

Белорусский национальный технический университет
Факультет информационных технологий и робототехники
Кафедра программного обеспечения информационных систем и технологий

СОГЛАСОВАНО
Заведующий кафедрой
Полозков Ю.В.

СОГЛАСОВАНО
Заместитель декана ФИТР
Бадак Б.А.

Электронный учебно-методический комплекс

КОМПЬЮТЕРНЫЕ МЕТОДЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

для специальности
6-05-0611-01 "Информационные системы и технологии"
профилизация "Информационные системы и технологии в проектировании и
производстве"

Составитель: Напрасников В.В.

Рассмотрено и утверждено
на заседании совета ФИТР 30 апреля 2025 г.,
протокол N 10

Минск ◇ БНТУ ◇ 2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели ЭУМК: ЭУМК предназначен для изучения дисциплины «Компьютерное моделирование прикладных задач» для обучающихся I ступени получения высшего образования по специальности 1-40 01 01 «Программное обеспечение информационных технологий» и 1-40 05 01 «Информационные системы и технологии». ЭУМК содержит набор методических материалов по этой дисциплине.

Особенности структурирования и подачи учебного материала: ЭУМК состоит из четырех частей.

Теоретический раздел содержит краткие лекционные материалы, посвященные изложению в наглядном виде основных понятий и определений.

Практический раздел содержит задания для проведения лабораторных занятий, а также программные коды примеров, рассмотренных в теоретическом разделе. Данные программные коды могут быть использованы для выполнения индивидуальных заданий.

Раздел контроля знаний содержит вопросы для организации текущего контроля знаний.

Вспомогательный раздел содержит программу дисциплины, список рекомендуемой литературы.

Рекомендации по работе с ЭУМК:

- ЭУМК представлен .pdf-файлом;
- требования к системе: IBM PC-совместимый ПК стандартной конфигурации, Adobe Reader. Программа работает в среде Windows XX;
- открытие ЭУМК производится посредством запуска файлов с расширением .pdf – в Adobe Reader

Оглавление

1	ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	4
	Тема 1. Понятие о моделировании.....	4
	Тема 2. Начало работы в среде MATHCAD.....	9
	Тема 3. Численные методы решения нелинейных уравнений	44
	Тема 4. Численные методы решения систем линейных алгебраических уравнений	56
	Тема 6. Моделирование и методы типа Монте-Карло.....	124
	Тема 7. Метод наихудшего случая и статистический анализ	150
	Тема 8. Моделирование конкурирующих стратегий	159
	Тема 9. Моделирование систем массового обслуживания.....	174
	Тема 10. Методы приближения инженерных и экономических данных	192
2.	ПРАКТИЧЕСКИЙ РАЗДЕЛ.....	217
	Лабораторная работа Модели колебательных систем	217
	Лабораторная работа Модель «хищник жертва».....	218
	Лабораторная работа Случайные величины в моделировании.....	221
	Лабораторная работа Конкурирующие стратегии эксплуатации автобуса	224
3.	РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....	229
4.	ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ.....	230

1 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

Тема 1. Понятие о моделировании

Виды моделирования

В повседневной жизни и научных исследованиях человек желает предсказать поведение объектов или развитие явлений. Для этого **реальный объект или явление он заменяет искусственным объектом, который называется моделью**. Этот искусственный объект строится исследователем исходя из его представлений о поведении реального объекта и принятых им упрощениях. Под упрощением понимается отбрасывание несущественных свойств реальных объектов.

Не всегда представления человека о реальном объекте правильно отражают его свойства. Модель называется адекватной, если на её основе можно правильно предсказать интересующие исследователя свойства.

Виды моделирования:

- натурное;
- полунатурное;
- аналоговое;
- математическое.

Натурное моделирование. При таком моделировании объект воспроизводится с возможно большей точностью и к нему прикладываются внешние воздействия, по возможности точно воспроизводящие натуральные ситуации (например, краш-тест автомобиля). К достоинствам относится точность получаемых результатов. Недостатки: дороговизна проведения эксперимента, принципиальная невозможность в некоторых случаях выполнить натуральный эксперимент, иногда погрешности натурального эксперимента могут оказаться выше, чем при математическом моделировании, такие погрешности могут возникнуть за счет погрешности измерительных аппаратов.

Полунатурное моделирование. При этом изменяется либо масштаб объекта, либо материал, либо другое его важное свойство. По эксперименту, выполненному на макете, с помощью коэффициентов пересчета, воспроизводится реальная ситуация. Недостатки: сложность выполнения и однократность эксперимента, сложность измерений.

Аналоговое моделирование. На основе принципа единства мира явления, имеющие разную природу, описываются одинаковыми уравнениями. Исходя из этого, можно, решая одинаковые уравнения,

интерпретировать полученные решения для описания объекта другой природы. АВМ – аналоговые вычислительные машины – бывают:

- электрические;
- пневматические;
- гидравлические.

В электрических АВМ на верхней панели выгравированы символы электрических элементов, выполняющих различные математические операции (сложение, умножение, интегрирование, дифференцирование). Каждый элемент имеет один или несколько входов и один выход. Соединяя гибкими проводниками гнезда входов и выходов различных элементов, составляется электрическая схема, позволяющая решить соответствующее уравнение. После включения машины, решение мгновенно получается в виде сигнала на экране осциллографа. Недостатки: сложность получения коэффициентов для пересчета электрического сигнала к переменной реальной задачи.

Математическое моделирование. В этом виде моделирования в качестве искусственного объекта, заменяющего реальный объект (модели), выступает чаще всего уравнение или системы уравнений. Реже могут быть графы, матрицы, тензоры или другие математические объекты. Поэтому моделирование называется математическим. Процесс построения математической модели имеет множество этапов. Основные из них изображены на рисунке 1.

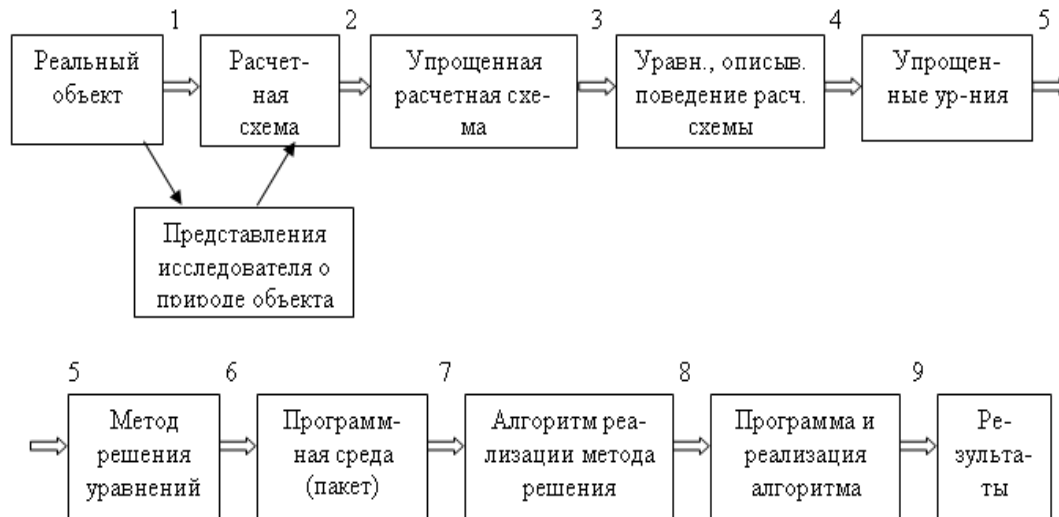


Рисунок 1.1. Этапы математического моделирования с использованием компьютера.

На этом этапе могут возникнуть гносеологические ошибки, связанные с неправильным представлением исследователя о функционировании объекта.

Чаще всего решить задачу в общем виде не удастся и приходится вводить дополнительные предположения, абстрагируясь от несущественных свойств объекта (свойства, которые оказывают незначительное влияние на исследуемые свойства технического объекта). На этом этапе вопрос «существенные - не существенные» исследователь решает в зависимости от своего опыта и квалификации.

Когда упрощённая расчётная схема получена, нужно подобрать правильные уравнения, описывающие поведение сплошной среды или системы с сосредоточенными параметрами. Например, если рассматривать задачу о деформировании биметаллической пластины, то уравнения должны учитывать как упругие, так и температурные свойства объекта. Если та же пластина деформировалась лишь под влиянием внешних сил, которые медленно меняются во времени при постоянной температуре, то достаточно использовать уравнения обычной теории упругости.

При рассмотрении тонких эффектов, полученные уравнения могут оказаться очень сложными. Если же интересующая исследователя точность решения может быть достигнута при пренебрежении этими эффектами, то часто удаётся упростить эти уравнения. Например, одним из таких упрощений является линеаризация. При этом в решении удерживается первое линейное слагаемое, а слагаемые более высоких порядков отбрасываются.

Одни и те же уравнения можно решить различными численными методами. Эти методы могут обладать различной точностью.

На следующем этапе подбирается нужная среда программирования или пакет, обладающие возможностями для решения соответствующих уравнений.

Теперь создаётся программная реализация, которая может содержать как синтаксические, так и семантические ошибки.

При создании программной реализации могут возникнуть погрешности, связанные с неправильной подготовкой входных и выходных параметров. Это в дальнейшем затруднит стыковку с другими программными модулями.

Очень важный этап анализа результатов. Следует иметь в виду, что решение задачи здесь только начинается. Нужно правильно интерпретировать результаты, представить их в обозримом виде для предоставления постановщику задачи.

Как видно из этого перечисления, источников ошибок очень много. Отсюда вытекает необходимость тестирования модели.

Тестирование компьютерной модели

Тестирование модели – это искусственный подбор такой ситуации, в которой поведение объекта можно заранее предсказать. Тестирование должно быть всесторонним. Чем больше тестов выполнено, тем увереннее можно говорить об адекватности модели.

Другой способ тестирования заключается в том, что исследуется реальный объект на основании натурального эксперимента, а результат этого эксперимента сравнивается с результатом моделирования. Это способ обычно считается более представительным. Следует лишь иметь в виду, что и здесь возможны ошибки, связанные с погрешностями измерительной аппаратуры.

Третий способ тестирования: результат численного эксперимента, полученного на основе разработанной модели, сравнивают с результатом другого численного эксперимента, полученными на основе существующих, хорошо проверенных универсальных пакетов и программ.

Преимущества математического моделирования

- Универсальность математических моделей. Поскольку математическая модель является параметрической, возможны исследования различных объектов простым измерением параметров.
- Относительная дешевизна математического моделирования по сравнению с другими видами.
- Простая переносимость компьютерных моделей из одной организации в другую.

Понятие о системах со сосредоточенными и с распределёнными параметрами

Рассмотрим пример:

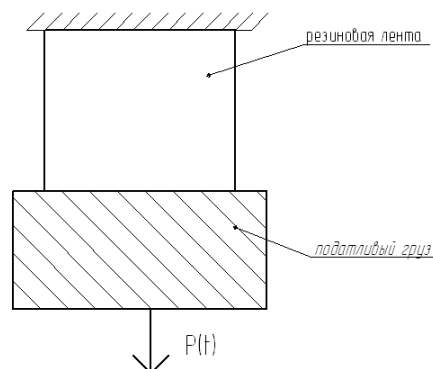


Рисунок 1.2. Схема нагружения

Возможны различные ситуации. Первая такова:

- Податливость груза гораздо меньше, чем податливость резиновой ленты (жесткость груза больше жёсткости ленты):

$$k_{гр} \geq k_{ленты} \quad (1.1)$$

- Пусть масса груза значительно больше массы ленты

$$m_{гр} \geq m_{ленты} \quad (1.2).$$

В этом случае массой ленты можно пренебречь, а груз считать абсолютно жёстким. Приходим к следующей расчётной схеме:



Рисунок 1.3. Расчетная схема при допущениях (1), (2)

Мы считаем, что вся податливость сосредоточена в пружине, а вся масса сосредоточена в грузе. При выполнении этих предположений такая система называется **системой со сосредоточенными параметрами**.

Поведение таких систем в математическом плане описывается обыкновенными дифференциальными уравнениями (ОДУ). В таких уравнениях присутствуют производные только по одной переменной (в рассматриваемом случае по времени). Методы решения уравнений: метод Эйлера, метод Рунге-Кутты и другие.

Если же податливость (масса) ленты соизмерима с податливостью (массой) груза, то предположения (1.1) и (или) (1.2) не выполняется, и система называется системой с распределёнными параметрами. Поведение таких систем описывается дифференциальными уравнениями в частных производных (ДУЧП).

Такие уравнения описывают поведение сплошной среды, где каждая точка влияет на любую другую точку. Методы их решения: методы конечных элементов, метод граничных элементов, метод конечных разностей.

Тема 2. Начало работы в среде MATHCAD

О некоторых пакетах инженерных расчетов и моделирования

В настоящее время существует немалое количество математических пакетов, и все они используются в экономическом и математическом моделировании, так как математика и экономика очень тесно связаны. Наиболее распространенные из них – это MathCAD, Matlab, Derive, Eureka, Mathematica, Maple. Данные пакеты многофункциональны.

EUREKA

Это интегрированная система автоматизации математических, физических, химических, электро - и радиохимических и прочих научно-технических расчетов «Eureka» позволяет:

- выполнить типовые математические и экономические расчеты;
- вычислять производные и интегралы;
- решать системы уравнений;
- искать экстремумы;
- выводить данные в табличной форме;
- строить по выбору график одной из функций;
- работать с комплексными числами.

«Eureka» также интегрирует в одной системе редактор для подготовки файлов, вычислитель, верификатор, проверяющий правильность вычислений, генератор отчетов, простой графопостроитель. Данный пакет работает в многооконной оболочке, позволяющей одновременно наблюдать описание решаемой задачи, результаты вычислений и их проверки, готовить отчет о работе и график выбранной функции.

DERIVE

Математический пакет «Derive» является системой символьной математики, т.е. позволяет производить символьные вычисления. Пакет обладает богатыми графическими возможностями. Задания и результаты вычислений представлены на экране в привычной математической записи. Интерфейс системы прост, но исключительно удобен для пользователя. Пакет можно эффективно использовать при решении широкого круга математических задач от планиметрии до теории вероятностей и статистики, а также производить финансовые расчеты.

«Derive» имеет несколько десятков встроенных функций:

- элементарные и специальные функции;
- действия с комплексными числами;
- решение задач математического анализа: отыскание пределов

функций, производных, определенных и неопределенных интегралов, конечных сумм и сумм числовых рядов, бесконечных произведений;

- операции векторной алгебры;
- действия с матрицами, вычисление обратной матрицы, собственных значений матрицы.

«Derive» имеет библиотеку функций-утилит, предназначенных для решения специальных задач, есть возможность пополнения библиотеки функциями пользователя.

MATLAB

«Matlab» является одной из старейших и проработавших систем автоматизации автоматических расчетов. Она была разработана С.В. Молером и с конца 70-х годов широко использовалась на больших ЭВМ. Система Matlab оказала большое внимание на разработку ряда пакетов для выполнения матричных операций, расчета систем управления, в свою очередь, вобрав в себя лучшие из средств, накопленных за более чем 30-летнюю историю развития матричных методов вычислений на ЭВМ.

«Matlab» - расширяемая система, и ее можно легко приспособить к решению нужных классов задач. Возможности ее весьма обширны, по скорости выполнения задач система не уступает многим другим подобным системам.

Своим названием (Matrix Laboratory – «матричная лаборатория») система «Matlab» обязана ориентации на матричные и векторные вычисления. Она выполняет операции над векторами и матрицами даже в режиме простых вычислений без какого-либо программирования.

Система содержит средства, особенно удобные для электро- и радиотехнических расчетов (операции с комплексными числами, полиномами, обработка данных, анализ сигналов и цифровая фильтрация). «Matlab» содержит также операторы построения графиков в декартовой и полярной системах координат, трехмерных поверхностей. На одном графике данная система может представить множество кривых, различающихся цветом и отличительными символами. Графики «Matlab» выводит в одном или несколько окон.

Будучи ориентированной, на работу с реальными данными, эта система выполняет все вычисления в арифметике с плавающей точкой. Система также поддерживает выполнение операций с массивами данных, регулирует сингулярное и спектральное разложения, вычисление ранга и чисел обусловленности матриц, поддерживает работу с алгебраическими полиномами, решение нелинейных уравнений и задач оптимизации, интегрирование в квадратурах, решение дифференциальных и разностных уравнений. В системе реализована удобная операционная среда, которая

позволяет формулировать проблемы и получать решения в привычной математической форме, не прибегая к рутинному программированию. Каждая из вышеописанных систем имеет свои достоинства и недостатки. Одни из них чрезвычайно сложны для освоения, требуют основательной математической подготовки и предназначены в первую очередь для профессионалов, имеют встроенные языки программирования математических действий и дополнительные библиотеки электронных справочников, другие, простые в освоении, обладают ограниченными возможностями и неудобным интерфейсом пользователя.

