\mathbf{3}}, \boldsymbol{e}^{\mathbf{4}}, \boldsymbol{e}^{\mathbf{5}}\right\}$;

$\mathbf{Table}\ [\boldsymbol{i} * \boldsymbol{j}, \{\boldsymbol{i}, \mathbf{3}\}, \{\boldsymbol{j}, \mathbf{3}\}] \Rightarrow \{\{\mathbf{1}, \mathbf{2}, \mathbf{3}\}, \{\mathbf{2}, \mathbf{4}, \mathbf{6}\}, \{\mathbf{3}, \mathbf{6}, \mathbf{9}\}\}$.

<u>Формат функции **Range**.</u>

$\mathbf{Range}[\boldsymbol{i_{min}}, \boldsymbol{i_{max}}, \boldsymbol{d_i}]$ – формирует список из целых чисел от $\boldsymbol{i_{min}}$ до $\boldsymbol{i_{max}}$ с шагом $\boldsymbol{d_i}$. Например, $\mathbf{Range}[\mathbf{2}, \mathbf{12}, \mathbf{3}] \Rightarrow \{\mathbf{2}, \mathbf{5}, \mathbf{8}, \mathbf{11}\}$.

Если $\boldsymbol{i_{min}}$ и $\boldsymbol{d_i}$ опущены, то формируется список $\{\mathbf{1,2,3,\ldots},\boldsymbol{i_{max}}\}$, например, $\mathbf{Range}[\mathbf{10}] \Rightarrow \{\mathbf{1,2,3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10}\}$.

<u>Формат функции **Array**.</u>

1) $\mathbf{Array}\ [\boldsymbol{f}, \mathbf{n}] \Rightarrow \{f[\mathbf{1}], f[\mathbf{2}], \ldots, f[\boldsymbol{n}]\}$
2) $\mathbf{Array}[f, \{\boldsymbol{n_1}, \boldsymbol{n_2}\}] \Rightarrow$

$$\{\{f[\mathbf{1},\mathbf{1}], f[\mathbf{1},\mathbf{2}], \ldots, f[\mathbf{1},\boldsymbol{n_2}]\}, \{f[\mathbf{2},\mathbf{1}], f[\mathbf{2},\mathbf{2}], \ldots, f[\mathbf{2},\boldsymbol{n_2}]\}, \ldots, \{f[\boldsymbol{n_1},\mathbf{1}], \ldots, f[\boldsymbol{n_1},\boldsymbol{n_2}]\}\}$$

Структура и формы представления списков

1)**Length**[s] – длина списка.

$\mathbf{Length}[\{\mathbf{1}, \mathbf{7}, \mathbf{a}, \mathbf{Cos}[\mathbf{x}]\}] \Rightarrow \mathbf{4}$

2)**MatrixForm** [**m**] – матричная форма сложного списка.

$\mathbf{m} = \{\{\mathbf{1}, \mathbf{2}, \mathbf{3}\}, \{\mathbf{4}, \mathbf{5}, \mathbf{6}\}, \{\mathbf{7}, \mathbf{8}, \mathbf{9}\}\}$.

$\mathbf{MatrixForm}[\mathbf{m}] \Rightarrow \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{pmatrix}$ или $\mathbf{m}//\mathbf{MatrixForm} \Rightarrow \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{pmatrix}$.

3)**TableForm**[**m**] – табличная форма сложного списка.

$\mathbf{TableForm}[\mathbf{m}] \Rightarrow \begin{matrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{matrix}$ или $\mathbf{m}//\ \mathbf{TableForm} \Rightarrow$

$\begin{matrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} & \mathbf{9} \end{matrix}$

4)**Dimensions** [s] – список размерностей сложного списка **s**.

$\mathbf{Dimensions}[\mathbf{m}] \Rightarrow \{\mathbf{3}, \mathbf{3}\}$

$\mathbf{Dimensions}[\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\}, \{\mathbf{e}, \mathbf{f}\}] \Rightarrow \{\mathbf{3}, \mathbf{2}\}$

$\mathbf{Dimensions}[\{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}\}] \Rightarrow \{\mathbf{4}\}$– список, размерность списка.

Dimensions$[\{\mathbf{a}, \mathbf{a}, \mathbf{a}\}] \Rightarrow \{\mathbf{3}\}$.
Dimensions$[\mathbf{a} + \mathbf{a}^2 + \mathbf{a}^3] \Rightarrow \{\mathbf{3}\}$.

Взятие функций от списков

Пусть $\boldsymbol{f}$ – математическая функция или операция, **s** – список. Взятие функций от списка можно оформить тремя способами:

1) $\boldsymbol{f}$ **[s]**, 2) $\boldsymbol{f}$**@** **s**, 3) **s** // $\boldsymbol{f}$.

Пример: Пусть $\boldsymbol{s1} = \left\{\mathbf{0}, \frac{\boldsymbol{\pi}}{\mathbf{6}}, \frac{\boldsymbol{\pi}}{\mathbf{4}}, \frac{\boldsymbol{\pi}}{\mathbf{3}}, \frac{\boldsymbol{\pi}}{\mathbf{2}}, \boldsymbol{\pi}\right\}, \boldsymbol{s2} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}\}, \boldsymbol{s3} = \{\mathbf{4}, \mathbf{5}, \mathbf{6}\}$.

Тогда:

$\mathbf{Cos}[\mathbf{s1}] \Rightarrow \left\{\mathbf{1}, \frac{\sqrt{3}}{2}, \frac{\sqrt{2}}{2}, \frac{1}{2}, \mathbf{0}, -\mathbf{1}\right\}$,$\mathbf{Cos} \;@\; \mathbf{s1} \Rightarrow \left\{\mathbf{1}, \frac{\sqrt{3}}{2}, \frac{\sqrt{2}}{2}, \frac{1}{2}, \mathbf{0}, -\mathbf{1}\right\}$ $\mathbf{s1}$ // $\mathbf{Cos} \Rightarrow \left\{\mathbf{1}, \frac{\sqrt{3}}{2}, \frac{\sqrt{2}}{2}, \frac{1}{2}, \mathbf{0}, -\mathbf{1}\right\}$;**Plus**[**s2,s3]**или**s2+s3**⇒{**5,7,9**}; $\mathbf{e}^{\mathbf{s2}} \Rightarrow \{\mathbf{e}^{\mathbf{1}}, \mathbf{e}^{\mathbf{2}}, \mathbf{e}^{\mathbf{3}}\}$; $\mathbf{s3}^{\mathbf{2}} \Rightarrow \{\mathbf{16}, \mathbf{25}, \mathbf{36}\}$.

Функции выявления структуры списков

MemberQ[**s, form**] – принадлежность элемента списку **s**. Например,

MemberQ$[\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, \mathbf{c}\,] \Rightarrow$**True.**

Position[**s, form**] – номер элемента в списке **s**.

Position$[\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, \mathbf{c}] \Rightarrow \{\{\mathbf{3}\}\}$.

Part[**s**,**i**] – выбор **i**–го элемента списка **s**.

Например, **Part**$[\{\mathbf{1}, \mathbf{b}, \mathbf{x}\}$,**2**] ⇒{**b**}.

Select[**s, crit**] – выбор элементов списка **s**
по критерию **crit**.

Select$[\{\mathbf{1}, \mathbf{a}, \mathbf{2}, \mathbf{b}, \mathbf{3}, \mathbf{c}\}$, **NumberQ**]⇒$\{\mathbf{1}, \mathbf{2}, \mathbf{3}\}$.

Вывод элементов списка

MatrixForm[s] – выводит список в виде массива (матрицы).

TableForm[s] – выводит список в форме таблицы.

Sort[s] – сортирует список в каноническом порядке по возрастанию: сначала числа, затем буквы в алфавитном порядке.

Transpose[**m**] – транспонирование матрицы $\mathbf{m} \to \mathbf{m}'$.
Union[**s**] – сортирует, удаляя повтор списка.
First[**s**] – первый элемент списка или выражение.
Last[**s**] – последний элемент списка или выражения.
Delete[**s, i**] – удалить **i**–й элемент списка **s**.
ReplacePart[**s, x, i**] – **i**–й элемент списка **s** заменяется на **x**.
Sort[**s**, **Greater**] – сортирует список в порядке убывания.
Insert[**s, x, i**] – на**i**–е место в списке вставляет **x**.

7.6. Операции линейной алгебры

Векторы, матрицы и действия над ними

Векторы в системе **Mathematica** задаются как одноуровневые списки. Матрицы – как двухуровневые списки. Чтобы выделить$(\mathbf{i}, \mathbf{j})$–йэлемент
цы $\mathbf{m} = \{\{\mathbf{m_{11}}, \mathbf{m_{12}}, \mathbf{m_{13}}\}, \{\mathbf{m_{21}}, \mathbf{m_{22}}, \mathbf{m_{23}}\}, \{\mathbf{m_{31}}, \mathbf{m_{32}}, \mathbf{m_{33}}\}\}$, нужно задать **m** [[$\mathbf{i}, \mathbf{j}$]]. При этом выделится элемент матрицы **m** с индексами $\mathbf{i}, \mathbf{j}$. Аналогично выделяется **i**– я строка матрицы **m**:задается **m** [[**i**]]. Действия с векторами и матрицами определяются следующим образом:

c * v – умножение вектора **v**на скаляр **c**;
c * m – умножение матрицы **m** на число **c**;
v .u или **Dot**[**v, u**] – скалярное произведение векторов;
v .m – произведение вектора на матрицу,
m .v – произведение матрицы на вектор,
$\mathbf{m_1}$.$\mathbf{m_2}$ – произведение двух матриц;
v×**u** или **Cross**[**v, u**] – векторное произведение
двух векторов.

Формирование матриц и вычисление функций от матриц

Следующие функции относятся к функциям, формирующим матрицы. **IdentityMatrix**[**n**] – $\mathbf{E_n}$– формирует единичную матрицу **n**–го порядка, например: **IdentityMatrix**[**3**]

$\Rightarrow\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$;

DiagonalMatrix[{$\boldsymbol{d_1}, \boldsymbol{d_2}, \ldots, \boldsymbol{d_n}$}] – формирует диагональную матрицу **n**– го порядка,

например **DiagonalMatrix**[{$\boldsymbol{d_1}, \boldsymbol{d_2}$}] $\Rightarrow\begin{pmatrix} \boldsymbol{d_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{d_2} \end{pmatrix}$;

Inverse[**m**] $\Rightarrow\mathbf{m^{-1}}$ – формирование обратной матрицы $\mathbf{m^{-1}}$поотношению к матрице **m**;

Det[**m**] – вычисляет определитель квадратной матрицы **m**;

Minors[**m, k**] – выдаёт список миноров **k** –го порядка матрицы **m**;

MatrixPower[**m, n**] – вычисление **n**– й степени квадратной матрицы **m**.

7.7.Решение уравнений и систем уравнений

Решение нелинейных уравнений и систем уравнений

Многие математические задачи сводятся к решению одного уравнения или системы уравнений. В общем случае решаемые уравнения и системы – не линейны. Для решения уравнений как одиночных, так и систем в численном и символьном виде в системе**Mathematica** имеются следующие основные функции:**Solve, NSolve, Reduce, Roots, FindRoot.**

Функция **Solve** служит для точного решения уравнений.

Формат функции **Solve[уравнение, неизвестное]** – для решения одного уравнения и **Solve[{система уравнений}, {список неизвестных}] –** для систем уравнений.

Примеры: Решить уравнение:$x^2 - 5x + 4 = 0$.

Решение: $x_1=1$, $x_2=4$.

Решение в системе**Mathematica**:**Solve**[**x²–5x+4==0, x**]

⇒{{**x→ 1},{x→ 4**}}

Решить систему уравнений: $\begin{cases} 2x+3y=1 \\ x+y=0 \end{cases}$.

Решение: x =–1, y = 1.
Решение в системе**Mathematica**:**Solve** [{**2 x + 3y == 1, x + y == 0}, {x, y**}] $\Rightarrow$ {{**x** $\rightarrow$ **−1**}, {**y** $\rightarrow$ **1**}}

Функция **NSolve** служит для приближенного решения уравнений.

Формат функции **NSolve[уравнение, неизвестное] –** для решения одного уравнения и **NSolve[{система уравнений}, {список неизвестных}] –** для систем уравнений.

Примеры: Решить уравнение: $x^3 – x + 1 = 0$.

Решение:
$x_1 = – 1{,}3247$; $x_2 = 0{,}6623 – 0{,}5622\, i$; $x_2 = 0{,}6623 + 0{,}5622\, i$.
Решение в системе **Mathematica**: **NSolve** [**x³– x + 1 = = 0, x**] $\Rightarrow$ {{**x**$\rightarrow$ **– 1.3247**}, {**x** $\rightarrow$ **0.6623 – 0.5622 I** }, {**x** $\rightarrow$ **0.6623 + 0.5622 I**}}.

Решить систему уравнений: $\begin{cases} x + y + 1 = 0 \\ x + 3y = \sqrt{2} \end{cases}$.

Решение: x= –2,2071; y = 1, 2071.

Решение в системе **Mathematica**: **NSolve**[{**x+y+1 == 0, x+3y==** $\sqrt{2}$ **}, {x, y}**] $\Rightarrow$ **{ {x** $\rightarrow$ **– 2.20711}, {y** $\rightarrow$ **1.20711} }**

Функция **Reduce** служит для решения уравнений с параметрами (с учетом их особых значений). Формат функции **Reduce[уравнение, неизвестное] –** для решения одного уравнения и **Reduce [{система уравнений}, {список неизвестных}] –** для систем уравнений.
Решить уравнение: $ax^3 + bx = 0$.
Решение: $x_1 = 0$; $x_2 = -\frac{\sqrt{b}}{\sqrt{a}}\, i$; $x_2 = \frac{\sqrt{b}}{\sqrt{a}}\, i$, $a \neq 0$.
Решение в системе **Mathematica**: **Reduce[a * x³ + b * x == 0, x**]
$\Rightarrow$**b == 0 &&a == 0 || a ≠ 0 && (x ==** $-\frac{\sqrt{\mathbf{b}}}{\sqrt{\mathbf{a}}}$ **I || x == +**$\frac{\sqrt{\mathbf{b}}}{\sqrt{\mathbf{a}}}$ **I) || x==0**

Функция **Roots** служит для поиска корней уравнения. Формат функции

Roots[уравнение, неизвестное] – для решения одного уравнения и

Roots[{система уравнений}, {список неизвестных}] – для систем уравнений.

Решить уравнение: $x^2 - 5x + 4 = 0$.

Решение: $x_1=1$, $x_2=4$.