Общие сведения о MATHCAD

К универсальным программам, пригодным для решения поставленной задачи, относится, например, программа MathCad, которая представляет собой автоматизированную систему, позволяющую динамически обрабатывать данные в числовом и аналитическом (формульном) виде.

Фирма MathSoft быстро завоевала признание на рынке программного обеспечения, выпустив за 3 года целую серию пакетов MathCAD. В названии пакета присутствует известная аббревиатура CAD (Computer Aided Design), русский эквивалент САПР (система автоматизированного проектирование). Так обычно называются продукты, нацеленные на создание удобной программной среды для дизайнера (проектировщика). Аббревиатура CAD отражает стремление разработчиков создать программу, для которой не нужно быть специалистом по компьютерам. В целом можно сказать, что программный пакет MathCAD оправдывает такое название.

Каждая версия MathCAD выпускается в двух вариантах – Standard (обычная) и Professional (профессиональная). К названию профессиональной добавляется аббревиатура PRO.

Особое же место среди систем автоматизации математических расчетов занимает пакет «MathCAD». Пакет чрезвычайно прост в использовании. Его интерфейс настолько удобно сделан, что пользователь работает с рабочим листом программы, как с листом бумаги, где он пишет формулы и математические выражения в их привычной нотации. MathCAD является одной из самых мощных и эффективных математических систем, которая занимает особое место среди множества таких систем и по праву может называться самой современной, универсальной и массовой математической системой для всех пользователей. Отличительная черта этой системы – входной язык, максимально приближенный к математическому языку или языку научных статей и книг. Объединение в этой системе текстового редактора с возможностью использования

общепринятого языка позволяет пользователю получить готовый итоговый документ.

«MathCAD» столь же гибок, как самые мощные электронные таблицы и языки программирования, но легок в освоении и приятен в использовании. Система «MathCAD» содержит текстовый редактор, мощный графопостроитель и графический процессор.

Текстовый редактор служит для ввода и редактирования текстов. Текст может состоять из слов, математических выражений и формул, спецзнаков.

Вычислитель обладает уникальными возможностями:

- обеспечивает вычисления по сложным математическим формулам;
- имеет большой набор встроенных математических функций;
- позволяет вычислять ряды, суммы и произведения, определенные интегралы и производные;
- работать с комплексными числами;
- решать линейные и нелинейные уравнения;
- выполнять векторные и матричные операции.

В вычислитель входят и такие мощные средства как линейная и сплайн-интерполяция, регрессия, прямое и обратное быстрое преобразование Фурье. Легко можно менять разрядность чисел и погрешность итерационных методов.

«MathCAD» позволяет записывать на экране компьютера формулы в их привычном виде. Но формулы в «MathCAD» могут значительно больше, чем просто хорошо выглядеть. С их помощью можно решить почти любую мыслимую математическую задачу символьно либо численно. Можно реализовать текст в любых местах вокруг уравнений, чтобы документировать процесс решения.

Графический процессор служит для создания графиков. Графический процессор сочетает чрезвычайную простоту общения с пользователем с самыми изысканными возможностями графических средств. Простые графики нескольких функций пользователь может начать строить буквально в первые секунды знакомства с системой. По мере приобретения навыков работы с графическим процессором легко осваиваются и другие графические средства – графики в логарифмическом масштабе, масштабные сетки с любым числом делений, линии, отмеченные точками, прямоугольниками и ромбиками. Графика ориентирована на решение типичных математических задач. Возможно быстрое изменение размеров графиков, наложение их на текстовые надписи и перемещение в любое место документа. Можно создавать двумерные и трехмерные графики. Можно пользоваться иллюстрациями из других приложений Windows.

Также данный пакет обладает возможностью программирования. MathCad - это система, ориентированная на пользователя, который не обязан знать абсолютно ничего о программировании. Создатели MathCAD изначально поставили перед собой такую задачу, чтобы дать возможность профессионалам-математикам, физикам и инженерам самостоятельно проводить сложные расчеты, не обращаясь за помощью к программистам. Несмотря на блестящее воплощение этих замыслов, выяснилось, что вовсе без программирования MathCAD серьезно теряет в своей силе, в основном, из-за недовольства пользователей, знакомых с техникой создания программ и желающих осуществить свои расчеты в привычном для себя программистском стиле. Вместо знакомых принципов программирования, пользователям старых версий MathCAD предлагалось комбинировать несколько специфичных встроенных функций и ранжированные переменные. Последние версии MathCAD имеют не очень мощный, но весьма элегантный собственный язык. С одной стороны, он дает возможность программисту эффективно применять программный код в документах MathCAD. С другой, простота и интуитивность языка программирования позволяет быстро ему обучиться. Наконец, программные модули внутри документа MathCAD сочетают в себе и обособленность (поэтому их легко отличить от остальных формул), и простоту смыслового восприятия. Несмотря на небольшое число операторов, язык программирования MathCAD позволяет решать самые различные, в том числе и довольно сложные, задачи и является серьезным подспорьем для расчетов. В ранних версиях MathCAD встроенного языка программирования не было. Чтобы применять привычные операции проверки условий и организовывать циклы, приходилось изобретать причудливую смесь из встроенных функций условия и комбинаций ранжированных переменных. Несмотря на принципиальную эквивалентность определения функций и переменных через встроенные функции MathCAD или программные модули, программирование имеет ряд существенных преимуществ, которые в ряде случаев делают документ более простым и читаемым: возможность применения циклов и условных операторов, простота создания функций и переменных, требующих нескольких простых шагов, возможность создания функций, содержащих закрытый для остального документа код, включая преимущества использования локальных переменных и обработку исключительных ситуаций (ошибок). MathCAD имеет удобный многооконный интерфейс: одновременно может быть открыто несколько документов, каждый из которых занимает отдельное окно в пределах основного окна программы. Документ MathCAD

представляет собой макет некоторого математического документа. Пользователь обладает богатыми возможностями «проектирования»: в любое место документа можно добавить математическое выражение, график, объект для импорта и экспорта данных и т.д. Программа MathCAD обеспечивает в автоматическом режиме, дополняя введенные формулы полученными результатами. В итоге документ MathCAD выглядит как лист бумаги с выкладками математики. В терминах решения прикладных задач он содержит описание модели, исходные данные и полученный результат. Для импорта и экспорта данных в него вставляются компоненты, которые отвечают за интеграцию с другими приложениями.

Интерфейс MathCAD является дружественным к пользователю, в нем соблюдены современные стандарты построения приложений Windows. Пользователь Windows легко может освоить MathCAD. Упрощенный набор функций обеспечивает наглядность и простоту использования. Удобство использования MathCAD и наглядность представления математических формул достигнуты за счет ограничения функциональных возможностей пакета.

Справочная система MathCAD построена на основе HTML-Help и очень удобна. В пакет MathCAD включены многочисленные примеры использования всех возможностей, он популярен у нас в стране, его шестая версия даже была русифицирована, о нем появилось достаточное количество литературы на русском языке.

ПОСТРОЕНИЕ ВЫРАЖЕНИЙ

1. Запустить пакет.
2. Ознакомиться со всеми кнопками математической палитры.
3. Вычислить выражение: $66 \cdot 78$. Для этого набрать на клавиатуре: $66*78$ и нажать клавишу "=".
4. Вычислить выражение:

$$\frac{3}{4 + 5^2}$$

Чтобы создать дробь, надо:

а) напечатать символ "/" или

б) нажать кнопку "÷" на математической палитре.

5^2 можно создать следующими способами:

а) напечатать " 5^2 ";

б) напечатать "5" и нажать кнопку, помеченную x^y на математической палитре.

Напечатать выражение обоими способами.

Вычислить результат, нажав клавишу "=".

3. Вычислить выражение: $7^5/88$.

Чтобы в числителе оказалось все выражение "7⁵", необходимо заключить его в выделяющую рамку. Для изменения размеров выделяющей рамки используются клавиши "←", "↑", "↓", "→" и клавиша пробела [Space].

4. Используя клавиши математической палитры, вычислить выражения:

$$5!; 79^2; 56/8*32; 5,7^8; (61^{3*5})/90; 8^{-1}; -3*|-7,4|; \sqrt[3]{36,9} * (8,5-4).$$

5. Сохранить документ.

6. Вычислить следующие выражения:

$$\frac{5 - 3 \cdot 6^2}{-4 + \sqrt{7 + 6.5}} \quad \sqrt[3]{67 \cdot 90 - \frac{(4 + 2)}{51}} \quad \sqrt[5]{\frac{(45 - 87) \cdot (9 + 6.8)}{56^2 + 3 \cdot 4.8}}$$

Существует три способа вставки математических функций в рабочий лист Mathcad:

- 1) кнопка "Вставка функции" на панели инструментов;
- 2) кнопки вставки функций на математической палитре;
- 3) через пункт меню **Вставка → Функция**.

7. Вычислить тригонометрические выражения, используя кнопку "Вставка функции" на панели инструментов:

$$\cos(45); \sin(67); \tan(34).$$

8. Вычислить выражения, используя кнопки вставки функций на математической палитре:

$$\frac{\cos(45) - \log(3)}{\ln(52)} \quad (\sin(\pi) + \tan(\pi \cdot 4)) \cdot \cos(56) \quad \frac{\pi \cdot e^4}{\sqrt{\log(66)}}$$

9. Вычислить выражения, используя пункт меню **Вставка → Функция**:

$$\frac{\operatorname{asin}(45)}{\operatorname{atan}\left(\frac{\pi}{78}\right)} \quad \log(88) \cdot \cos(1) \quad \frac{\sin(56)}{\ln(3)}$$

При вычислении тригонометрических функций, Mathcad возвращает результат в радианах. Чтобы преобразовать результат из радианов в градусы, в месте ввода единиц необходимо напечатать *deg*. Например: $\cos(45 \cdot \text{deg})$.

10. Вычислить следующие тригонометрические выражения:
 $\cos(90 \cdot \text{deg})$; $\sin(270 \cdot \text{deg})$; $\cos(180 \cdot \text{deg}) - \sin(60 \cdot \text{deg})$.

11. Напечатать следующее выражение, используя кнопки вставки символов на математической палитре:

$$\xi^3 \frac{\lambda \cdot 68}{\sqrt{\alpha + 30}}$$

12. Напечатать выражение:

$$\frac{68^{54}}{89 \cdot 7.3} - \sqrt{12 + 86 \cdot f^4} \\ (58 + a^2) \cdot d$$

13. Отредактировать его, приведя к следующему виду:

Для этого установить курсор мыши внутри выражения и для перемещения использовать клавиши " \leftarrow ", " \uparrow ", " \downarrow ", " \rightarrow ".

$$\frac{68^{54}}{89 + 7.3} - \sqrt{12 \cdot 86 - f^4} \\ \sqrt{(58 + a^2) \cdot d}$$

Операции деления, возведения в степень и извлечения корня являются в Mathcad "цепкими" операциями. Это означает, что после создания одной из этих операций, все затем печатаемое будет частью знаменателя, показателя степени или подкоренного выражения, пока явно не переместить курсор, нажимая [Space] или клавишу " \uparrow ".

14. Напечатать выражение:

$$x^3 + \frac{1}{a + b} \cdot x^{c+d} - \sqrt{x+1} - 1$$

15. . Напечатать выражение:

$$\frac{x - 3 \cdot a^2}{-4 + \sqrt{y + 1} + 1}$$

16. Напечатать следующее выражение, используя кнопки математической палитры:

$$\sum_{i=0}^{\infty} \frac{d}{dx} (x^2 - 56) \cdot \sqrt{x}$$

РАБОТА С ОБЛАСТЯМИ

1. Запустить пакет.
2. Открыть свой файл (сохраненный на предыдущем занятии).

Выделение областей.

1. Нажмите и удерживайте левую кнопку мыши, чтобы закрепить один угол выделяющего прямоугольника.
2. Не отпуская кнопки мыши, переместите мышь, чтобы заключить все, что нужно выделить, внутри пунктирного выделяющего прямоугольника.
3. Как только выделяющий прямоугольник охватит все, что нужно выделить, освободите кнопку мыши. Mathcad заключит все выбранные области в пунктирные прямоугольники.

Копирование областей.

Как только области выделены, их можно копировать, либо:

- ◆ используя **Копировать** и **Вставить** (через буфер обмена), либо
- ◆ перетаскивая мышью.

1. Выделите области, как описано выше.
2. Скопируйте в буфер обмена (или нажав правую кнопку мыши, команда **Копировать**; или **Копировать** из меню **Правка**).
3. Щелкните мышью там, куда нужно поместить копию областей. Убедитесь, что выбрано свободное место - курсор должен иметь вид визира (крестика).
4. Щелкнуть правой кнопкой мыши, команда **Вставить**; или **Вставить** из меню **Правка**.
5. Перетащить с помощью мыши выделенные области на свободное поле рабочего листа.

Выравнивание областей.

Как только области выделены, их можно выравнивать с помощью пункта меню **Выровнять области** меню **Формат**. Чтобы задать направление выравнивания, выберите нужный пункт соответствующего подменю; также можно воспользоваться кнопками выравнивания областей на панели инструментов.

При выборе команды меню **Выровнять области** ⇒ **По левой границе** происходит следующее:

- ◆ Mathcad проводит невидимые вертикальные линии через левые края самой правой и самой левой выбранных областей, затем проводит вертикальную линию посередине между ними.
- ◆ Все выделенные области перемещаются таким образом, чтобы их левые края расположились по этой линии.

Команда **Выровнять области** ⇒ **По верхней границе** действует подобным же образом, только выравнивание производится по верхним краям областей.

1. Выделить области на рабочем листе.
2. Произвести их выравнивание по вертикали.
3. Произвести выравнивание по горизонтали.

Выравнивание областей может привести к их перекрыванию. Чтобы исправить эту погрешность, необходимо воспользоваться командой меню **Отделить области** из меню **Формат**.

Команда **Отделить области** может привести к перемещению большого числа областей рабочего документа. Если нужно разделить области аккуратнее, переместите области вручную.

Можно легко определить, перекрываются области или нет, с помощью команды **Области** из меню **Вид**. Mathcad отобразит пустое пространство серым цветом и оставит области белыми.

Проделать эту операцию.

Чтобы вернуть белую окраску незанятого пространства рабочего листа, выберите повторно **Области** из меню **Вид** и отключите флажок.

Удаление областей.

Чтобы удалить одну или несколько областей:

- Выделите области с помощью мыши.
- Выберите **Вырезать** из меню **Правка** или из контекстного меню после нажатия правой кнопки мыши.

Команда **Вырезать** удаляет выделенные области из рабочего документа и помещает их в буфер обмена. Если в сохранении этих областей нет нужды, можно выбрать также команду **Очистить** из меню **Правка** - она не изменяет содержимое буфера обмена.

Другие способы выделения областей.

Существует три различных способа выделения областей:

- 1) Охватывание мышью (см. выше).
- 2) Щелчок мышью на области при нажатой клавише **[Shift]**.
- 3) Выделение ряда последовательных областей щелчком мыши на первой и последней областях при нажатой клавише **[Ctrl]**.

Щелчок мышью при нажатой клавише **[Shift]** удобен, когда нужно выделить область или снять с нее выделение, не затрагивая другие области. Например:

- ◆ нужно добавить еще несколько областей к уже выделенным;
- ◆ нужно отменить выделение одной из нескольких выделенных областей.

Проделать все описанные операции.

ТЕКСТ

Чтобы создать текстовую область, необходимо проделать следующие операции:

- ◆ Щелкните на свободном месте рабочего листа, чтобы установить курсор в том месте, где предполагается начать текстовую область. Затем выберите команду **Вставка ⇒ Текстовая область**. Mathcad создает текстовую область, крестообразный курсор преобразуется в маркер ввода и появляется текстовая рамка.
- ◆ Другой способ: напечатать несколько знаков и нажать клавишу **[Space]**.
- ◆ Начиная печатать нужный текст.
- ◆ После завершения печати текста щелкните вне текстовой области. Текстовая рамка исчезнет.

Для редактирования текста необходимо щелкнуть между двумя символами в текстовой области. Редактирование производится как и в любом другом текстовом редакторе.

Для перемещения и удаления текстовых областей выполняются те же операции, что и при работе с математическими областями.

Изменение размеров текстовой области производится растягиванием за выделенные прямоугольники, расположенные на границе области.

1. Наберите текст:

Пример потока жидкости.

Задача: определить размер трубки при заданном давлении и потоке.

Измените начертание шрифта на курсив через пункт меню **Формат ⇒ Текст** (предварительно выделив текст).

2. Напечатать следующее выражение:

$$y^2 + \frac{\sqrt{x^3 - 78}}{z}$$

3. Создать текстовую область и напечатать текст:

Произведем вычисление необходимых параметров по формуле:

4. Скопировать набранную формулу и вставить в текстовую область в конце фразы.

Для печати греческих символов используется палитра греческих символов на математической панели инструментов.

1. Создать следующий фрагмент документа:

Для обозначения переменных используются как латинские, так и греческие символы: x , y , t , ρ , ω , η и другие. В качестве примера можно привести формулу:

$$d_j = \frac{1}{\sqrt{n}} \cdot \sum_{k=0}^{n-1} \psi_k \cdot e^{-2\pi i \cdot \frac{j}{n} \cdot k}$$

При наборе формулы для печати нижнего индекса используется клавиша "["; для печати знака "=" сочетание клавиш "Ctrl"+"=".

2. Напечатать текст:

Аппроксимирующая функция: $F(x, \alpha, \beta) = \alpha \cdot \beta \cdot x^{\beta-1} \cdot \exp(-\alpha \cdot x^\beta)$

Сумма квадратов отклонений должна быть минимизирована:

$$SSE(\alpha, \beta) = \sum_i (y_i - F(x_i \cdot \alpha \cdot \beta))^2$$

Оценка дисперсии:

$$\frac{1}{\sqrt{f}} = -2.0 \cdot \log\left(\frac{\varepsilon \cdot D^{-1}}{3.7} + \frac{2.51}{R \cdot \sqrt{1}}\right)$$

Уравнения и вычисления

Запустить пакет.

Определение переменной.

Определение переменной задает значение переменной всюду ниже места расположения определения. Чтобы определить переменную, нужно:

- ◆ Напечатать имя переменной, подлежащей определению.
- ◆ Напечатать символ двоеточие ":". В рабочем документе появляется знак ":=".
- ◆ Напечатать выражение, заканчивающее определение. Оно может содержать числовые константы и любые ранее определенные переменные и функции.

Задание. Определите следующим образом переменные:

$$t := 11.5 \quad s := 100 \quad v := \frac{s}{t} \quad m := 15 \quad KE := \frac{1}{2} \cdot m \cdot v^2$$

Сохранить файл под именем "Вычисления".

Определение функции.

В Mathcad могут быть также определены локальные функции. В отличие от переменной, значение функции зависит от значений аргументов. Определение функции аналогично определению переменной. Имя функции должно стоять слева, знак определения ":" должен стоять посередине, а выражение - справа. Основное различие состоит в том, что имя теперь включает *список аргументов*.

Задание. Вычислить расстояние между точкой (x,y) и началом координат и расстояния между точками.

1. Определить переменные и функцию. Для этого на рабочем листе напечатать следующий текст:

Вычисление расстояния между точками

$$\begin{array}{ll} x1:=0 & y1:=1.5 \\ x2:=3 & y2:=4 \\ x3:=-1 & y3:=1 \end{array} \quad \text{dist}(x,y):=\sqrt{x^2+y^2}$$

2. Вычислить расстояние между точкой и началом координат.

Напечатать текст:

Вычисление расстояния до начала координат

Печатать поочередно функции, нажимая после каждой знак вычислений "=". Mathcad будет возвращать вычисленный результат: расстояние между указанной точкой и началом координат:

$$\begin{array}{l} \text{dist}(x1,y1) \\ \text{dist}(x2,y2) \\ \text{dist}(x3,y3) \end{array}$$

3. Вычислить расстояние между точками. Печатать на экране текст, нажимая после каждой функции знак "=":

$$\text{dist}(x_2-x_1, y_2-y_1)$$

$$\text{dist}(x_3-x_1, y_3-y_1)$$

$$\text{dist}(x_3-x_2, y_3-y_2)$$

В результате рабочий лист должен иметь вид:

The screenshot shows a spreadsheet window with a toolbar at the top (Normal, Arial, 10, Bold, Italic, Underline). The main content area contains the following text and formulas:

Вычисление расстояния между точками

$x_1 := 0$ $y_1 := 1.5$

$x_2 := 3$ $y_2 := 4$

$x_3 := -1$ $y_3 := 1$ $\text{dist}(x, y) := \sqrt{x^2 + y^2}$

Вычисление расстояния до начала координат

$\text{dist}(x_1, y_1) = 1.5$

$\text{dist}(x_2, y_2) = 5$

$\text{dist}(x_3, y_3) = 1.414$

+ Вычисление расстояния между точками

$\text{dist}(x_2 - x_1, y_2 - y_1) = 3.905$

$\text{dist}(x_3 - x_1, y_3 - y_1) = 1.118$

$\text{dist}(x_3 - x_2, y_3 - y_2) = 5$

4. Напечатать:

$$x := 2$$

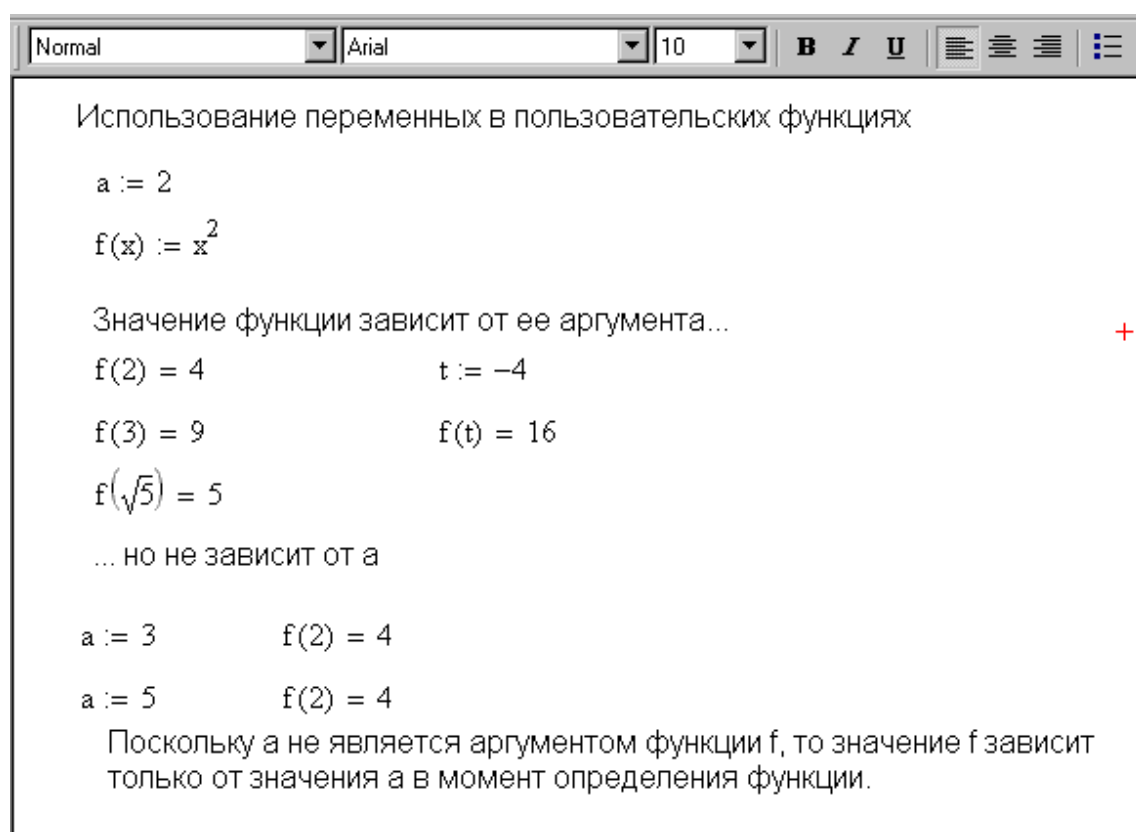
$$f(x) := \frac{3 \cdot x}{2} \cdot \ln(x)$$

5. Вычислить значение функции $f(x)$ при значении $x=2$: напечатать ниже набранного условия $f(x)$ и нажать клавишу "=".

6. Вычислить значение этой же функции $f(x)$ при значении $x=7$. Для этого в определении аргумента $x:=2$ изменить "2" на "7" и щелкнуть мышью на пустом поле листа. Mathcad автоматически произведет пересчет.

Если нужно, чтобы функция зависела от значения переменной, то эта переменная должна быть использована как аргумент. В противном случае Mathcad использует для вычисления функции фиксированное значение этой переменной в том месте рабочего документа, где находится определение этой функции.

Задание. Выполнить следующий пример (рабочий лист должен иметь вид, показанный в примере):



Использование переменных в пользовательских функциях

$a := 2$

$f(x) := x^2$

Значение функции зависит от ее аргумента...

$f(2) = 4$ $t := -4$

$f(3) = 9$ $f(t) = 16$

$f(\sqrt{5}) = 5$

... но не зависит от a

$a := 3$ $f(2) = 4$

$a := 5$ $f(2) = 4$

Поскольку a не является аргументом функции f , то значение f зависит только от значения a в момент определения функции.

Переменные могут быть определены в одном и том же документе неоднократно, например, дважды. Mathcad будет использовать первое определение для всех выражений ниже первого определения и выше второго. В выражениях, находящихся ниже второго определения, Mathcad использует второе определение.

Задание. Выполнить следующий пример:

Normal Arial 10 B I U

Сначала определим...

$V := 1000$ $n := 3$ $R := 0.0820562$ $T := 373$

$P := \frac{n \cdot R \cdot T}{V}$ $P = 0.092$

Изменим определение для V и T...

$V := 500$ $T := 323$

Эти новые определения прилагаются ко всем вычислениям ниже в рабочем документе...

$P := \frac{n \cdot R \cdot T}{V}$ $P = 0.159$ +

Вычисление выражений.

Чтобы вычислить выражение, нужно:

- ◆ Напечатать выражение, содержащее любую допустимую комбинацию чисел, переменных и функций. Любые переменные или функции, используемые в этом выражении, должны быть определены в рабочем документе заранее.
- ◆ Нажать клавишу "=". Mathcad вычисляет значение выражения и показывает его после знака =.

Задание.

1. Выполнить следующий пример:

Normal Arial 10 B I

Движение с постоянной скоростью

$t := 11.5$

$s := 100$

$v := \frac{s}{t} \quad v = 8.696$

$m := 15 \quad m \cdot v = 130.435$

$\frac{1}{2} \cdot m \cdot v^2 = 567.108$

$KE := \sqrt{m \cdot \frac{1}{v} + 54} \quad KE = 7.465$

$$x := 4 \quad y := 7 \quad z := -6.8$$

$$ID := x^4 + \sqrt{\frac{y}{67} - z^2}$$

3. Переопределить переменные, присвоив им следующие значения:
 $x := -0.6$; $y := 12$; $z := 4$.

4. Произвести следующие вычисления:

$$x := 0.98 \quad y := 4 \quad z := 90 \quad t := 1$$

$$M := \frac{z^2}{x - 56} \cdot \sqrt{t - (x^4 + z \cdot y)}$$

$$s := -4 \quad m := 89 \quad n := 0.9$$

$$F := \frac{s}{m^2 - 2 \cdot n} + 2 \cdot x^4 - 4 \cdot \sqrt{\frac{n}{m^{-3}}}$$

5. Изменить значения переменных x, y, z, t, s, m, n и вновь произвести вычисления.

1. Найти значения выражения:

$$F(X) := \sum_{k=0}^3 \frac{3!}{k! \cdot (3-k)!} \cdot x^k \cdot (2)^{3-k}$$

для случаев $F(2); F(-5)$.

Для этого после определения функции $F(x)$ (после того, как напечатали выражение в рамке) печатать поочередно $F(2)$ и $F(-5)$, нажимая после каждой функции знак « \Rightarrow ».

Другой способ:

2. Напечатать:

$$x := 2$$

$$\sum_{k=0}^3 \frac{3!}{k! \cdot (3-k)!} \cdot x^k \cdot (2)^{3-k}$$

3. Выделить переменную x (так, чтобы она оказалась в синем уголке). **Символы \rightarrow Переменные \rightarrow Вычислить.**

4. Подобным же образом вычислить: $x^3 - 10x + 6$.

Mathcad возвращает результат в виде вектора с тремя длинными значениями. Выделить весь вектор, нажать « \Rightarrow ». Результатом будет вектор с числами.

5. Вычислить следующие выражения (и подобным же образом упростить результаты):

$$\frac{x^4}{2 \cdot x} + 67 \cdot x^2 - 8$$

$$\sqrt{x} - 10 \cdot x^2$$

$$\frac{\sqrt{a + 67 \cdot c}}{a^2} - 0.8 \cdot c^2 \qquad b^3 - a \cdot b^2 + \frac{86}{c \cdot b} - c \cdot a^2$$

Нахождение корней уравнения и решения неравенств.

Чтобы найти корни уравнения или решить неравенство, надо:

- ◆ напечатать выражение;
- ◆ выделить в любом месте переменную, относительно которой уравнение решается;
- ◆ **Символы → Переменные → Вычислить.**

1. Найти корни уравнений:

$x^3 - 5 \cdot x^2 - 4x + 20 > 0$	$\frac{1}{2}x^2 + x + 2$
$\frac{x^2}{56} + \frac{1}{x} - 6 \cdot x < 9.6$	$\sin(x) = \frac{1}{3} \cdot \tan(x)$
$\frac{\cos(x)}{2} = 25 \cdot x$	$e^x = -1$

Символьное решение системы уравнений.

- ◆ Напечатать слово *Given*.
- ◆ Напечатать систему уравнений.
- ◆ Напечатать функцию *Find*, соответствующую системе уравнений.
Аргументами функции являются переменные, относительно которых система решается.
- ◆ Нажать [Ctrl] = . (клавиша **Ctrl**, сопровождаемая точкой). Mathcad отображает символьный знак равенства →.

1. Выполнить пример:

Given	
$x + 2 \cdot \pi \cdot y = a$	Find(x, y) →
$4 \cdot x + y = b$	
$\left[\begin{array}{c} \frac{-(-2 \cdot \pi \cdot b + a)}{(-1 + 8 \cdot \pi)} \\ \frac{(4 \cdot a - b)}{(-1 + 8 \cdot \pi)} \end{array} \right]$	

1. Напечатать систему уравнений:

$$z \cdot x - b \cdot y + x = b \cdot a$$

$$x + y + z = 56 \cdot (a + 4 \cdot b)$$

Решить относительно переменных (x,y) ; (x,z) ; (y,z) .

Символьное транспонирование матриц . Нахождение в символьном виде детерминанта.

1. Напечатать матрицы:

$$\begin{pmatrix} x & 1 & a \\ -b & x^2 & -a \\ 1 & b & x^3 \end{pmatrix} \quad \begin{pmatrix} \lambda & 2 & \lambda^2 \\ 0 & a \cdot \lambda & a \cdot b \\ 0 & 1 & 4 \end{pmatrix}$$

2. Для каждой из них выполнить команды:

Символы → Матрицы → Транспонирование

Символы → Матрицы → Определитель.

ГРАФИКИ

В Mathcad допустимо в одной графической области, используя одни и те же координатные оси, создавать несколько графиков, отображающих различные зависимости.

Создать график можно несколькими способами:

1. **Формат → График → X-Y зависимость;**

2. на математической палитре выбрать **Графики → Декартов график;**

3. нажать на клавиатуре **@**.

Mathcad создает пустой график с полями ввода.

Вставить графики всеми способами.

Пустое поле в середине горизонтальной оси предназначено для независимой переменной графика.

Пустое поле в середине вертикальной оси содержит выражение, график которого нужно построить.

1. На одном из пустых графиков ввести в поле по оси абсцисс переменную x , по оси ординат – выражение функции $5 \cdot x^2 - 6/x + 3$.

2. Щелкнуть мышью на рабочем листе вне области графика или нажать **F9**.

3. Над другим пустым графиком определить функцию:

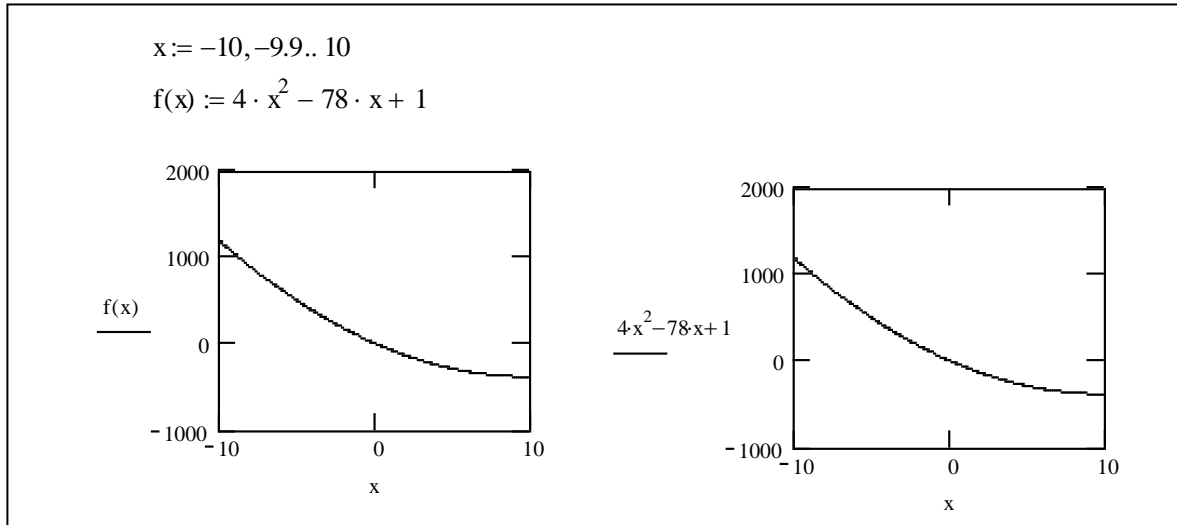
Ввести по оси абсцисс z , по оси ординат - $f(z)$.