Решение в системе **Mathematica**: **Roots** [$\mathbf{x^2-5x+4==0,\ x}$] $\Rightarrow \mathbf{x == 1 \,||\, x == 4}$

Решить систему уравнений: $\begin{cases} \mathbf{2x + 3y = 1} \\ \mathbf{x + y = 0} \end{cases}$.

Решение: $x = -1$, $y = 1$.

Решение в системе **Mathematica**:**Root** [{**2 x + 3y == 1, x + y == 0}, {x, y}]** $\Rightarrow \mathbf{x == -1 \,||\, y == 1.}$

Функция **FindRoot** служит для нахождения одного приближенного корня уравнения. Формат функции: **FindRoot[уравнение, {x, x0}]**, где **x** – неизвестное, **x0** – ближайшее приближенное значение неизвестного.

Найти корень уравнения: $\sin(x) = x^2$, ближайший к значению x0 = 0,9.

Решение: $x \approx 0{,}876726$.

Решение в системе **Mathematica**: **FindRoot** [**Sin**[x] $==$ $\mathbf{x^2}$, **{x, 0.9}]** $\Rightarrow$ {x$\rightarrow$ $\mathbf{0.876726}$}.

Решение систем линейных уравнений

1) **LinearSolve**[**m,b**] – выдаёт вектор $\mathbf{x^0} \Rightarrow \{\mathbf{x_1^0}, \mathbf{x_2^0}, \dots, \mathbf{x_n^0}\}$, являющийся решением матричного уравнения **m.x = b** (**m** – квадратная матрица **n**– го порядка системы **n** линейных уравнений с **n** неизвестными, **b** – матрица-столбец правых частей системы уравнений).

2) **RowReduce**[$\mathbf{m_1}$] – решить систему методом Гаусса, где $\mathbf{m_1} = (\mathbf{m}|\mathbf{b})$ – расширенная матрица системы. Для системы двух линейных уравнений с двумя неизвестными решение выглядит так: $\mathbf{x^0} \Rightarrow \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{x_1^0} \\ \mathbf{0} & \mathbf{1} & \mathbf{x_2^0} \end{pmatrix}$.

3) **Solve[**{список уравнений}, {список неизвестных}**]** – решить систему**n**линейных уравнений с **n**неизвестными (см. функцию **Solve**, рассмотренную в п. 1). Для системы двух линейных уравнений с двумя неизвестными решение выглядит так: $\mathbf{x^0} \Rightarrow \{\{\mathbf{x1} \to \mathbf{x_1^0}\},\ \{\mathbf{x2} \to \mathbf{x_2^0}\}\}$.

4) Пример: решить в **Mathematica** систему двух уравнений с двумя неизвестными $\begin{cases} x_1 - 2x_2 = 3 \\ 2x_1 + 3x_2 = -1 \end{cases}$ (решение: $x_1 = 1, x_2 = -1$) с помощью рассмотренных выше функций.

1) $m = \{\{1,-2\},\{2,3\}\};\ b = \{3,-1\};$ **LinearSolve**[**m,b**] $\Rightarrow \{\mathbf{1}, \mathbf{-1}\}$;

2) $m_1 = \begin{pmatrix} 1 & -2 & 3 \\ 2 & 3 & -1 \end{pmatrix}$; $\mathbf{RowReduce}[\mathbf{m_1}] \Rightarrow$ $\{\{\mathbf{1}, \mathbf{0}, \mathbf{1}\}, \{\mathbf{0}, \mathbf{1}, \mathbf{-1}\}\} \Rightarrow \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{-1} \end{pmatrix}$;

3) **Solve**$[\{\mathbf{x1 - 2x2 == 3, 2x1 + 3x2 = -1}\}, \{\mathbf{x1, x2}\}] \Rightarrow$ $\{\{\mathbf{x1 \to 1}\}, \{\mathbf{x2 \to -1}\}\}$.

7.8. Операции математического анализа.

Вычисление пределов

Формат функции взятия предела от функции $\mathbf{y = f(x)}$ при $\mathbf{x \to x_0}$ $\left(\mathbf{lim}_{\mathbf{x \to x_0}} f(\mathbf{x})\right)$

$\mathbf{Limit}$[выражение, $\mathbf{x \to x_0}$].

Например, $\mathbf{Limit}\left[\frac{\mathbf{Sin[x]}}{\mathbf{x}}, \mathbf{x \to 0}\right] \Rightarrow \{\mathbf{1}\}$

Функция **Limit** в качестве необязательных аргументов допускает две опции: **Direction** → **Automatic** (или –1 (*предел справа*), или +1(*предел слева*)) и **Analytic** → **False** (или **True**) задаёт режим обработки функций, из которых составлено выражение. В случае **True** функции раскладываются в ряд Тейлора. Если в системе **Mathematica** невозможно вычислить предел, то возвращается невычисленное выражение.

$\mathbf{Limit}\left[\mathbf{ArcTan}\left[\frac{\mathbf{1}}{\mathbf{x}}\right], \mathbf{x \to -0, Direction \to +1}\right] \Rightarrow -\frac{\boldsymbol{\pi}}{\mathbf{2}}$;

$\mathbf{Limit}\left[\mathbf{ArcTan}\left[\frac{1}{x}\right], \mathbf{x} \to -\mathbf{0}\right] \Rightarrow \frac{\pi}{2};$

Дифференцирование функций

Операции дифференцирования в системе **Mathematica** осуществляют две функции:

D – частного дифференцирования и **Dt** - полного дифференцирования.

$\mathbf{D}[\,f[\mathbf{x}], \mathbf{x}]$ – частная производная $\frac{\partial f}{\partial \mathbf{x}}$; или $\boldsymbol{\partial}_{\mathrm{x}}\, f[\mathbf{x}]$;

$\mathbf{D}[f[\mathbf{x}], \{\mathbf{x}, \mathbf{n}\}]$ – частная производная **n** – го порядка $\frac{\partial^{\mathbf{n}} f}{\partial \mathbf{x}^{\mathbf{n}}}$;

$\mathbf{D}[f[\mathbf{x_1}, \mathbf{x_2}, \dots], \mathbf{x_1}, \mathbf{x_2}, \dots]$–смешанная частная производная $\frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}\dots f(x_1, x_2, \dots, x_n)$;

$\mathbf{Dt}[f]$– полный дифференциал $\boldsymbol{\partial} f$;

$\mathbf{Dt}[f, \mathbf{x}]$ – полная производная $\frac{\partial f}{\partial \mathrm{x}}$;

$\mathbf{Dt}[f, \mathbf{x_1}, \mathbf{x_2} \dots]$ – смешанная полная производная $\frac{\partial}{\partial \mathbf{x_1}}\frac{\partial}{\partial \mathbf{x_2}} \dots f$.

$\mathbf{Dt}[f, \{\mathbf{x}, \mathbf{n}\}]$ – полная производная **n** – го порядка. Вместо $\mathbf{Dt}[f, \mathbf{x}]$ и $\mathbf{Dt}[f, \mathbf{x_1}, \mathbf{x_2} \dots]$ можно использовать шаблоны $\boldsymbol{\partial}_{\square}\square$ и $\boldsymbol{\partial}_{\square,\square}\square$ на палитре **BASICINPUT**.