4. **F9** или щелкнуть мышью вне области графика.

$$f(z) := \sqrt{z} + z^2 - 8$$

Линия под выражением функции указывает тип траектории и цвет, используемый для отображения данной кривой.

5. Выполнить пример:

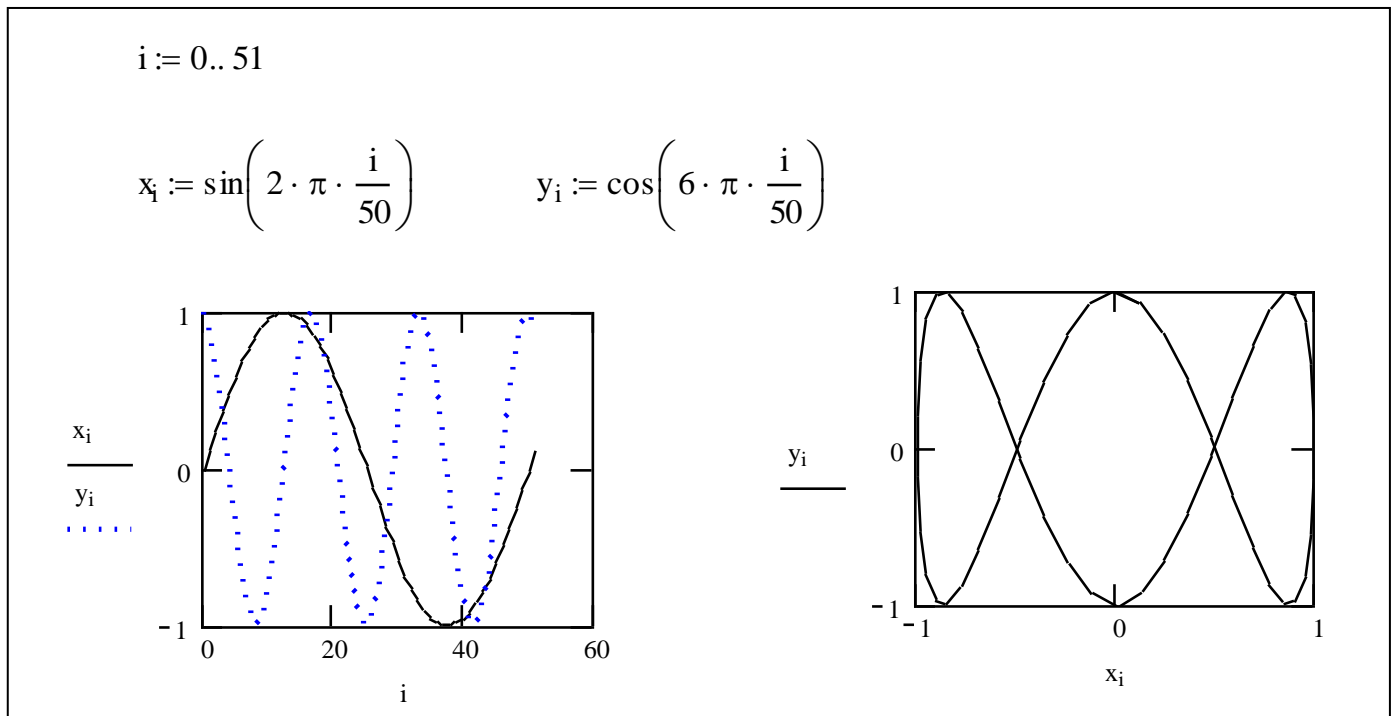


При определении переменной x указан желаемый диапазон изменения значений аргумента: -10 – нижняя граница;

-9.9 – указывает шаг изменения значения аргумента (можно не указывать; в этом случае шаг принимается по умолчанию);

10 – верхняя граница.

6. Выполнить пример:



Форматирование осей.

Чтобы изменить формат графика, необходимо:

1. Выделить график, щелкнув на нем мышью.
 2. **Формат** → **График** → **X-Y зависимость**.
- Появляется диалоговое окно форматирования осей.
Другой способ:

1. Выделить график.
2. Установить курсор мыши на графике и нажать правую клавишу мыши.

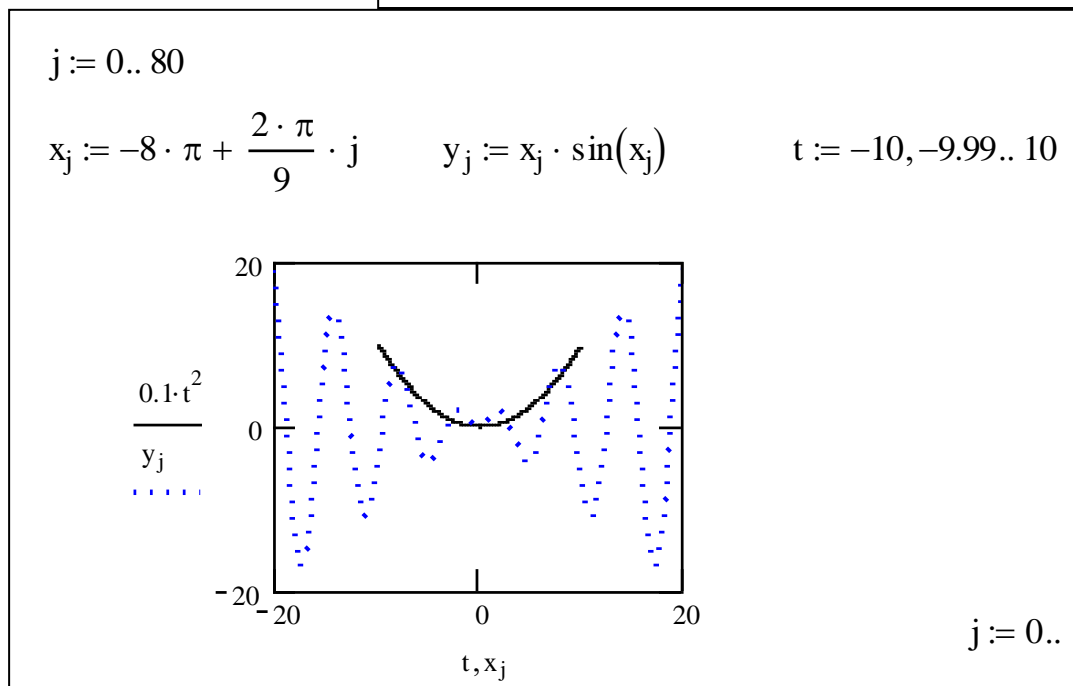
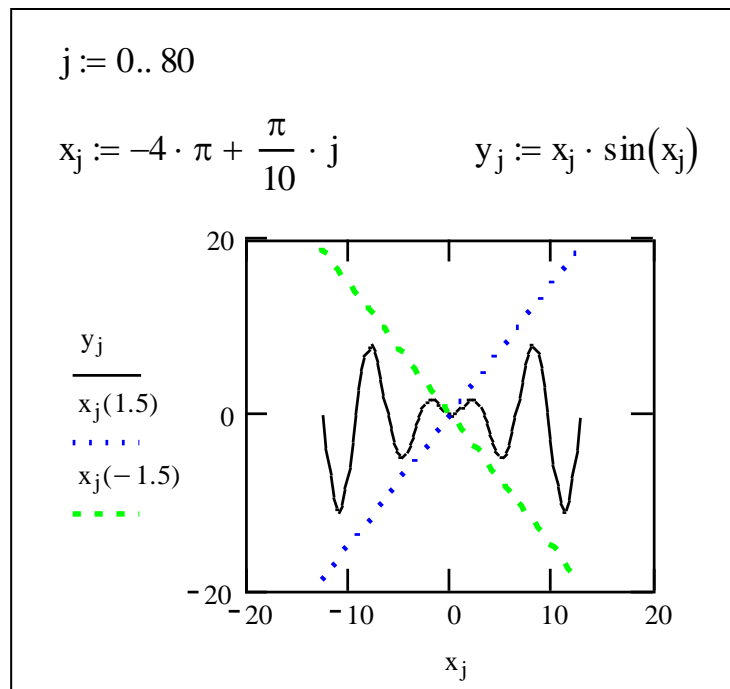
Третий способ: дважды щелкнуть мышью в области графика.

Испытать все способы.

В диалоговом окне находится полный набор установок для каждой оси.

1. Изменить соответствующие установки на разных графиках, нажимая в конце «**ОК**», чтобы принять изменения и закрыть диалоговое окно.
2. На одном из графиков изменить установки и нажать клавишу «**Применить**», чтобы увидеть перерисованный график, не закрывая диалоговое окно.

Выполните примеры:



Установка границ на осях координат.

Существуют следующие возможности устанавливать границы на осях координат:

- ◆ Автоматически, с включенным режимом Авто масштаб.
- ◆ Автоматически, с выключенным режимом Авто масштаб.
- ◆ Вручную, вводя границы непосредственно на графике.

Проверить на одном из графиков включенный и выключенный режим Авто масштаб.

На одном из графиков изменить вручную границы на осях:

1. Щелкнуть мышью на графике. Mathcad отображает четыре дополнительных числа, под одному на каждую границу на осях.
2. Изменить эти числа. Щелкнуть мышью вне графика.

Форматирование отдельных кривых.

1. Выделить график с тремя кривыми.
2. Открыть диалоговое окно форматирования графика, выбрать закладку **След**.
3. Для каждой линии поочередно изменить символ, линию, цвет, тип и вес, нажимая клавишу **Применить**, чтобы посмотреть результат.
4. Отключить флажок **Скрыть легенду**. **Применить**.

Оформление графика.

1. На панели инструментов установить шрифт Times New Roman (или другой кириллический).
2. Выделить график.
3. Вызвать диалоговое окно форматирования.
4. Открыть закладку **Метки**.
5. В поле «**Название**» ввести «**График**». Включить флажок «**Показать название**».
6. В поле «**Ось X**» ввести «**Название оси абсцисс**», в поле «**Ось Y**» – «**Название оси ординат**». **ОК**.

Изменение масштаба на графике.

Для увеличения элемента графика.

1. Выделить график.
2. Нажать правую клавишу мыши. В появившемся контекстном меню выбрать **Масштаб**.
3. На чертеже графика нажать мышью в одном углу области, которую нужно увеличивать.
4. Нажав и удерживая левую кнопку мыши, переместить мышью на нужное расстояние. Когда вся область, которую нужно увеличить, попадет в пунктирный прямоугольник, отпустите левую кнопку мыши. Координаты выбранной области отображаются в полях Min и Max диалогового окна.

5. Нажать кнопку **Масштаб+**, чтобы перерисовать график, не закрывая диалоговое окно.
6. Вернуться к первоначальному виду - кнопка **Масштаб-**.
7. Вновь выделить область графика для увеличения. Нажать **ОК**.
8. Вернуться к первоначальному виду графика: выделить график; нажать правую клавишу мыши; в контекстном меню войти в команду **Масштаб**; нажать кнопку **Обзор**.

Считывание координат точек графика.

1. Выделить график.
2. Нажать правую клавишу мыши. В появившемся контекстном меню выбрать **Трассировка**.
3. Внутри чертежа нажать кнопку мыши и переместить по той кривой, координаты точек которой хотите получить. Значения абсциссы и ординаты показаны в рамках «**X-значение**» и «**Y-значение**».
4. Поочередно скопировать координаты X и Y в буфер обмена и вставить на рабочий лист:
 - а) нажать клавишу «**Копировать X**» и вставить скопированное значение на рабочий лист;
 - б) нажать клавишу «**Копировать Y**» и вставить скопированное значение на рабочий лист.

Выполнить задание:

1. На одном чертеже построить графики для любых трех функций.
2. Для всех кривых установить цвет “черный”.
3. Для одной кривой установить тип “ступенчатый” (“step”).
4. Для другой – “гистограмма” (“bar”).
5. Для третьей установить параметры: вес – 3; символ – o’s.

ПРОГРАММИРОВАНИЕ

Mathcad имеет встроенный язык программирования, позволяющий решать широкий круг задач. Тем не менее, нужно отметить, что возможность программирования не является главным достоинством системы. Главное достоинство – возможность решения множества типичных задач без программирования.

Программа Mathcad есть частный случай выражения Mathcad.

Пример. Простая программа для вычисления функции

$$f(x, w) = \frac{x}{w}$$

1. Введите левую часть определения функции $f(x, w)$ и знак $:=$.
2. Откройте панель программирования на математической палитре. Нажмите на панели кнопку «**Add Line**» или на клавиатуре клавишу **]**. Появится вертикальный

столбец с двумя полями ввода для занесения операторов, образующих программу. Поля ввода для дополнительных операторов открываются с помощью щелчка по кнопке «**Add Line**» или нажатия клавиши **]**.

3. Перейдите в верхнее поле ввода. Напечатайте z и нажмите кнопку «**←**» на панели программирования.

4. В поле ввода справа от «**←**» введите z/w (присвоили z значение z/w).

5. Последнее поле ввода предназначено для задания возвращаемого функцией значения $\log(z)$. Программа имеет вид:

$$f(x, w) := \left| \begin{array}{l} z \leftarrow \frac{x}{w} \\ \log(z) \end{array} \right.$$

Теперь эту функцию можно использовать точно так же, как любую другую функцию.

6. Выше написанной программы определите значения x и w :

$$x := 2 \quad w := 9$$

7. После программы напечатайте $f(x, w)$ и нажмите клавишу **=**.

Программа может состоять из любого числа операторов. Чтобы прибавить оператор, нужно щелкнуть по кнопке «**Add Line**» или нажать клавишу **]**. Mathcad добавляет поле ввода внизу выделенного к этому моменту оператора. Чтобы исключить позицию ввода, ее нужно выделить, заключив в выделяющую рамку, и нажать клавишу **[Del]**.

Программа Mathcad есть выражение, состоящее из последовательности операторов, каждый из которых является, в свою очередь, выражением. Программа возвращает значение. Этим значением является значение последнего выражения, выполненного программой.

Выполнить примеры (набирая и текст):

$a := -4$	$b := 0.9$	$c := 6$	
$q(a, b, c) := \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$			Хотя можно определить сложную функцию целиком на одной строке...
$q(a, b, c) := \left \begin{array}{l} \text{discr} \leftarrow b^2 - 4 \cdot a \cdot c \\ \text{num} \leftarrow -b + \sqrt{\text{discr}} \\ \text{denom} \leftarrow 2a \\ \frac{\text{num}}{\text{denom}} \end{array} \right.$...часто удобнее разбить определение на несколько простых шагов
			$q(a, b, c) = -1.117$

$x := 7$	$y := 0.5$
функция	$f(x, y) := \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}} + \sqrt{x^2 + y^2}$
может быть записана в виде	
$f(x, y) := \left \begin{array}{l} r \leftarrow \sqrt{x^2 + y^2} \\ \frac{\sin(r)}{r} + r \end{array} \right.$	
$f(x, y) = 7.113$	

Условные операторы.

$f(x, z) := \frac{\sqrt{2 \cdot (x + z^2)} - (x + z^2)}{x + z}$

Написать программу для вычисления функции:

Перед этим задать значения переменных.

Проверить правильность написания программы, вычислив значение функции после задания в виде уравнения и после выполнения программы. Рабочий документ будет иметь вид:

$x := 4$	$z := 8$
$f(x, z) := \frac{\sqrt{2 \cdot (x + z^2)} - (x + z^2)}{x + z}$	$f(x, z) = -4.695$
$f(x, z) := \left \begin{array}{l} \blacksquare \\ \blacksquare \end{array} \right.$	(текст программы)
$f(x, z) = -4.695$	

Условные операторы.

Обычно Mathcad выполняет операторы программы в порядке сверху вниз. Но могут встретиться случаи, когда какой-нибудь оператор нужно выполнить только

в случае выполнения некоторого условия. Этого можно добиться с помощью оператора *if*.

Он задается в виде:

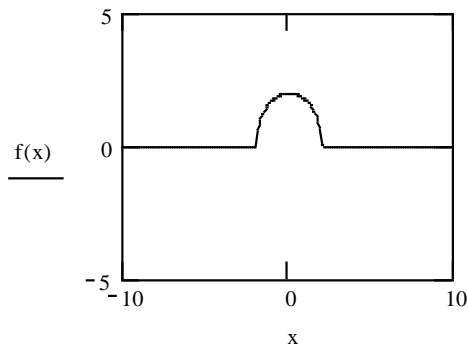
Выражение *if* Условие

Оператор *otherwise* – оператор иного выбора. Обычно используется совместно с оператором *if*.

Выполните примеры:

$$f(x) := \begin{cases} 0 & \text{if } |x| > 2 \\ \sqrt{4 - x^2} & \text{otherwise} \end{cases}$$

$$x := -10, -9.9..10$$



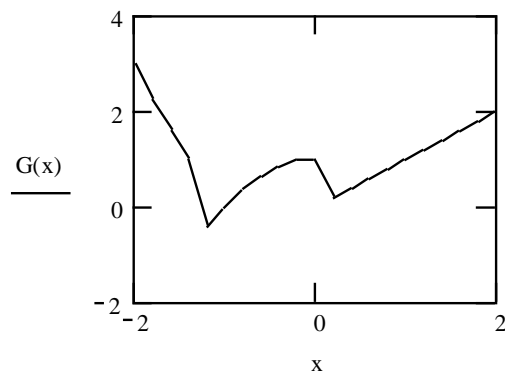
границы по осям на графике
установлены вручную

эквивалентное определение той же функции:

$$g(x) := \text{if}(|x| > 2, 0, \sqrt{4 - x^2})$$

$$x := -2, -1.8..2 \quad F(x) := x^2 - 1$$

$$G(x) := \begin{cases} F(x) & \text{if } -2 \leq x < -1.3 \\ (-F(x)) & \text{if } -1.3 \leq x < 0 \\ x & \text{otherwise} \end{cases}$$



Циклы.

Mathcad предлагает два вида циклов, отличающихся по способу определения условий завершения цикла:

[Оглавление](#)

1) если заранее точно известно необходимое число выполнений цикла, то целесообразно использовать цикл типа **for**;

2) если цикл должен завершиться по выполнении некоторого условия, причем момент выполнения этого условия заранее не известен, то целесообразно использовать цикл типа **while**.

Оператор **while** служит для организации циклов, действующих до тех пор, пока выполняется некоторое Условие. Этот оператор записывается в виде:
while Условие.

Выполнить пример: использование цикла типа **while** для отыскания первого вхождения заданного числа в матрицу:

<code>m := 0..2500</code>	- создание вектора	
<code>v_m := 1 + sin(m)</code>		
<code>t(v, thres) :=</code>	<code>j ← 0</code>	- инициализация счетчика
	<code>while v_j ≤ thres</code>	
	<code> j ← j + 1</code>	
	<code>j</code>	- возврат значения
<code>t(v, 1.98) = 8</code>	- величина 1.98 впервые превосходится восьмым элементом вектора	
	$v_m =$	
		1
		1.841
		1.909
		1.141
		0.243
		0.041
		0.721

Оператор **break**. Часто удобно выйти из цикла или остановить выполнение программы при выполнении некоторого условия. Например, для программы на рис. 1 существует возможность заикливания. Если каждый элемент из v меньше, чем $thres$, то условие никогда не станет ложным и поиск выйдет за пределы вектора, что приведет к сообщению об ошибке «индекс вне границ». Чтобы это не случилось, можно использовать оператор `_break`, как показано в следующем примере.

Выполнить пример:

$m := 0..2500$ - создание вектора
 $v_m := 1 + \sin(m)$
 $t(v, \text{thres}) :=$

$j \leftarrow 0$ (break) if $\max(v) \leq \text{thres}$ while $v_j \leq \text{thres}$ $j \leftarrow j + 1$ $\left(\begin{array}{c} j \\ v_j \end{array} \right)$	- инициализация счетчика - возврат значения
---	--

$t(v, 1.56) = \left(\begin{array}{c} 1 \\ 1.841 \end{array} \right)$ - величина 1.56 превосходитя первым элементом вектора. Его начение равно 1.841.

Оператор ***for*** служит для организации циклов с заданным числом повторений. Он записывается в виде:

For Var e Nmin..Nmax

Выполнить примеры:

$\text{sum}(n) :=$

$s \leftarrow 0$ for $i \in 1..n$ $s \leftarrow s + i$	$n := 44$
--	-----------

$\text{sum}(44) = 990$

Программа эквивалентна оператору

$\text{join}(r, s) :=$	$\sum_{i=1}^n i = 990$
$m \leftarrow 0$ for $x \in r, s$ $v_m \leftarrow x$ $m \leftarrow m + 1$ v	

$r := \left(\begin{array}{c} 100 \\ 101 \\ 102 \end{array} \right)$ $s := \left(\begin{array}{c} 1 \\ 2 \end{array} \right)$ $\text{join}(r, s) = \left(\begin{array}{c} 1 \\ 2 \\ 100 \\ 101 \\ 102 \end{array} \right)$

Оператор *continue* используется для продолжения работы после прерывания программы. Обычно он применяется совместно с операторами задания циклов *while* и *for*, обеспечивая после прерывания возврат в начало цикла.

Оператор *return* прерывает выполнение программы и возвращает значение своего операнда, стоящего следом за ним. Например, в приведенном ниже случае **Return 0 if x<0**

Будет возвращаться значение 0 при любом $x < 0$.

Оператор *on error* является оператором возвращения ошибок, позволяющим создавать конструкции обработчиков ошибок. Этот оператор задается в виде:

Выражение_1 on error Выражение_2

Если при выполнении **Выражения_1** возникает ошибка, то выполняется **Выражение_2**. С оператором *on error* связана функция *error*, которая обычно используется для возврата текстового сообщения об ошибке.

Выполнить пример:

$$F(x) := \begin{cases} \text{error}(\text{"x должно быть положительным"}) & \text{if } x \leq 0 \\ \frac{1}{x} & \text{otherwise} \end{cases}$$

$$F(3) = 0.333$$

$$F(0) = \blacksquare$$

$$F(-3) = \blacksquare$$

Щелкните на неопределенном выражении, чтобы увидеть сообщение об ошибке.

Подпрограммы.

Программа является выражением, состоящим из операторов, каждый из которых содержит выражения. Таким образом, оператор в программе может быть другой программой.

Примеры программ – Ресурс Центр.

Выполнить пример:

Вычисление произведения последовательности целых чисел от 1 до n

$$\text{proizv}(n) := \begin{cases} p \leftarrow 1 \\ \text{for } i \in 1..n \\ p \leftarrow p \cdot i \\ p \end{cases}$$

$$\text{proizv}(5) = 120$$

$$\text{proizv}(45) = 1.196 \times 10^{56}$$

В отличие от двумерных графиков, которые используют дискретные аргументы и функции, трехмерные графики требуют матрицы значений.

Чтобы создать поверхность:

- ◆ определите матрицу значений, которую необходимо отобразить графически. Mathcad будет использовать номер строки и столбца матрицы в качестве координат по осям x и y. Элементы матрицы будут представлены на графике как высоты выше или ниже плоскости x-y;
- ◆ выберите **График поверхности** из меню **Графика**;
- ◆ напечатайте имя матрицы в поле ввода.

1. Создайте графики для следующих матриц значений:

$$M := \begin{pmatrix} 1 & 3 & 4 \\ 2.5 & 10 & 4 \\ -4 & 1 & -6 \end{pmatrix} \quad n := \begin{pmatrix} 4 & 90 \\ 60 & 5 \end{pmatrix} \quad l := \begin{pmatrix} 45 & 5 & 2 & 101 \\ 78 & 67 & -5 & 3 \\ 3 & 89 & 0.88 & 45 \\ -0.9 & 4 & 1 & -97 \end{pmatrix}$$

2. Щелкнуть на графике для матрицы l правой кнопкой мыши. Выбрать в контекстном меню пункт **Формат**.
3. В появившемся диалоговом окне **3-D Plot Format** выбрать закладку **Общее**. В разделе **Показать как:** переключать поочередно переключатели **Поверхность**, **Контур**, **Точки данных**, **Область векторов**, **Диаграмма**, **Путь**, нажимая после каждого клавишу **Применить** (чтобы посмотреть вид получившегося графика, не закрывая диалоговое окно).
4. Выбрать закладку **Внешний Вид**. В разделе **Свойства заливки** включить флажок **Сетка**, выбрать **Цвет → Палитра**. Посмотреть поочередно все свойства заливки (нажимая клавишу **Применить**).
5. Отключить флажок **Сетка**. Посмотреть те же свойства.
6. Аналогично рассмотреть разделы **Свойства линии**, **Свойства точек**.
7. Выбрать закладку **Название**. Ввести название «Трехмерный график», установить переключатель **Сверху**. **ОК**.
8. Выбрать закладку **Специальный**. Рассмотреть **Свойства контура**.
9. Просмотреть остальные закладки.
10. Щелкнуть на графике правой кнопкой мыши и поочередно включать все пункты контекстного меню.

График функции двух переменных.

Типичный поверхностный график показывает значения функции двух переменных. Чтобы создать такой график, необходимо сначала образовать матрицу, содержащую значения этой функции, а затем построить поверхностный график этой матрицы. Стандартные этапы создания такого графика функций двух переменных показаны в нижеследующем примере:

11. Определите функцию двух переменных.
12. Решите, сколько необходимо точек по осям x и y . Определите дискретные аргументы i и j , чтобы индексировать эти точки. Например, если необходимо иметь по 10 точек в каждом направлении введите:
 $i:=0..9$ $j:=0..9$.
13. Определите x_i и y_j как равномерно располагаемые точки на осях x и y .
14. Заполните матрицу M значениями $f(x_i, y_j)$.
15. Выберите **График поверхности** из меню **Графика**.
16. Напечатайте **M** в поле ввода и щелкните вне графической области.

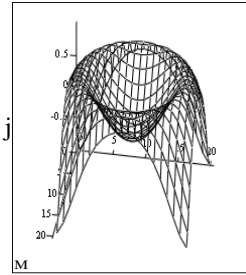
График поверхности

$$N:=20$$

$$i:=0..N \quad j:=0..N \quad x_i := -1.5 + 0.15 i \quad y_j := -1.5 + 0.15 j$$

$$f(x,y) := \sin(x^2 + y^2)$$

$$M_{i,j} := f(x_i, y_j)$$



17. Выполните для графика пункты 2 – 10.

Создание параметрических поверхностей.

Необходимо:

- ◆ напечатать имена трех матриц, имеющих одинаковое число строк и столбцов, в поле ввода внизу графической области;
- ◆ Mathcad интерпретирует эти матрицы как x -, y - и z -координаты точек поверхности.

Выполните пример:

$$m := 0..20$$

$$n := 0..20$$

$$r := 2$$

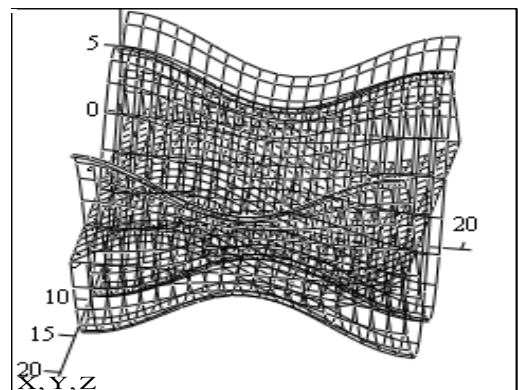
$$R := 6$$

$$\phi_m := \frac{2 \cdot \pi \cdot m}{20} \quad \theta_n := \frac{2 \cdot \pi \cdot n}{20}$$

$$X_{m,n} := (R + r \cdot \cos(\theta_n)) \cdot \cos(\phi_m)$$

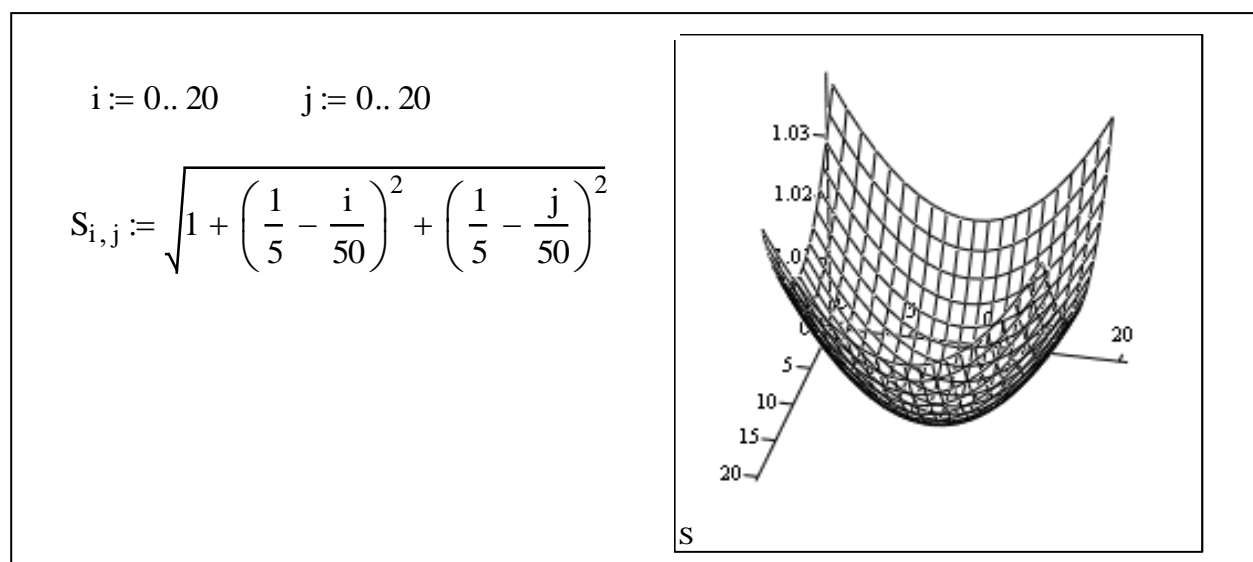
$$Z_{m,n} := r \cdot \sin(\theta_n)$$

$$Y_{m,n} := (R + r \cdot \cos(\theta_n)) \cdot \sin(\phi_m)$$



Нельзя преобразовать параметрическую поверхность в другой тип трехмерного графика.

Выполнить пример:



1. Щелкнуть на графике правой кнопкой мыши. Выбрать в контекстном меню пункт **Формат**.
2. В появившемся диалоговом окне **3-D Plot Format** выбрать закладку **Ось**. Для закладки **X-ось** в разделе **Формат оси** отключить флажок **Нумерация**. Клавиша **Применить**.
3. Прделать то же самое для закладок **Y-ось** и **Z-ось**. В конце – **ОК**.

ТРЕХМЕРНЫЕ ГИСТОГРАММЫ

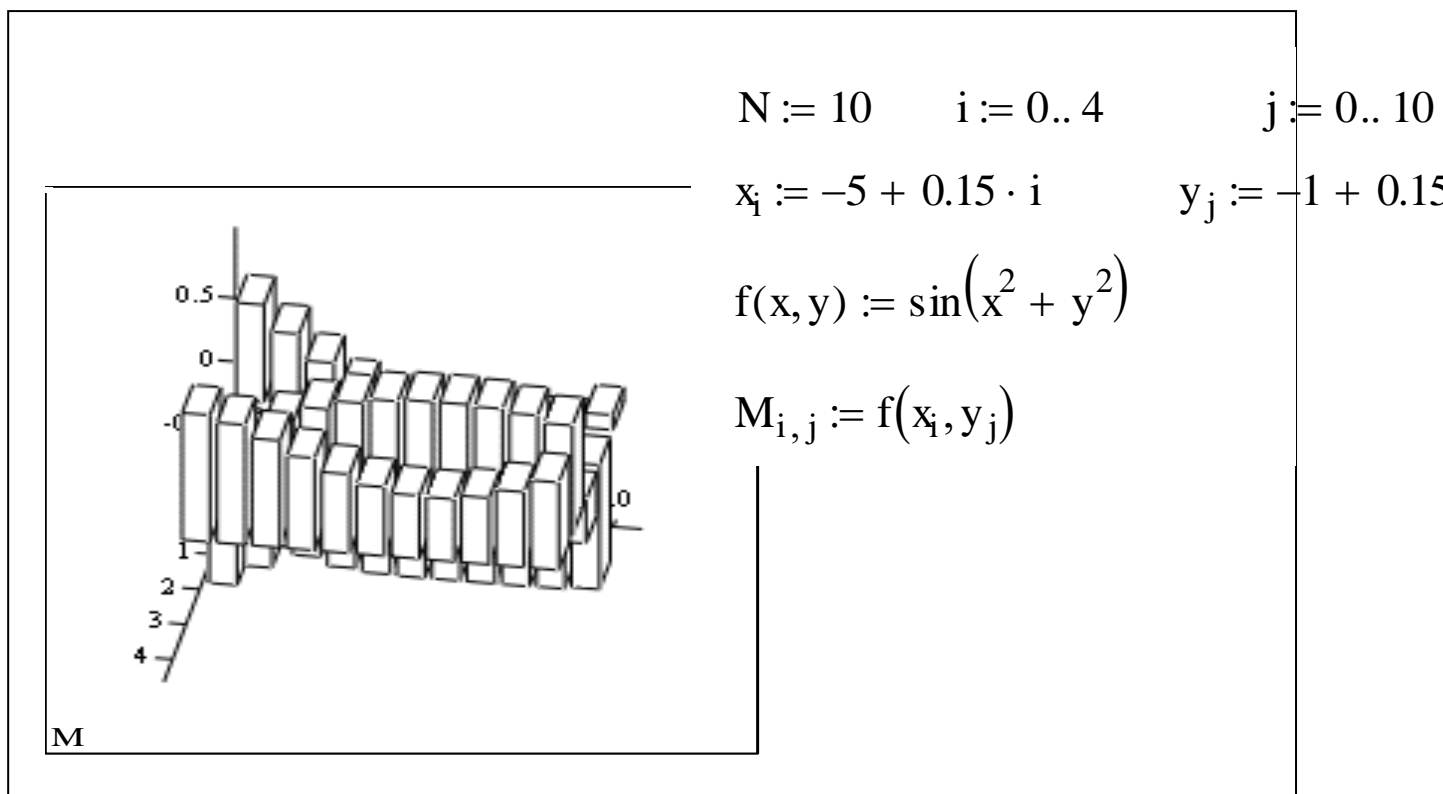
Чтобы создать гистограмму:

- ◆ Определите матрицу значений, которую необходимо представить в графическом виде. Mathcad будет использовать номера строк и столбцов матрицы в качестве абсцисс и ординат. Элементы матрицы будут представлены на графике как столбики с соответствующими высотами относительно плоскости ху.
- ◆ Выберите **Трехмерная гистограмма** из меню **Графика**.
- ◆ Напечатайте имя матрицы в поле ввода.