Найти, например, производные:

$\mathbf{D}\left[\sqrt{\mathbf{x}}, \mathbf{x}\right]$или $\boldsymbol{\partial}_{\mathbf{x}}\sqrt{\mathbf{x}} \Rightarrow \frac{1}{2\sqrt{\mathbf{x}}}$; $\mathbf{D}[\mathbf{x}^2 - 3\mathbf{x} + 5, \mathbf{x}]$ или $\boldsymbol{\partial}_{\mathbf{x}}(\mathbf{x}^2 - 3\mathbf{x} + 5) \Rightarrow 2\mathbf{x} - 3.$

Найти частные производные: $\mathbf{D}[\mathbf{x}^2 + 2\mathbf{xy} + \mathbf{y}^2, \mathbf{x}]$ или $\boldsymbol{\partial}_{\mathbf{x}}(\mathbf{x}^2 + 2\mathbf{xy} + \mathbf{y}^2) \Rightarrow 2\mathbf{x} + 2\mathbf{y}.$

Смешанные частные производные: $\mathbf{D}[\mathbf{x}^2 - 3\mathbf{xy}^2 + \mathbf{y}^3, \mathbf{x}, \mathbf{y}]$ или $\boldsymbol{\partial}_{\mathbf{x},\mathbf{y}}(\mathbf{x}^2 - 3\mathbf{xy}^2 + \mathbf{y}^3) \Rightarrow -6\mathbf{y}.$

Полный дифференциал 1−го порядка: $\mathbf{Dt}[\mathbf{x}^2 + 6\mathbf{xy} + 4\mathbf{y}^2] \Rightarrow 2\mathbf{x} * \mathbf{Dt}[\mathrm{x}] + 6\mathbf{y} * \mathbf{Dt}[\mathrm{x}] + 6\mathbf{x} * \mathbf{Dt}[\mathrm{y}] + 8\mathbf{y} * \mathbf{Dt}[\mathrm{y}]$
$(\,\boldsymbol{\partial}(\mathbf{x}^2 + 6\mathbf{xy} + 4\mathbf{y}^2) = 2\mathbf{x}\partial\mathbf{x} + 6\mathbf{y}\partial\mathbf{x} + 6\mathbf{x}\partial\mathbf{y} + 8\mathbf{y}\partial\mathbf{y}\,)$;

Найти дифференциал от функции $\mathbf{sin}\, \boldsymbol{x}$:

$\mathbf{Dt}[\mathbf{Sin}[\mathrm{x}]] \Rightarrow \mathbf{Cos}[\mathrm{x}] * \mathbf{Dt}[\mathrm{x}].$

$\int f(\mathbf{x})\mathbf{dx} = \mathbf{F}(\mathbf{x}) + \mathbf{C}$– неопределенный интеграл.

$\int_{\mathbf{a}}^{\mathbf{b}} f(\mathbf{x})\mathbf{dx} = \mathbf{F}(\mathbf{b}) - \mathbf{F}(\mathbf{a})$– определенный интеграл.

Integrate$[f, \mathbf{x}]$ или $\int f[\mathbf{x}]\mathbf{dx}$ – вычисление неопределенного интеграла в системе **Mathematica**. Следует иметь в виду, что при этом выдается лишь первообразная (в ответе отсутствует произвольная константа $\mathbf{C}$).

Integrate$[f, \{\mathbf{x}, \mathbf{a}, \mathbf{b}\}]$ – вычисление определенного интеграла в системе **Mathematica**.

Integrate$[f, \{\mathbf{x}, \mathbf{a}, \mathbf{b}\}, \{\mathbf{y}, \mathbf{c}, \mathbf{d}\}, \ldots]$ – кратный интеграл от функции f двух переменных $\mathbf{x}$ и $\mathbf{y}$.

Integrate$[\{f_{\mathbf{1}}, f_{\mathbf{2}}, \ldots, f_{\mathbf{n}}\}, \{\mathbf{x}, \mathbf{a}, \mathbf{b}\}]$– определенный интеграл от нескольких функций.

$\int_{\mathbf{a}}^{\mathbf{b}} f_{\mathbf{1}}\mathbf{dx}$; $\int_{\mathbf{a}}^{\mathbf{b}} f_{\mathbf{2}}\mathbf{dx}$; ... $\int_{\mathbf{a}}^{\mathbf{b}} f_{\mathbf{n}}\mathbf{dx}$; **a** и **b** могут быть равными $-\infty$ или $+\infty$.

Примеры вычисления неопределенных интегралов:

1) $\int \mathbf{x}^2\mathbf{dx} \Rightarrow \frac{\mathbf{x}^3}{\mathbf{3}}$; 2) **Integrate**$[\mathbf{Sin}[\mathbf{x}], \mathbf{x}] \Rightarrow -\mathbf{Cos}[\mathbf{x}]$.

Численное интегрирование в системе **Mathematica** осуществляется с помощью функции **NIntegrate**, имеющей формат **NIntegrate**$\left[f[\mathbf{x}], \{\mathbf{x}, \mathbf{x_0}, \mathbf{x}_k\}\right]$.

Пример: **NIntegrate**$\left[\sqrt{\mathbf{2x+1}}, \{\mathbf{x}, \mathbf{0}, \mathbf{1}\}\right] \Rightarrow \mathbf{1.3987}$.
Функция **NIntegrate** имеет ряд опций, о которых можно узнать с помощью функции **Options**[**NIntegrate**]или, обратившись к системе **Help**.

7.9. Графика

В системе **Mathematica** имеется десять встроенных функций: **Plot**, **ListPlot**, **ParametricPlot**, **ContourPlot**, **ListContourPlot**, **DensityPlot**, **ListDensityPlot**, **Plot3D**, **ListPlot3D**, **ParametricPlot3D**, предназначенных для построения графиков. Три из них, названия которых оканчиваются на **3D**, строят изображения графических объектов в трёхмерном пространстве, остальные дают графические объекты на плоскости.

Графические функции двумерной графики и их опции

Графическая функция **Plot**.

1-й формат: $\mathbf{Plot}[f[\mathbf{x}],\{\mathbf{x},\mathbf{a},\mathbf{b}\},$ список опций$]$ –печать графика одной функции. Например, функция $\mathbf{Plot}[\boldsymbol{S}\mathbf{in}[\mathbf{x}],\{\mathbf{x},\mathbf{0},\mathbf{2\pi}\}]$ строит график функции $\boldsymbol{S}\mathbf{in}[\mathbf{x}]$ на периоде [0, 2$\boldsymbol{\pi}$].

Можно графики именовать:

например, $\mathbf{g}=\mathbf{Plot}[\mathbf{Sin}[\mathbf{x}],\{\mathbf{x},\mathbf{0},\mathbf{2\pi}\}]$ – задает графический объект **g**.

2-й формат: $\mathbf{Plot}[\{f_1[\mathbf{x}],f_2[\mathbf{x}],\ldots,f_n[\mathbf{x}]\},\{\mathbf{x},\mathbf{a},\mathbf{b}\},$ список опций$]$– печать нескольких графиков, заданных на промежутке [**a,b**].

Пример: Построить три синусоиды $\mathbf{Sin\,x},\mathbf{Sin2x},\mathbf{Sin3x}$ на промежутке[0, 2$\boldsymbol{\pi}$].