1. Построить гистограмму для матрицы

$$M := \begin{pmatrix} 56 & 1 & 78 \\ -9.8 & 4 & -9 \\ 7 & 3 & 0 \end{pmatrix}$$

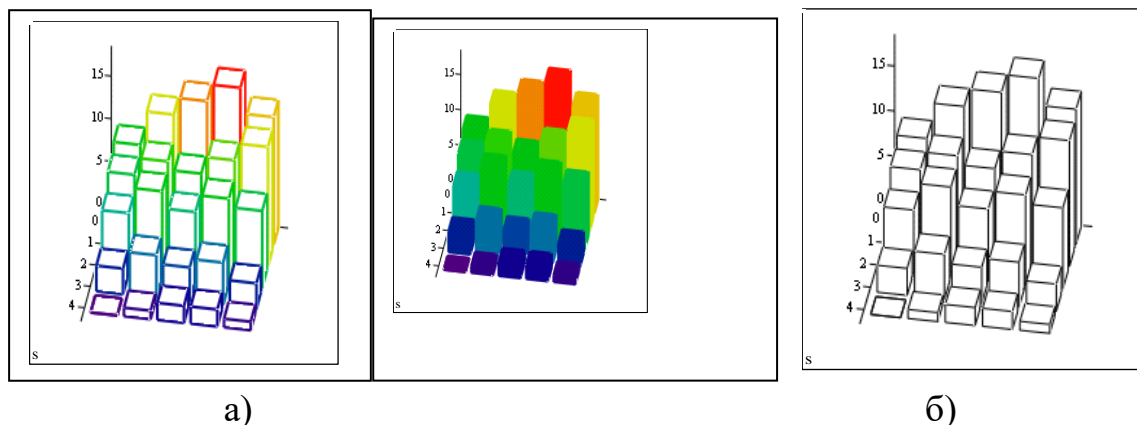
2. Выполнить пункты 2 – 10 (см. начало лабораторной работы).
3. Выполнить пример:



4. Щелкнуть на графике правой кнопкой мыши. **Формат** → закладка **Общие**. В разделе **Вид** изменять параметры **Вращение, наклон, Искривление, Масштаб**, нажимая после изменений клавишу **Применить**.
5. Построить гистограмму:

$$s := \begin{pmatrix} 10 & 14 & 16 & 18 & 15 \\ 9 & 11 & 10 & 12 & 14 \\ 7 & 10 & 8 & 10 & 9 \\ 3 & 5 & 4 & 5 & 3 \\ 0 & 1 & 2 & 2 & 1 \end{pmatrix}$$

6. Преобразовать ее в гистограммы вида (на рабочем листе должны быть размещены все три варианта):



7. Для диаграммы а) вставить название «Трехмерная гистограмма» сверху; для диаграммы б) – снизу.

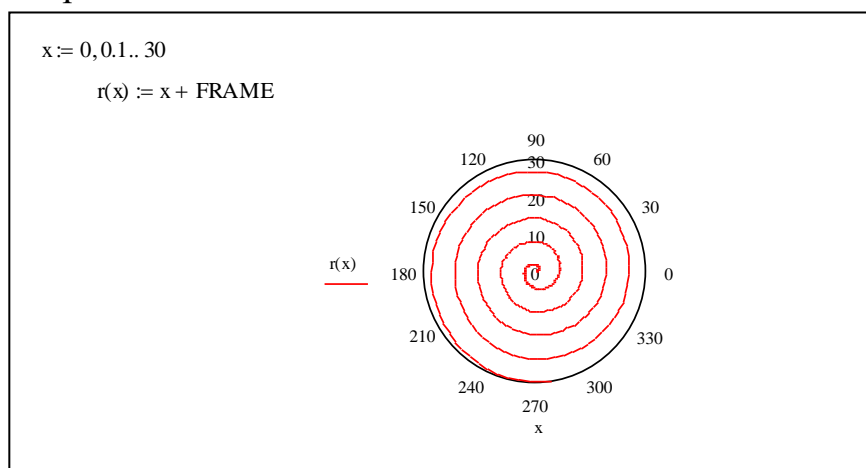
8. Для диаграммы а): **Формат** → закладка **Специальный**. В разделе **Вид диаграммы** применять опции **Матрица, Наложение, Бок о бок**.

АНИМАЦИЯ

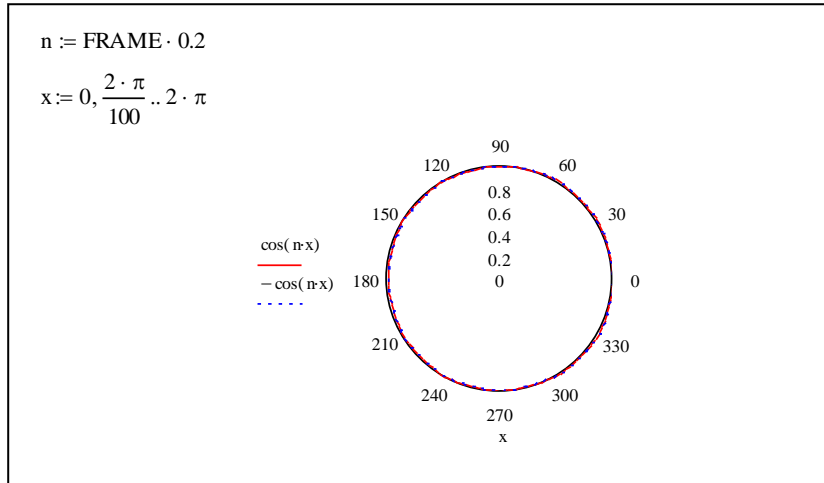
Mathcad имеет встроенную переменную FRAME, чье единственное назначение – управление анимациями. Этапы создания любой анимации следующие:

- ◆ Создайте объект, чей вид зависит от значения переменной FRAME. Этот объект не обязан быть графиком, он может быть чем угодно.
- ◆ Выберите **Вид** → **Анимация** для вызова диалогового окна анимации.
- ◆ Заключите в выделяющий прямоугольник ту часть рабочего документа, которую нужно анимировать.
- ◆ В диалоговом окне установите верхние и нижние границы переменной FRAME. Переменная FRAME будет увеличиваться с единичным шагом от нижней границы до верхней.
- ◆ В поле **Скорость** введите значение скорости воспроизведения.
- ◆ Нажмите клавишу **Анимация** в диалоговом окне. Выбранное изображение возникнет в миниатюрном виде внутри диалогового окна.
- ◆ Чтобы воспроизвести анимацию, нажмите крайнюю левую клавишу со стрелкой в появившемся диалоговом окне **Playback**.

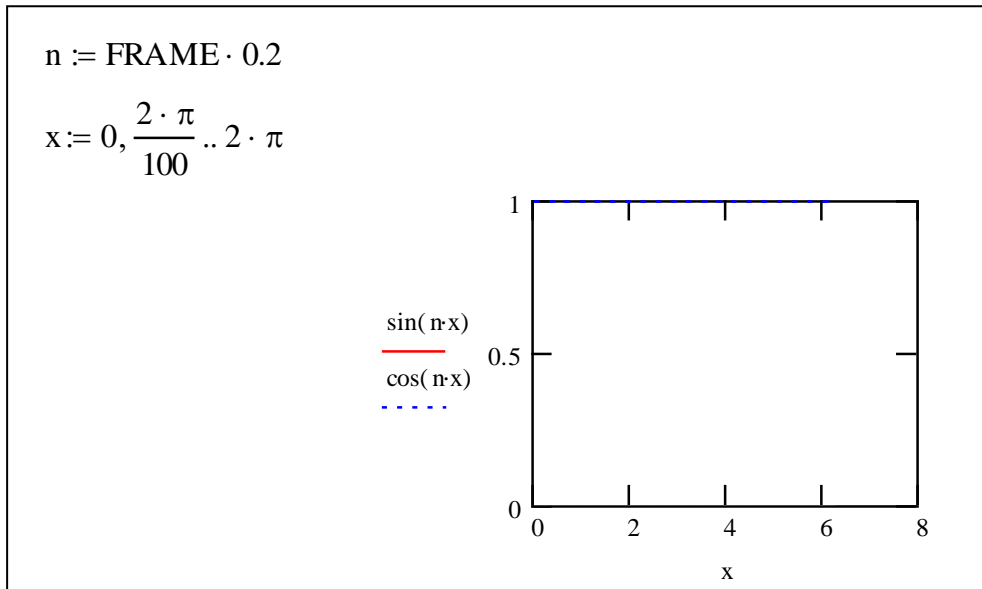
Выполнить пример: создать полярный график и для него анимацию: воспроизвести для FRAME от 1 до 50:



Создать анимацию (рост четырехлистного клевера);



Создать анимацию (американские горы); воспроизвести для FRAME от 1 до 100:



Тема 3. Численные методы решения нелинейных уравнений

Метод бисекции (деления отрезка пополам)

Метод бисекции или *метод деления отрезка пополам* — простейший численный метод для решения нелинейных уравнений вида $F(x)=0$. Предполагается только непрерывность функции $F(x)$.

Задача заключается в нахождении корней нелинейного уравнения

$$F(x) = 0 \quad (3.1)$$

Для начала итераций необходимо знать интервал $[x_L, x_R]$ значений x , где находится единственный корень. Произведение значений функции на краях этого интервала получится меньше нуля:

$$F(x_L)F(x_R) < 0 \quad (3.2)$$

То есть функция меняет знак на данном интервале. Выберем точку внутри интервала

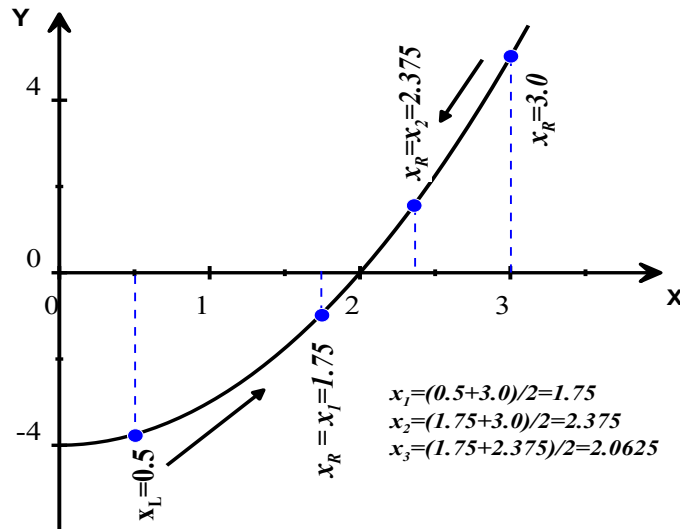


Рис. 3.1 Графическое представление метода бисекций (деления отрезка пополам)

$$x_M = \frac{(x_R + x_L)}{2} \quad (3.3)$$

Разобьём этот интервал на два $[x_L, x_M]$ и $[x_M, x_R]$. Теперь найдём новый интервал, в котором функция меняет знак. Пусть $F(x_L)F(x_M) < 0$ и соответственно корень находится внутри интервала $[x_L, x_M]$. Тогда обозначим $x_R = x_M$ и повторим описанную процедуру до достижения требуемой точности. За количество итераций N первоначальный отрезок делится в 2^N раз. На рисунке 1.1 приведено графическое представление данного метода.

Ниже приведена программная реализация данного численного метода в пакете MathCAD:

```

HalfSegmentDiv(F, x0, x1, e) :=
  ∞ if F(x0) F(x1) > 0
  otherwise
    while |x0 - x1| ≥ e
      x ← (x0 + x1) / 2
      x0 ← x if F(x) F(x0) > 0
      x1 ← x otherwise
    x

```

Данная функция имеет следующие входные параметры:

- F – нелинейная функция $F(x)$;
- x_0 – начало интервала на котором производится поиск решения уравнения $F(x)=0$;

- x_1 – конец интервала на котором производится поиск решения уравнения $F(x)=0$;

- ϵ – точность решения

В случае если на данном интервале решения не обнаружено, функция возвращает ∞ .

Метод хорд (метод линейной интерполяции)

Приведём метод, основанный на нахождении x_{i+1} по двум предыдущим приближениям x_i и x_{i-1} с помощью линейной интерполяции, называемый *методом хорд*.

Идея метода состоит в том, что по двум точкам $M_{i-1}(x_{i-1}; F(x_{i-1}))$ и $M_i(x_i; F(x_i))$ построить прямую $M_{i-1}M_i$ (то есть хорду, соединяющую две точки графика $y = F(x)$) и взять в качестве следующего приближения x_{i-1} абсциссу точки пересечения этой прямой с осью Ox . Иными словами, приближённо заменить на этом шаге функцию $F(x_i)$ её *линейной интерполяцией*, найденной по двум значениям x : x_{i-1} и x_i . (*Линейной интерполяцией* функции $F(x)$ назовём такую линейную функцию $L(x)$, значения которой совпадают со значениями $F(x)$ в двух фиксированных точках, в данном случае – в точках x_{i-1} и x_i).

В зависимости от того, лежат ли точки x_{i-1} и x_i по разные стороны от корня x^* или же по одну и ту же сторону, получаем следующие графические представления:

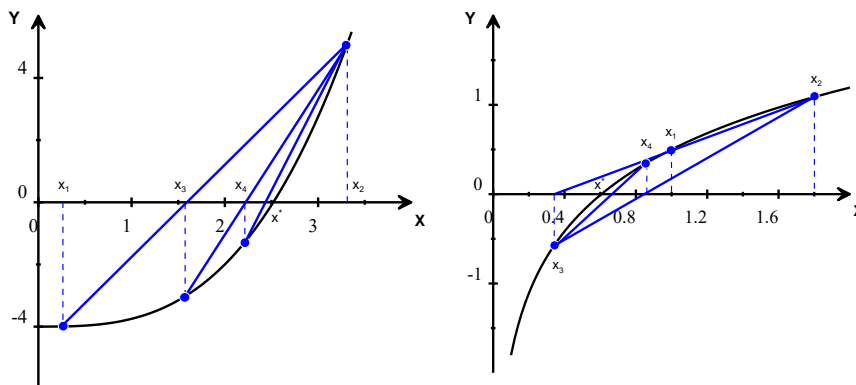


Рис.3.2. Построение последовательного приближения по методу хорд: два случая

Итак, очередное последовательное приближение будет зависеть от двух предыдущих: $x_{i+1} = \varphi(x_{i-1}, x_i)$. Найдём выражение для функции φ . Интерполяционную линейную функцию $L(x)$ будем искать как функцию с угловым коэффициентом, равным разностному отношению:

$$k_i = \frac{F(x_i) - F(x_{i-1})}{x_i - x_{i-1}}, \quad (3.4)$$

построенному для отрезка между x_{i-1} и x_i , график которой проходит через точку M_i :

$$L(x) = F(x_i) + \frac{F(x_i) - F(x_{i-1})}{x_i - x_{i-1}} (x - x_i). \quad (3.5)$$

Решая уравнение $L(x) = 0$, находим

$$x_{i+1} = \frac{x_{i-1}F(x_i) - x_iF(x_{i-1})}{F(x_i) - F(x_{i-1})} = x_i - \frac{F(x_i)}{\frac{F(x_i) - F(x_{i-1})}{x_i - x_{i-1}}}, \quad (3.6)$$

то есть

$$x_{i+1} = \frac{x_{i-1}F(x_i) - x_iF(x_{i-1})}{F(x_i) - F(x_{i-1})} = x_i - \frac{F(x_i)}{k_i}. \quad (3.7)$$

Заметим, что величина k_i может рассматриваться как разностное приближение для производной $F'(x)$ в точке x_i . Тем самым полученная формула (3.7) – это разностный аналог итерационной формулы метода Ньютона.

Вычисление по формуле вычисления ведутся по формуле (3.7) при $i=1,2,3\dots$, начиная с двух приближений x_0 и x_1 , взятых, по возможности, поближе к корню x^* . При этом не предполагается, что x^* лежит между x_0 и x_1 (и что значения функции F в точках x_0 и x_1 имеют разные знаки). При этом не гарантируется, что корень попадёт на отрезок между x_{i-1} и x_i на каком-либо следующем шаге (хотя это и не исключено). В таком случае затруднительно дать оценку погрешности, с которой x_{i+1} приближает истинное значение корня x^* , поэтому довольствуются таким эмпирическим правилом: вычисления прекращают, когда будет выполнено неравенство $|x_{i+1} - x_i| < \varepsilon$, где ε – желаемая точность нахождения корня. При этом полагают приближённое значение корня равным $\tilde{x} = x_{i+1}$.

Приведем программную реализацию метода хорд в пакете MathCAD:

```

ChordsMethod(F, x0, x1, e) :=
  x0 ← x0
  x1 ← x1
  i ← 1
  while |x_{i-1} - x_i| > e
    k ← (F(x_i) - F(x_{i-1})) / (x_i - x_{i-1})
    x_{i+1} ← x_i - F(x_i) / k
    i ← i + 1
  x_i

```

Данная функция имеет следующие входные параметры:

- F – нелинейная функция $F(x)$;
- x_0 – начало интервала на котором производится поиск решения уравнения $F(x)=0$;
- x_1 – конец интервала на котором производится поиск решения уравнения $F(x)=0$;
- e – точность решения

В случае если на данном интервале решения не обнаружено, функция возвращает ∞ .

Метод касательных (метод Ньютона)

Рассмотрение метода одной касательной позволяет предположить, что итерации станут приближаться к корню ещё быстрее, если мы будем выбирать касательную вместо секущей не только на первом, а на каждом шаге. Ясно, что тогда формула итераций будет иметь вид

$$x_{i+1} = x_i - \frac{F(x_i)}{F'(x_i)} \quad (3.8)$$

Этот метод называется *методом касательных*, или *методом Ньютона*. Действительно, последовательные приближения метода Ньютона сходятся гораздо быстрее, чем в общем методе итераций (скорость сходимости приближений в котором, напомним, та же, что у геометрической прогрессии со знаменателем γ при $0 < \gamma < 1$).

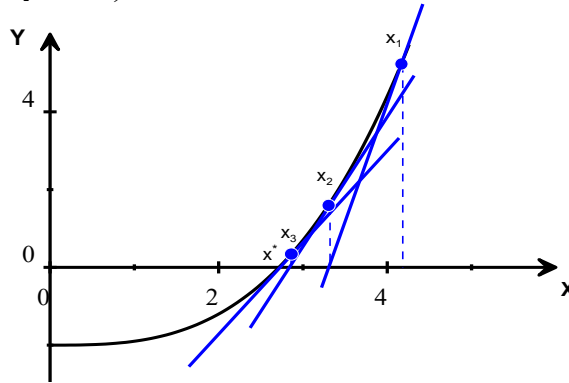


Рис.3.3. Последовательные приближения метода Ньютона (касательных)

Геометрический смысл метода Ньютона состоит в том, что на каждом шаге мы строим касательную к графику в точке очередного последовательного приближения x_i , а за следующее приближение x_{i+1} берём точку пересечения этой касательной с осью Ox . Тем самым наклон прямой подстраивается на каждом шаге наилучшим образом (ведь кривизну графика, связанную со второй производной, мы не учитываем, и поэтому неизвестно, в какую сторону от касательной отклонится график).

Заметим, что по-другому идею метода Ньютона мы можем описать так: на каждом шаге вместо исходного уравнения $F(x)=0$ мы решаем приближённое, *линеаризованное* в точке x_i уравнение:

$$F(x_i) + F'(x_i)(x - x_i) = 0, \quad (3.9)$$

в котором левая часть – это многочлен Тейлора первого порядка для функции $F(x)$ в точке x_i , то есть линейная функция

$$L_{x_i}(x) = F(x_i) + F'(x_i)(x - x_i). \quad (3.10)$$

Решением линеаризованного уравнения $L_{x_i}(x) = 0$ служит следующее приближение x_{i+1} , в то время как решением исходного точного уравнения $F(x)$ служит искомый корень x^* .

Приведем программную реализацию метода касательных или метода Ньютона в пакете MathCAD:

```

NewtonMethod(F, x0, e) :=
  x ← x0
  F1(x) ←  $\frac{d}{dx} F(x)$ 
  y ←  $\frac{F(x)}{F1(x)}$ 
  while |y| > e
    x ← x - y
    y ←  $\frac{F(x)}{F1(x)}$ 
  x

```

Данная функция имеет следующие входные параметры:

- F – нелинейная функция $F(x)$;
- x_0 – начальное приближение поиска решения уравнения $F(x)=0$;
- e – точность решения

Метод секущих

В качестве функции $\lambda(x)$ берут любую постоянную λ_0 , знак которой совпадает со знаком производной $F'(x)$ в окрестности E (и, в частности, на отрезке, соединяющем x_0 и x^*). Постоянная λ_0 не зависит также и от номера шага i . Тогда формула итераций оказывается очень проста:

$$x_{i+1} = x_i - \lambda_0 F(x_i),$$

и на каждой итерации нужно один раз вычислить значение функции $F(x)$.

Выясним смысл этой формулы, а также смысл условия о совпадении знаков $F'(x)$ и λ_0 . Рассмотрим прямую, проходящую через точку $(x_i; F(x_0))$ на графике $y = F(x)$ с угловым коэффициентом $\operatorname{tg} \alpha = \frac{1}{\lambda_0}$. Тогда уравнением этой прямой будет:

$$y = F(x_i) + \frac{1}{\lambda_0} (x - x_i).$$

Найдём точку пересечения этой прямой с осью Ox из уравнения

$$F(x_i) + \frac{1}{\lambda_0} (x - x_i) = 0,$$

откуда $x = x_i - \lambda_0 F(x_i)$. Следовательно, эта прямая пересекает ось Ox как раз в точке следующего приближения. Тем самым получаем следующую геометрическую интерпретацию последовательных приближений. Начиная с точки x_0 , через соответствующие точки графика $y = F(x)$ проводятся секущие с угловым коэффициентом $k = \frac{1}{\lambda_0}$ того же знака, что производная $F'(x)$. (Заметим, что, во-первых, значение производной вычислять не обязательно, достаточно лишь знать, убывает функция $F(x)$ или возрастает; во-вторых, что прямые, проводимые при разных x_i , имеют один и тот же угловой коэффициент k и, следовательно, параллельны друг другу.) В качестве следующего приближения к корню берётся точка пересечения построенной прямой с осью Ox .

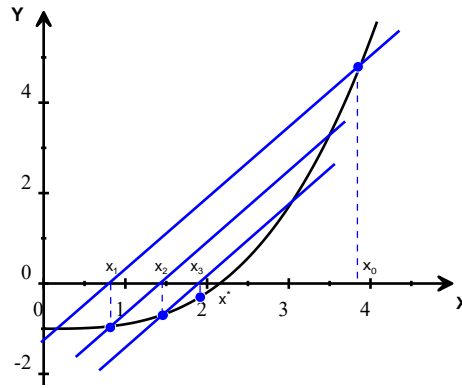


Рис. 3.4. Графическая интерпретация метода секущих

На рисунке 1.4 изображены итерации при $F'(x) > 0$, в случае $k = \frac{1}{\lambda_0} > F'(x_0)$. Мы видим, что последовательные точки x_i приближаются к корню, оставаясь всё время с одной стороны от него.

Программная реализация данного метода в пакете MathCAD выглядит так:

```
SectantMethod(F, x0, x1, e) :=
  while |x1 - x0| > e
    x ← x1 - (F(x1)(x1 - x0) / (F(x1) - F(x0)))
    x0 ← x1
    x1 ← x
  x
```

Данная функция принимает следующие параметры:

- F – нелинейная функция $F(x)$;
- x_0 – начало интервала на котором производится поиск решения уравнения $F(x)=0$;
- x_1 – конец интервала на котором производится поиск решения уравнения $F(x)=0$;
- e – точность решения

Метод простых итераций

Предположим, что уравнение $F(x)=0$ при помощи некоторых тождественных преобразований приведено к виду $F(x)=x$.

Заметим, что такое преобразование можно вести разными способами, и при этом будут получаться разные функции $\varphi(x)$ в правой части уравнения. Уравнение $F(x)=0$ эквивалентно уравнению $x = \varphi(x)$ при любой функции $\varphi(x) \neq 0$. Таким образом, можно взять $\varphi(x) = x + \lambda(x)F(x)$ и при этом выбрать функцию (или постоянную) $\lambda \neq 0$ так, чтобы функция $\varphi(x)$ удовлетворяла тем свойствам, которые понадобятся нам для обеспечения нахождения корня уравнения.

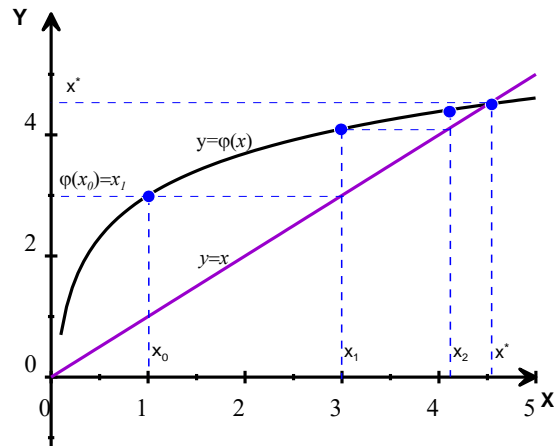


Рис. 3.5. Точка x^* – решение уравнения $x = \varphi(x)$

Для нахождения корня уравнения $x = \varphi(x)$ выберем какое-либо начальное приближение x_0 (расположенное, по возможности, близко к корню x^*). Далее будем вычислять последующие приближения $x_1, x_2, \dots, x_i, x_{i+1}$ по формулам

$$x_1 = \varphi(x_0), x_2 = \varphi(x_1), \dots, x_{i+1} = \varphi(x_i), \dots,$$

то есть, используя каждое вычисленное приближение к корню в качестве аргумента функции $\varphi(x)$ в очередном вычислении. Такие вычисления по одной и той же формуле $x_{i+1} = \varphi(x_i)$, когда полученное на предыдущем шаге значение используется на последующем шаге, называются *итерациями*. *Итерациями* называют часто и сами значения x_i , полученные в этом процессе (то есть, в нашем случае, последовательные приближения к корню).

Заметим тот факт, что x^* – корень уравнения $x = \varphi(x)$, означает, что x^* есть абсцисса точки пересечения графика $y = \varphi(x)$ с прямой $y = x$. Если же при каком-либо x_0 вычислено значение $x_1 = \varphi(x_0)$, и взято в качестве нового аргумента функции, то это означает, что через точку графика $(x_0; \varphi(x_0))$ проводится горизонталь до прямой $y = x$, а оттуда опускается перпендикуляр на ось Ox . Там и будет находиться новый аргумент x_1 .

Замечание.

Для обеспечения сходимости в методе простой итерации функцию $\varphi(x)$ иногда получают так:

$$\varphi(x) = x - \frac{f(x)}{k},$$

считая, что $k \geq Q/2$, где $Q = \max|f(x)'|$.

Число k в промежутке $[a, b]$ имеет тот же знак, что и $f(x)'$.

Ниже приведен программный код метода простых итераций написанный средствами программирования пакета MathCAD.

$$\text{SingleIterationMethod}(F, x_0, N, e) := \begin{cases} x \leftarrow F(x_0) \\ i \leftarrow 0 \\ \text{while } i \leq N \wedge |F(x) - x| > e \cdot |x| \\ \quad \begin{cases} i \leftarrow i + 1 \\ x \leftarrow F(x) \end{cases} \\ x \end{cases}$$

Данная функция имеет следующие входные параметры:

- F – нелинейная функция $F(x)$;
- x_0 – начальное приближение при поиске решения уравнения $F(x)=0$;
- N – число итераций процесса;
- e – точность решения.

Задача 1.1

Площадь кругового сегмента, дуга которого α , определяется формулой:

$$Q = \frac{1}{2} \pi R^2 (\alpha - \sin \alpha)$$

где α – радиальная мера дуги. Необходимо найти сегмент, площадь которого равна $1/5$ площади круга, то есть найти угловую меру дуги сегмента.

Площадь круга определяется по формуле:

$$S = \pi R^2,$$

Из условий получаем равенство:

$$\frac{1}{5} \pi R^2 = \frac{1}{2} \pi R^2 (\alpha - \sin \alpha)$$

Таким образом, нам необходимо решить нелинейное уравнение:

$$\frac{1}{5} \pi R^2 - \frac{1}{2} \pi R^2 (\alpha - \sin \alpha) = 0$$

Чтобы решить данное уравнение методом простых итераций требуется привести его к виду $F(x) = x$

$$\frac{1}{5} \pi R^2 - \frac{1}{2} \pi R^2 \alpha + \frac{1}{2} \pi R^2 \sin \alpha = 0$$

Поделим последнее уравнение на πR^2 (поскольку выражение πR^2 не может равняться нулю), получим

$$\frac{1}{5} - \frac{1}{2} \alpha + \frac{1}{2} \sin \alpha = 0.$$

Выразим из этого выражения α для метода простых итераций:

$$\alpha = 0.4 + \sin \alpha, \text{ то есть } \alpha = \phi(\alpha)$$

На рисунках 1.6-1.7 приведены графические решения для данной задачи

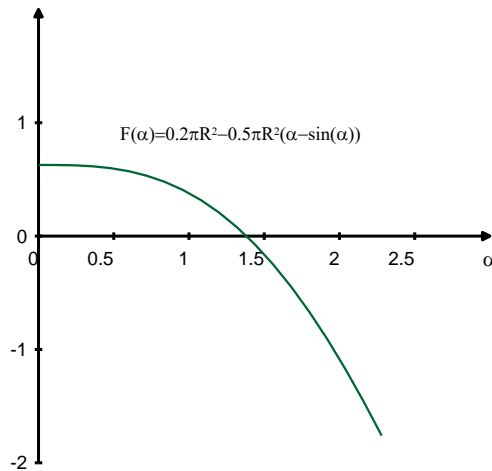


Рис. 3.6 График функции $F(\alpha) = \frac{1}{5}\pi R^2 - \frac{1}{2}\pi R^2(\alpha - \sin \alpha)$

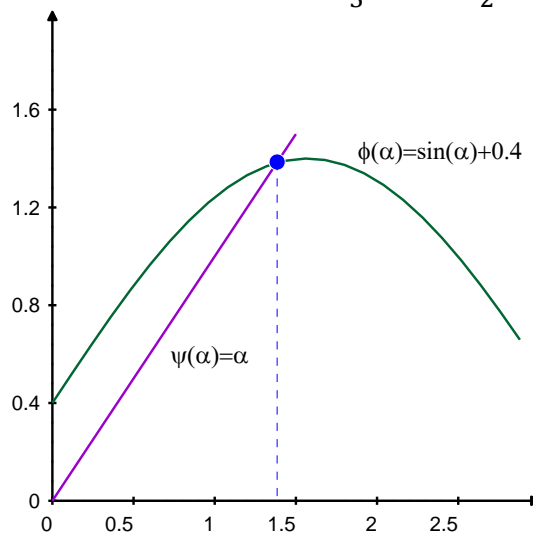


Рис. 3.7. График функции $F_1(\alpha) = \varphi(\alpha) = 0.4 + \sin \alpha$ для решения методом простых итераций

Сравним результаты, полученные путем решения собственными средствами пакета MathCAD для данной задачи, с решениями, полученными разными методами решения:

$R := 1$ Пусть радиус равен единице.

$$F(\alpha) := 0.2\pi R^2 - 0.5\pi \cdot R^2(\alpha - \sin(\alpha))$$

Функция, корни которой необходимо найти.

$$F1(\alpha) := \sin(\alpha) + 0.4$$

Функция для решения задачи методом простых итераций.

Получим решение стандартными средствами MathCAD:

[Оглавление](#)

$$\alpha_0 := \text{root}(F(\alpha), \alpha, 0, 2\pi)$$

$$\alpha_0 = 1.3823$$

Вызываем функцию для решения уравнений методом деления отрезка пополам:

$$\alpha_1 := \text{HalfSegmentDiv}(F, 0, 3, 0.001)$$

$$\alpha_1 = 1.3821$$

А теперь для решения уравнений методом деления хорд:

$$\alpha_2 := \text{ChordsMethod}(F, 0.02, 0.05, 0.001)$$

$$\alpha_2 = 1.3823$$

Используем метод Ньютона (метод касательных):

$$\alpha_3 := \text{NewtonMethod}(F, 0.01, 0.001)$$

$$\alpha_3 = 1.3823$$

Вызываем функцию для решения уравнений методом секущих:

$$\alpha_4 := \text{SectantMethod}(F, 0, 3, 0.001)$$

$$\alpha_4 = 1.3823$$

Используем метод простых итераций (используем подготовленную ранее функцию F1)

$$\alpha_5 := \text{SingleIterationMethod}(F1, 0.02, 20, 0.001)$$

$$\alpha_5 = 1.3816$$

Итак, решение найдено: угловая мера дуги сегмента, для которого площадь составляет пятую часть площади круга, равна 1.3816 радиан или 79.199°.