Решение:$\mathbf{Plot}[\{\mathbf{Sin}[\mathbf{x}],\mathbf{Sin}[\mathbf{2x}],\mathbf{Sin}[\mathbf{3x}]\},\{\mathbf{x},\mathbf{0},\mathbf{2\pi}\}]$

Видоизменение графиков и их комбинирование

$\mathbf{Show}[\mathbf{g}]$–изображение графика по вычисленным данным;

$\mathbf{Show}[\mathbf{g},\mathbf{option}\rightarrow\mathbf{value}]$– изображение графика по вычисленнымданным с использованием опций;

$\mathbf{Show}[\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_n}]$ – изображение **n** графиков;

$\mathbf{GraphicsArray}[\{\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_n}\}]$ – представляет ряд вычисленных графических объектовбез их изображения;

$\mathbf{Show}\left[\mathbf{GraphicsArray}[\{\mathbf{g_1},\mathbf{g_2},\ldots,\mathbf{g_n}\}]\right]$ – изображение нескольких графиков на одной горизонтальной линии. Пример:

$$\begin{aligned}\mathbf{g1}=\mathbf{Plot}[\mathbf{Cosh}[x],\{x,-\mathbf{3},\mathbf{3}\},\mathbf{AspectRatio}\\ \rightarrow\mathbf{Automatic},\mathbf{PlotRange}\rightarrow\{\mathbf{0},\mathbf{5}\},\mathbf{PlotStyle}\\ \rightarrow\{\mathbf{Hue}[\mathbf{0.55}],\mathbf{Thickness}[\mathbf{0.012}]\}]\end{aligned}$$

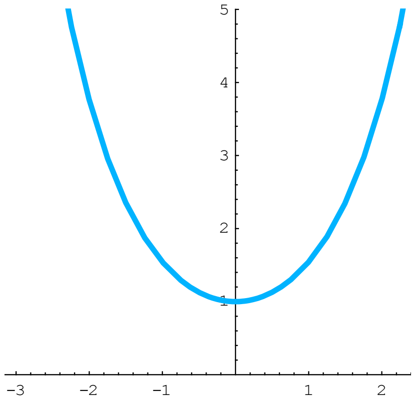


Рис. 9. Первый графический объект **g1**.

g2=ParametricPlot[{Cos[t] + Log[Abs[Tan[t/2]]], Sin[t]}, {t, 0, π}, AspectRatio→Automatic, PlotRange→{{–3,3}, Automatic}, Ticks→{{–2,2}, {1}}, PlotStyle→{Hue[0.85], Thickness[0.012]}]

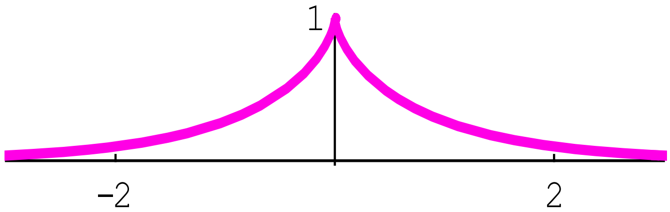


Рис. 10. Второй графический объект **g2**.

1) **Show**[**g1**, **g2**]⇒

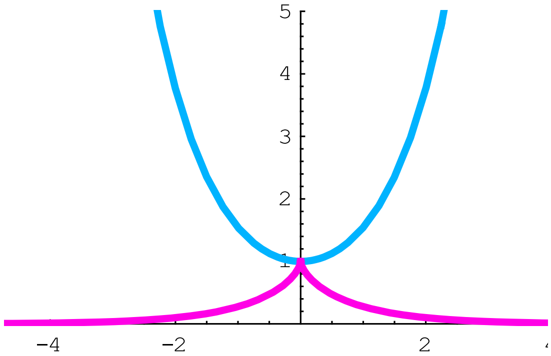
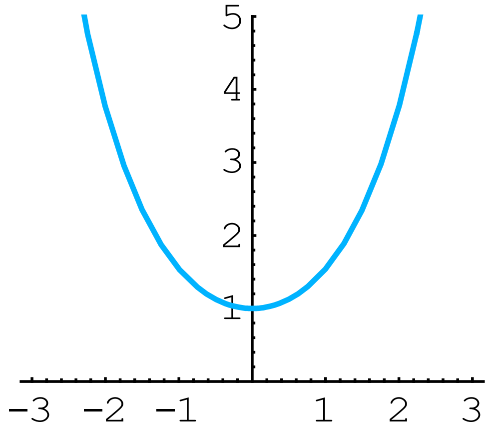


Рис. 11. Совмещение графических объектов **g1** и **g2**.

2) **Show**[**GraphicsArray**[{**g1**, **g2**}]]⇒



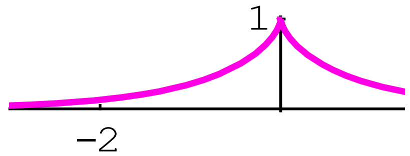


Рис. 12. Последовательное расположение графических объектов **g1** и **g2**.

Графические функции трехмерной графики

Plot3D – строит поверхность, уравнение которой в декартовой системе координат $z = f(x, y)$.

ParametricPlot3D – изображает пространственную кривую в пространстве.

ListPlot3D– изображение поверхности, заданной массивом чисел.

7.10. Решение дифференциальных уравнений

С помощью системы **Mathematica** можно находить аналитические и численные решения обыкновенных дифференциальных уравнений, а также решения дифференциальных уравнений в частных производных.

Решение дифференциальных уравнений в символьном виде

Для решения обыкновенных дифференциальных уравнений в символьном виде в системе **Mathematica** используются следующие средства:

1) Пусть дано обыкновенное дифференциальное уравнение n-го порядка вида: $\mathbf{F(x, y, y', y'', \ldots , y^{(n)}) = 0}$.

Формат решения такого дифференциального уравнения в системе **Mathematica**: **DSolve**[**д.у.**, **y[x]**, **x]**, где **д.у.** – дифференциальное уравнение, записанное в формате системы **Mathematica**, **y[x]** – искомая функция, **x**– независимая переменная.

2) Если дана система обыкновенных дифференциальных уравнений относительно неизвестных функций **y1[x], y2[x], …, yn[x]**, каждая из которых – функция одной независимой переменной **x**, то формат решения такой системы: **DSolve[{д.у.1, д.у.2, …, д.у.n},{y1[x], y2[x], …, yn[x]}, x],**

где **д.у.*i*** – уравнения системы дифференциальных уравнений, **y*i* [x]** – искомые функции независимой переменной **x** (*i*=1, 2, …, n).

В решении дифференциальных уравнений встречаются *постоянные интегрирования*. По умолчанию они обозначаются как **C[*i*]**.

Пример: найти общее решение линейного однородного дифференциального уравнения с постоянными коэффициентами **y''–y'– 6y = 0**. Общим решением такого дифференциального уравнения является $y = c_1e^{-2x} + c_2e^{3x}$. В системе **Mathematica** уравнение решается так:

DSolve[y''[x] –y' [x] – 6y[x] == 0, y[x], x]

$\Rightarrow$ {{y[x]→e^{-2x}C[1] + e^{3x}C[2]}}

3) Решение задачи Коши для одного дифференциального уравнения относительно неизвестной функции **y[x]** с начальным условием **y[x0]=y0**. Формат решения такой задачи: **DSolve[{д.у., н.у.},y[x], x],** где **д.у.** – дифференциальное уравнение, **н.у.**–начальное условия, **y[x]** – искомая функция от одной независимой переменной **x**.

4) Решение задачи Коши для системы дифференциальных уравнений относительно неизвестных функций **y1[x], y2[x], …, yn[x]** имеет формат: **DSolve[{д.у.1, д.у.2, …, д.у.n, н.у.1, н.у.2 , …, н.у.n},{y1[x], y2[x], …, yn[x]}, x],** где **д.у.*i*** – уравнения системы дифференциальных уравнений, соответствующие **н.у.*i*** – начальные условия, **y*i* [x]** – искомые функции от одной незави-

симой переменной **x** (i=1, 2, …, n).

Решение дифференциальных уравнений
в численном виде

Многие дифференциальные уравнения не имеют аналитических решений. Однако они могут решаться численными методами. Для численного решения систем дифференциальных уравнений используется функция **NDSolve**:

1) **NDSolve[д.у., y, {x, xmin, xmax}]**
ищет численное решение дифференциального уравнения **д.у.** относительно функции **y** независимой переменной **x** на промежутке от **xmin** до **xmax**.

2) **NDSolve[с.д.у., {y1,y2, …, yn}, {x, xmin, xmax}]**
ищет численное решение системы дифференциальных уравнений с.**д.у.** относительно функций **y*i*** независимой переменной **x** на промежутке от **xmin** до **xmax**.

Визуализация частных решений
дифференциальных уравнений

Часто желательно выводить результаты аналитического или численного решения дифференциальных уравнений в графическом виде.

На следующих примерах показано как это делается.

1) **ds1=DSolve[{y''[x] –8 y'[x]+15 y[x] == 0,y[0]==1, y'[0] == –2}, y[x], x]** ⇒

{{y[x]→ – (1/2) e^{3x} (–7+5 e^{2x})}}; **Plot[y[x]/.ds1, {x, 0, 0.4}]**

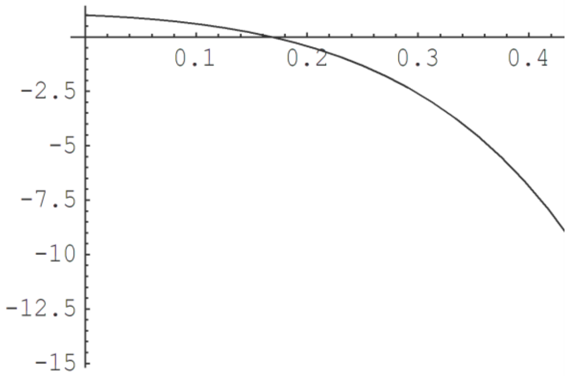


Рис. 13. График интегральной кривой (решения задачи Коши **ds1**)

2)**ds2=NDSolve[{x'[t]==y[t],y'[t]==-0.01*y[t]–Sin[x[t]],x[0]==0, y[0]==2.1},{x,y},{t,0,100}]**⇒ **{{x→InterpolatingFunction[{{0.,100.}},<>], y→InterpolatingFunction[{{0.,100.}},<>]}}**;

ParametricPlot[{x[t],y[t]}/.ds2,{t,0,100},PlotPoints→10000, PlotStyle→{RGBColor[0,0,1]}]

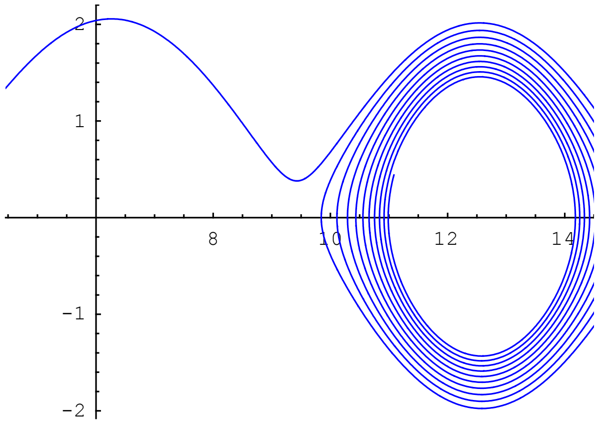


Рис. 14. График интегральной кривой
(численного решения задачи Коши **ds2**)

7.11. Решение оптимизационных задач

Поиск максимального и минимального чисел в списке

Для поиска максимального и минимального значений ряда чисел, входящих в список, в системе **Mathematica** имеются следующие средства:

Max[x1, x2, …] – возвращает наибольшее значение из **x*i***;

Max[{x1, x2, …},{y1, y2, …}, …] – возвращает наибольший элемент из нескольких списков;

Min[x1, x2, …] – возвращает наименьшее значение из **x*i***;

Min[{x1, x2, …},{y1, y2, …}, …] – возвращает наименьший элемент из нескольких списков.

Поиск локального экстремума аналитической функции

Для численного нахождения локального минимума аналитической функции используется функция **FindMinimum[f, {x, x0}]**, которая выполняет поиск локального минимума функции **f**,начиная со значения **x=x0**, и возвращает его значение.

Пример: **FindMinimum[–x*Exp[–2 x], {x, 1}]** ⇒**{ – 0.18394, {x→0.5}}**

Аналогично находится локальный максимум аналитической функции с помощью функции **FindMaxmum[f, {x, x0}]**, которая выполняет поиск локального максимума функции **f**,начиная со значения **x=x0**, и возвращает его значение.

Пример: **FindMaxmum[x^3 - 4x^2 – x + 1, {x, –1}]**⇒**{1.0607, {x→–0.11963}}**

Поиск глобального экстремума аналитической функции

Следующие две функции служат для поиска глобального минимума и максимума аналитически заданной функции:

ConstrainedMin[f, {система неравенств}, {x, y, …}] – ищет глобальный минимум функции **f** в области, определяемой заданной системой неравенств. При этом предполагается, что все переменные **x**, **y**, неотрицательны.

Minimize[f, {система неравенств}, {x, y, …}] – ищет глобальный минимум функции**f** в области, определяемой заданной системой неравенств. При этом предполагается, что все переменные **x**, **y**, неотрицательны. Аналогично находится максимум с помощью функции **Maximize**.

Решение задач линейного программирования

При решении задач линейного программирования может использоваться функция **LinearProgramming[c, m, b],** которая ищет вектор **x**, минимизирующий величину **c.x**в соответствии с условиями **m.x≥ b** и **x≥ 0**.

Пример. Решить двумерную задачу линейного программирования в системе Mathematicac использованием функций **LinearProgramming** и **Minimize**.

```
z = -3 x - 2 y
```

```
-3 x - 2 y
```

```
c = {-3, -2}
```

```
{-3, -2}
```

```
m = {{1, -1}, {-3, 2}, {2, 1}, {0, -1}}
```

```
{{1, -1}, {-3, 2}, {2, 1}, {0, -1}}
```

```
b = {-2, -6, 2, -3}
```

```
{-2, -6, 2, -3}
```

```
LinearProgramming[c, m, b]
```

```
{4, 3}
```

```
x0 = 4
```

```
4
```

```
y0 = 3
```

```
3
```

```
z0 = -3 x0 - 2 y0
```

```
-18
```

Другой способ

```
Minimize[-3 x - 2 y, {x - y ≥ -2, -3 x + 2 y ≥ -6, 2 x + y ≥ 2, -y ≥ -3}, {x, y}]
```

```
{-18, {x → 4, y → 3}}
```

Заключение

Данное учебное пособие реализовано как введение в математическое и компьютерное моделирование – важнейшие разделы прикладной математики, применительно к задачам, решаемым в землеустройстве. Существующий аппарат современной математики, мощные средства вычислительной техники, развитые компьютерные технологии обработки информации позволяют успешно решать любые практические задачи, стоящие перед обществом.

При подготовке пособия ставились цели: формирование системы компетенций и математической культуры, необходимых для успешного решения в будущем профессиональных и общественных задач обучающимися по направлению подготовки 21.04.02 Землеустройство и кадастры, в том числе задач оптимизации в области землеустройства и кадастров, приобретение навыков применения методов прикладной математики и средств компьютерного моделирования при решении практических задач, а также приобретение общих знаний и умений в области математического моделирования и мотивации к самообразованию.

Тема моделирования и разработки математических моделей процессов, явлений, объектов, систем весьма многогранна. В данном учебном пособии рассмотрены лишь некоторые основные формы математических моделей, пути разработки, исследования и расчета этих моделей.

Список использованной литературы

1. Адамар Ж. Четыре лекции по математике. Москва : Институт компьютерных исследований. 2002. 60 с.
2. Аксенов Е.П. Методы оптимальных решений. Учебное пособие. Пермь : ИПЦ «Прокростъ» ФГБОУ ВО Пермская ГСХА, 2016. 90 с.
3. Андрейченко К. П., Андрейченко Д. К. Математическое моделирование динамических систем: учебное пособие. Саратов : Изд-во Саратовского ГТУ, 2000. 140 с.
4. Андриевский Б.Р., Фрадков А.Л. Элементы математического моделирования в программных средах MATLAB и Scilab. СПб. : Наука, 2001. 286 с.
5. Арнольд В.И. «Жесткие» и «мягкие» математические модели. Москва : Изд.-во МЦНМО. 2000. 32 с.
6. Арнольд В.И. Что такое математика? Москва : Изд-во МЦНМО. 2008. 104 с.
7. Асанов А.З. Введение в математическое моделирование динамических систем. Казань : Изд-во Казанского гос. университета. 2007. 205 с.
8. Аюпов В.В. Исследование маневренных свойств автопоездов на основе системного подхода : монография. Пермь : ИПЦ «Прокростъ» ФГБОУ ВПО Пермская ГСХА, 2012. 96 с.
9. Аюпов В.В. Лабораторный практикум по компьютерной математике. Пермь : ИПЦ «Прокростъ» ФГБОУ ВО Пермская ГСХА, 2015. 60 с.
10. Аюпов В.В. Математическое моделирование технических систем : учебное пособие. Пермь : ИПЦ «Прокростъ» ФГБОУ ВО Пермская ГСХА, 2017. 242 с.
11. Аюпов В.В. Системно-структурный подход к преподаванию математики в вузе. // Научный журнал «Вестник Пермского университета. Математика. Механика. Информатика». Вып. 3. Пермь : ПГУ, 2010. С. 25-29.
12. Белов П.Г. Системный анализ и моделирование процессов в техносфере. Москва : 2001. 512 с.

13. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем. /СПб .: БХВ-Петербург, 2002. 464 с.

14. Берталанфи Л. фон. Общая теория систем – обзор проблем и результатов // Системные исследования: ник. Москва : Наука, 1969. С. 30-54.

15. Бир Ст. Кибернетика и управление производством. Пер. с англ. В. Я. Алтаева. Москва : Наука, 1963. 276 с.

16. Блехман И.И., Мышкис А.Д., Пановко Я.Г. Прикладная математика. Предмет, логика, особенности подходов. Киев. : Наукова думка. 1976. 270 с.

17. Болдин А.П. Основы научных исследований. Москва. 2012. 336 с.

18. Бурбаки Н. Архитектура математики. Очерки по истории математики / Перевод И. Г. Башмаковой, под ред. К. А. Рыбникова. Москва : ИЛ, 1963. 258 с.

19. Бурсиан Э.В. Задачи по физике для компьютера : учебное пособие для студентов физ.-мат. фак. пед. ин-тов. Москва : Просвещение, 1991. 256 с.

20. Валге А.М. Обработка экспериментальных данных и моделирование динамических систем при проведении исследований по механизации сельскохозяйственного производства. СПб. : 2002.179 с.

21. Введение в математическое моделирование : учебное пособие для вузов / В. Н. Ашихмин и др.; под ред. П. В. Трусова. Москва : Логос, 2005. 440 с.

22. Введение в математическое моделирование : учебное пособие / В.Н. Ашихмин и др.; под ред. П.В. Трусова. Москва : Интермет Инжиниринг, 2000. 336 с.

23. Волков С.Н. Экономико-математические методы и модели в землеустройстве : учебное пособие. Москва : Колос. 2007. 696 с.

24. Вольтерра В. Математическая теория борьбы за существование. Москва : Наука, 1976. 286 с.

25. Высшая математика для экономистов. под ред. Кремера Н.Ш. Москва: 2007. 479 с.

26. Голубева Н. В. Математическое моделирование систем и процессов. Изд-во : "Лань", 2013. 192 с.

27. Грэхем Р., Кнут Д., Поташник О. Конкретная математика. Основание информатики. Москва : Мир, 1998. 784 с.

28. Гулд Х., Табочник Я. Компьютерное моделирование в физике. в 2-х частях; пер. с англ. Москва : Мир, 1990. 352 с.

29. Давенпорт Дж., Сирэ И., Турнье Э. Компьютерная алгебра. Системы и алгоритмы алгебраических вычислений. Москва : Мир, 1991. 352 с.

30. Декарт Р. Правила для руководства ума. Сочинения. / Москва : Мысль. 1989. Т. 1. 656 с.

31. Дьяконов В. П. MATLAB6/6.1/6.5.Simulink4.5. Основы применения: Полное руководство пользователя. Москва : Солон-Пресс, 2002. 768 с.

32. Дьяконов В. П. Вейвлеты. От теории к практике. Москва : Солон-Р, 2002. 448 с.

33. Дьяконов В. П. Система компьютерной алгебры DERIVE : самоучитель. Полное руководство пользователя. Москва : Солон-Р, 2002. 320 с.

34. Дьяконов В. П. Системы символьной математики Mathematica 2 и Mathematica 3. М.осква : СК-Пресс, 1998. 318 с.

35. Дьяконов В. П., Абраменкова И. В. MATLAB. Обработка сигналов и изображений. СПб .: Питер, 2002. 608 с.

36. Дьяконов В.П., Абраменкова И.В., Пеньков А.А. Новые информационные технологии : учеб. пособие / под ред. В.П. Дьяконова; Смол. гос. пед. ун-т. Ч. 3: Основы математики и математическое моделирование. Смоленск 2003. 192 с.

37. Дьяконов В.П. Maple7 : учебный курс. СПб. : Питер, 2002. 672 с.

38. Дьяконов В.П. MathCAD 2001 : специальный справочник. /СПб .: Питер, 2002. 592 с.

39. Дьяконов В.П. Mathematica4 : учебный курс. СПб. : Питер, 2001.

40. Дьяконов В.П. Компьютерная математика. Теория и практика. Москва : Нолидж, 2001. 1296 с.

41. Ерусалимский Я.М. Дискретная математика. Теория, задачи, приложения. / Москва : Вузовская книга. 2009. 288 с.

42. Зельдович Я.Б., Мышкис А.Д. Элементы прикладной математики. 2002. 592 с.

43. Ильин В. А., Позняк Э. Г. Основы математического анализа. Москва : Физматлит, 2001. Ч. I. 648 с.

44. Ильин В. А., Позняк Э. Г. Основы математического анализа. / Москва : Физматлит, 2001. Ч. II. 464 с.

45. Интрилигатор М. Математические методы оптимизации и экономическая теория. 1975. 607 с.

46. Калман Р., Фалб П., Арбиб М. Очерки по математической теории систем. Москва : URSS. 2010.400 с.

47. Канторович Л.В. Математические методы организации и планирования производства. /Ленинград : Изд-во ЛГУ, 1939. 68 с.

48. Колмогоров А. Н. Математика – наука и профессия. Москва : Наука, 1988. 288 с.

49. Красс М.С., Чупрынов Б.П. Математика для экономистов. СПб. : Питер, 2005. 46 4с.

50. Кук Д., Бейз Г. Компьютерная математика. Москва : Наука. 1990. 384 с.

51. Купманс Т. Три эссе о состоянии экономической науки. Нью-Йорк. 1957. 365 с.

52. Лобачевский Н.И. Избранные труды по геометрии : серия Классики науки. Москва : Изд-во Академии Наук СССР, 1956. 596 с.

53. Лутманов С.В., Аюпов В.В., Гамилова Л.В. Задачи оптимизации в конечномерных пространствах :учебное пособие. Пермск. ун-т . Пермь, 2007. 160 с.

54. Максимей И.В. Имитационное моделирование на ЭВМ Москва : Радио и связь, 1988. 232 с.

55. Маркс К., Энгельс Ф. Сочинения. / 2-е изд. Т. 20. С. 37.
56. Марчук Г. И. Методы вычислительной математики. Москва : Наука, 1989. Т. 1. 456 с.
57. Математика. Ее содержание, методы и значение. / под ред. Александрова А.Д., Колмогорова А.Н., Лаврентьева М.А. Москва : Изд-во Академии наук СССР. 1956. 296 с.
58. Математический энциклопедический словарь / под ред. Ю. В. Прохорова. Москва : Советская энциклопедия, 1988. 847 с.
59. Месарович М., Такахара Я. Общая теория систем: математические основы / под ред. С. В. Емельянова. Москва : Мир, 1978. 312 с.
60. Мещерский И.В. Сборник задач по теоретической механике. Москва : Наука, 1986. 448 с.
61. Митропольский Ю. А. Проблемы асимптотической теории нестационарных колебаний. Москва : Наука, 1964.431 с.
62. Моисеев Н.Н. Математика ставит эксперимент. Москва : Наука. 1979. 224 с.
63. Мышкис А.Д. Элементы теории математических моделей. Москва : ЛЕНАНД. 2016. 200 с.
64. Налимов В.В. Теория эксперимента. Москва : Наука, 1971. 208 с.
65. Ногин В.Д. Методы оптимальных решений. СПб. : ЮТАС.2006. 108 с.
66. Разумовский О.С. Закономерности оптимизации в науке и практике. Москва : Наука, 1990. 176 с.
67. Реньи А. Трилогия о математике. Москва : Мир. 1980. 376 с.
68. Рузавин Г.И. Философские проблемы оснований математики. Москва : Наука, 1983. 304 с.
69. Рыбников К.А. История математики. Москва : Изд-во МГУ. 1974. 456 с.
70. Самарский А.А., Михайлов А.П. Компьютеры и жизнь (Математическое моделирование). Москва : Педагогика, 1987. 128 с.

71. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. М осква: Физматлит, 2005. 320 с.

72. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. Изд. 2. М. : Физматлит, 2001. 320 с.

73. Светлов В.А. Философия математики. Основные программы обоснования математики XX столетия.   Москва : ЛЕНАНД, 2016. 208 с.

74. Сдвижков О.А. MathCAD-2000: Введение в компьютерную математику. Москва : Издательско-торговая корпорация «Дашков и К», 2002. 204 с.

75. Семененко М.Г. Введение в математическое моделирование. / Москва : Солон-Р, 2002. 112 с.

76. Советов Б. Я., Яковлев С. А. Моделирование систем : учебное пособие. Изд-во : Высшая школа. 2001. 343 с.

77. Соколов А.В., Токарев В.В. Методы оптимальных решений. Общие положения. Математическое ние. Москва : Физматлит, 2011. Т. 1. 564 с.

78. Соколов А.В., Токарев В.В. Методы оптимальных решений. Том 2. Многокритериальность. Динамика. Неопределенность. Москва : Физматлит, 2012. 420 с.

79. Столниц Э., Дероуз Т., Салезин Д. Вейвлеты в компьютерной графике. Теория и приложения / пер. с англ. Ижевск, 2002. 272 с.

80. Тарасик В. П. Математическое моделирование технических систем: учебник для вузов.. Минск : Дизайн ПРО, 2004. 640 с.

81. Титов К.В. Компьютерная математика : учебное пособие. Москва : Инфра-М. 2016. 261 с.

82. Тихонов А.Н., Костомаров Д.П. Рассказы о прикладной математике. Москва : Физматлит, 1979.208 с.

83. Трифонова М.Ф., Заика П.М., Устюжанин А.П. Основы научных исследований : учебник. Москва : Колос. 1993. 240 с.

84. Турецкий В.Я. Математика и информатика : учебник. Москва : Инфра-М, 2000. – 560 с.

85. Усольцев Л.А. Прикладная математика : учебное пособие. Омск : Изд-во СибАДИ. 2008. 68 с.

86. Фёрстер Э., Рёнц Б. Методы корреляционного и регрессионного анализа. Москва : Финансы и статистика, 1981. 302 с.

87. Фрейденталь Г. Математика в науке и вокруг нас. Москва : Мир. 1977. 262 с.

88. Хартман Г. Современный факторный анализ. Москва : Статистика, 1972. 486 с.

89. Цисарь И.Ф., Крыкин М.А. Matlab_Simulink – лаборатория экономиста : учебное пособие. Москва : Изд-во «Анкил», 2001. 104 с.

90. Цисарь И.Ф., Нейман В.Г. Компьютерное моделирование экономики. Москва : Диалог-МИФИ, 2002. 304 с.

91. Шелобаев С.И. Математические методы и модели в экономике, финансах, бизнесе : учебное пособие для вузов. М. : ЮНИТИ-ДАНА, 2001. 376 с.

92. Шеннон Р. Имитационное моделирование систем. Искусство и наука. Москва : Мир, 1978. 417 с.

93. Шредер М. Фракталы, хаос, степенные законы. Миниатюры из бесконечного рая. / пер. с англ. Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001. 568 с.

94. Экономико-математические методы и прикладные модели : учебное пособие для вузов / В.В. Федосеев, [и др.]; под ред. В.В. Федосеева. Москва : ЮНИТИ, 2001. 391 с.

95. Яблонский С.В. Введение в дискретную математику. Москва : Высш. шк., 2003. 384 с.

96. Яглом И.М. Математические структуры и математическое моделирование. Москва : Сов.радио, 1980. 144 с.

Учебное издание

Аюпов Васыл Вафович,
Аюпов Александр Васылович

Прикладная математика

Учебное пособие
Редактор Е.А. Граевская
Подписано в печать 18. 10. 2017. Формат 60×84 $^{1}/_{16.}$
Усл. печ. л. 9,19. Тираж 50 экз. Заказ №132

ИПЦ «Прокростъ»
Пермской государственной сельскохозяйственной академии
имени академика Д.Н. Прянишникова,
614090, Россия, г. Пермь, ул. Петропавловская, 23 тел. (342) 210-35-34