Задача 1.2

Решить уравнение $\cos(x + 0.5) = x^3$.

Методы решения нелинейных уравнений написаны таким образом, что одним из параметров процедуры является функция $F(x)$. Таким образом, мы можем передавать в процедуру метода расчета произвольную функцию. В нашем случае это функция:

$$F(x) = x^3 - \cos(x + 0.5).$$

После некоторых преобразований получаем функцию для решения методом простых итераций:

$$F_1(x) = \sqrt[3]{\cos(x + 0.5)}.$$

Графики этих функций приведены на рисунках 1.8 и 1.9:

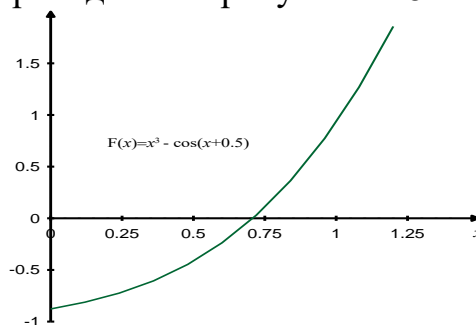


Рис. 3.8. График функции $F(x) = x^3 - \cos(x + 0.5)$

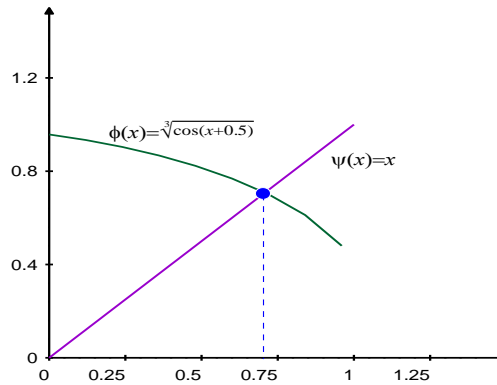


Рис. 3.9. График функции $F_1(x) = \sqrt[3]{\cos(x+0.5)}$ для решения методом простых итераций

Приведем сравнение для решения данного уравнения разными методами и внутренними средствами пакета MathCAD:

$$W(x) := x^3 - \cos(x + 0.5)$$

Функция,
корни которой
необходимо
найти.

$$W1(x) := \left(\sqrt[3]{\cos(x + 0.5)}\right)$$

Функция для
решения
задачи
методом
простых
итераций.

Получим решение стандартными средствами MathCAD:

$$x_0 := \text{root}(W(x), x, 0, \pi)$$

$$x_0 = 0.7080$$

Вызываем функцию для решения уравнений методом деления отрезка пополам:

$$x_1 := \text{HalfSegmentDiv}(W, 0, 3, 0.001)$$

$$x_1 = 0.7083$$

А теперь для решения уравнений методом хорд:

$$x_2 := \text{ChordsMethod}(W, 0.01, 0.05, 0.001)$$

$$x_2 = 0.7080$$

Используем метод Ньютона (метод касательных):

$$x_3 := \text{NewtonMethod}(W, 0.01, 0.001)$$

$$x_3 = 0.7082$$

Используем метод секущих:

$$x_4 := \text{SectantMethod}(W, 0, 3, 0.001)$$

$$x_4 = 0.7080$$

Используем метод простых итераций (используем подготовленную ранее функцию F1)

$$x_5 := \text{SingleIterationMethod}(W1, 0.02, 40, 0.001)$$

$$x_5 = 0.7084$$

Тема 4. Численные методы решения систем линейных алгебраических уравнений

Материалы к лекциям данного раздела сформированы на основании следующих источников:

1. Задачи по линейной алгебре. Часть 1: Учебно-методическое пособие / Под общей редакцией Е. И. Анно — М.: Экономический факультет МГУ имени М.В. Ломоносова, 2020. — 169 с. ISBN 978-5-906932-53-2

Метод Гаусса (метод последовательного исключения неизвестных)

Пусть дана система m линейных уравнений с n неизвестными $Ax = b$. Требуется найти ее общее решение, если она совместна, или установить ее несовместность. Выпишем расширенную матрицу системы

$$A^* = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{pmatrix}$$

Назовем элементарными операциями следующие действия с матрицами:

- 1) перестановка строк;
- 2) умножение строки на число, отличное от нуля;
- 3) сложение строки с другой строкой, умноженной на число.

Отметим, что при решении системы уравнений, в отличие от вычисления определителя и нахождения ранга, нельзя оперировать со столбцами.

Легко проверить, что если по матрице, полученной из A^* выполнением элементарной операции, восстановить систему уравнений, то новая система будет равносильна исходной.

Цель алгоритма – с помощью применения последовательности элементарных операций к матрице A^* добиться, чтобы каждая строка, кроме, быть может, первой, начиналась с нулей, и число нулей до первого ненулевого элемента в каждой следующей строке было больше, чем в предыдущей.

Шаг алгоритма заключается в следующем. Находим первый ненулевой столбец в матрице A^* . Пусть это будет столбец с номером i . Находим в нем ненулевой элемент и строку с этим элементом меняем местами с первой строкой. Чтобы не нагромождать дополнительных обозначений, будем считать, что такая смена строк в матрице A^* уже произведена, то есть $a_{1i} \neq 0$. Тогда ко второй строке прибавим первую, умноженную на число $-\frac{a_{2i}}{a_{1i}}$, к третьей строке прибавим первую, умноженную на число $-\frac{a_{3i}}{a_{1i}}$, и т.д. В результате получим матрицу

$$A_1^* = \begin{pmatrix} 0 & \cdots & 0 & a_{1i} & a_{1i+1} & \cdots & a_{1n} & b_1 \\ 0 & \cdots & 0 & 0 & a_{2i+1}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & a_{mi+1}^{(1)} & \cdots & a_{mn}^{(1)} & b_m^{(1)} \end{pmatrix}$$

(Первые нулевые столбцы, как правило, отсутствуют.)

Если в матрице A_1^* встретилась строка с номером k , в которой все элементы $a_{kj}^{(1)} = 0$ равны нулю, а $b_k^{(1)} \neq 0$, то выполнение алгоритма останавливаем и делаем вывод, что система несовместна. Действительно, восстанавливая систему уравнений по расширенной матрице, получим, что k -е уравнение будет иметь вид

$$0 \cdot x_1 + 0 \cdot x_2 + \dots + 0 \cdot x_n = b_k^{(1)}$$

Этому уравнению не удовлетворяет ни один набор чисел x_1, x_2, \dots, x_n .

Матрицу A_1^* можно записать в виде:

$$A_1^* = \begin{pmatrix} 0 & \dots & 0 & a_{1i} & a_{1i+1} & \dots & a_{1n} & b_1 \\ 0 & \dots & 0 & 0 & & & & \\ \dots & \dots & \dots & \dots & & & & \\ 0 & \dots & 0 & 0 & & & & \end{pmatrix},$$

где

$$B^* = \begin{pmatrix} a_{2i+1}^{(1)} & a_{2i+1}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ a_{3i+1}^{(1)} & a_{3i+1}^{(1)} & \dots & a_{3n}^{(1)} & b_3^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ a_{mi+1}^{(1)} & a_{mi+1}^{(1)} & \dots & a_{mn}^{(1)} & b_m^{(1)} \end{pmatrix}.$$

По отношению к матрице B^* выполняем описанный шаг алгоритма. Получаем матрицу

$$B_1^* = \begin{pmatrix} 0 & \dots & 0 & a_{2j}^{(2)} & a_{2j+1}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & \dots & 0 & 0 & a_{3j+1}^{(2)} & \dots & a_{3n}^{(2)} & b_3^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & a_{mj+1}^{(2)} & \dots & a_{mn}^{(2)} & b_m^{(2)} \end{pmatrix},$$

где $j > i$, $a_{2j}^{(2)} \neq 0$. Эту матрицу снова можно записать в виде

$$B_1^* = \begin{pmatrix} 0 & \dots & 0 & a_{2j}^{(2)} & a_{2j+1}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & \dots & 0 & 0 & & & & \\ \dots & \dots & \dots & \dots & & & & \\ 0 & \dots & 0 & 0 & & & & \end{pmatrix},$$

и к матрице C^* снова применим описанный выше шаг алгоритма.

Процесс останавливается, если после выполнения очередного шага новая уменьшенная матрица состоит из одних нулей или если исчерпаны все строки. Заметим, что заключение о несовместности системы могло остановить процесс и ранее.

Если бы мы не уменьшали матрицу, то в итоге пришли бы к матрице вида

$$\tilde{A}^* = \begin{pmatrix} 0 & \dots & 0 & a_{1i}^{(1)} & \dots & a_{1j-1}^{(1)} & a_{1j}^{(1)} & \dots & a_{1s-1}^{(1)} & a_{1s}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & \dots & 0 & 0 & \dots & 0 & a_{1j}^{(2)} & \dots & a_{1s-1}^{(2)} & a_{1s}^{(2)} & \dots & a_{1n}^{(2)} & b_1^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & a_{rs}^{(r)} & \dots & a_{rn}^{(r)} & b_r^{(r)} \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Далее выполняется так называемый обратный ход метода Гаусса. По матрице \tilde{A}^* составляем систему уравнений. В левой части оставляем неизвестные с номерами, соответствующими первым ненулевым элементам в каждой строке, то есть x_i, x_j, \dots, x_r . Заметим, что ранг \tilde{A}^* равен рангу A . Остальные неизвестные переносим в правую часть. Считая неизвестные в правой части некоторыми фиксированными величинами, несложно выразить через них неизвестные левой части.

Теперь, придавая неизвестным в правой части произвольные значения и вычисляя значения переменных левой части, мы будем находить различные решения исходной системы $Ax = b$. Чтобы записать общее решение, нужно неизвестные в правой части обозначить в каком-либо порядке буквами C_1, C_2, \dots, C_{n-r} , включая и те неизвестные, которые явно не выписаны в правой части из-за нулевых коэффициентов, и тогда столбец неизвестных можно записать в виде столбца, где каждый элемент будет линейной комбинацией произвольных величин C_1, C_2, \dots, C_{n-r} (в частности, просто произвольной величиной C_k). Эта запись и будет общим решением системы. Здесь $r = RgA$.

Если система была однородной, то получим общее решение однородной системы. Коэффициенты при C_1 , взятые в каждом элементе столбца общего решения, составят первое решение из фундаментальной системы решений, коэффициенты при C_2 – второе решение и т.д.

Фундаментальную систему решений однородной системы можно получить и другим способом. Для этого одной переменной, перенесенной в правую часть, нужно присвоить значение 1, а остальным – нули. Вычислив значения переменных в левой части, получим одно решение из фундаментальной системы. Присвоив другой переменной в правой части значение 1, а остальным – нули, получим второе решение из фундаментальной системы и так далее.

Задача 2.1

$$\begin{cases} 0.98x_1 + 0.88x_2 - 0.24x_3 = 1.36 \\ 0.16x_1 - 0.44x_2 - 0.88x_3 = -1.27 \\ 9.74x_1 - 10.0x_2 + 1.74x_3 = 5.31 \end{cases}$$

Решим данную систему уравнений средствами MathCAD:

$$\underline{A} := \begin{pmatrix} 0.98 & 0.88 & -0.24 \\ 0.16 & -0.44 & -0.88 \\ 9.74 & -10.0 & 1.74 \end{pmatrix} \quad B := \begin{pmatrix} 1.36 \\ -1.27 \\ 5.31 \end{pmatrix}$$

Записываем матрицу левой части и вектор правой части системы уравнений.

$X := \text{lsolve}(A, B)$

Получаем решение системы линейных уравнений, используя встроенную функцию MathCAD.

$$X^T = (1.055 \quad 0.719 \quad 1.276)$$

Просматриваем результаты решения системы уравнений.

А теперь напишем функции реализующие решение системы уравнений методом Гаусса-Жордана:

$\text{Exchange}(C, i) :=$	$k \leftarrow i + 1$ while $C_{k,i} = 0$ $k \leftarrow k + 1$ for $j \in 0 \dots \text{rows}(C) - 1$ $S \leftarrow C_{i,j}$ $C_{i,j} \leftarrow C_{k,j}$ $C_{k,j} \leftarrow S$	<p>Функция перестановки i-й и $i+1$-й строк матрицы:</p> <p>C – входная матрица,</p> <p>i – номер строки для перестановки.</p>
	C	

Следующая функция формирует расширенную матрицу системы C , которая является результатом склеивания матриц A и B , и приводит матрицу C к ступенчатой форме.

```

Simplex(A,B) :=
| N ← rows(A)
| C ← augment(A,B)
| for i ∈ 0 .. N - 2
|   | C ← Exchange(C,i) if Ci,i = 0
|   | for j ∈ 0 .. N
|   |   Ci,N-j ←  $\frac{C_{i,N-j}}{C_{i,i}}$ 
|   |   for m ∈ i + 1 .. N - 1
|   |     | α ← Cm,i
|   |     | for j ∈ i .. N
|   |     |   Cm,j ← Cm,j - α · Ci,j
| C

```

Следующая функция осуществляет обратный ход метода Гаусса после приведения матрицы C к ступенчатой форме.

```

Gauss(Simplex,A,B) :=
| C ← Simplex(A,B)
| N ← rows(A) - 1
| VN ←  $\frac{C_{N,N+1}}{C_{N,N}}$ 
| for j ∈ 1 .. N
|   VN-j ←  $\left(\frac{1}{C_{N-j,N-j}}\right) \cdot \left[ C_{N-j,N+1} - \sum_{k=0}^{j-1} [C_{N-j,(N-k)} \cdot V_{N-k}] \right]$ 
| V

```

X1 := Gauss(Simplex,A,B)

X1^T = (1.055 0.719 1.276)

Теперь вызываем нашу реализацию метода Гаусса. Просматриваем результаты решения системы уравнений.

Как видим результаты вычислений встроенными средствами MathCAD и результаты, полученные на основе представленной реализации метода Гаусса-Жордана, совпали.

Тема 5. Моделирование и численные методы решения обыкновенных дифференциальных уравнений

Материалы к лекциям данного раздела сформированы на основании следующих источников:

1. Численный анализ и оптимизация /В.М.Волков, О.Л.Зубко, И.Н.Катковская, И.Л.Ковалева, В.Г.Кротов, П.Лима. – Минск: РУП «Белгослес», 2017-2017 с.

Примеры задач, приводящих к решению обыкновенных дифференциальных уравнений (ОДУ).

Пример 1. Маятник под действием силы тяжести

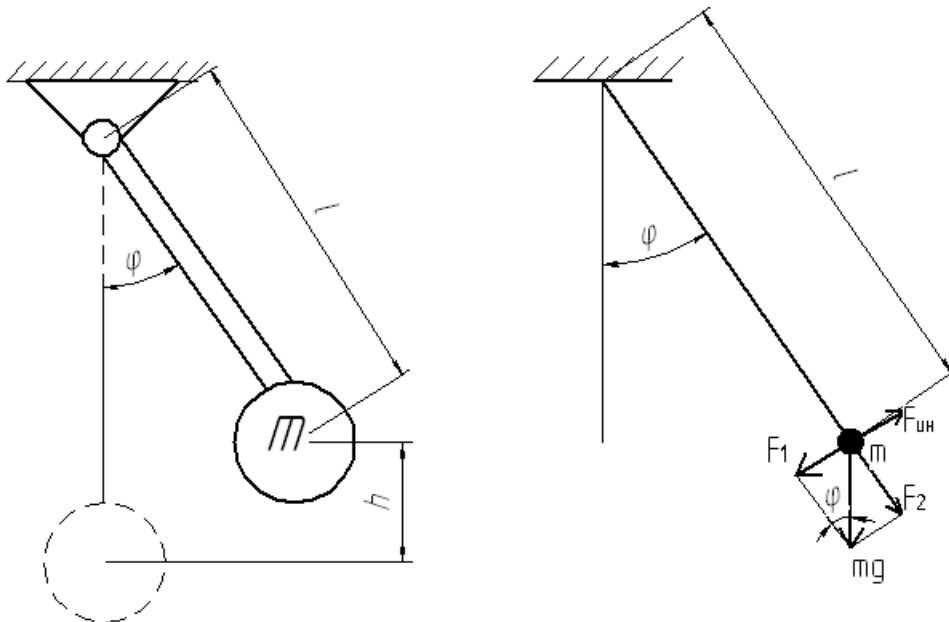


Рисунок 5.1. Маятник под действием силы тяжести. Схема колебательной системы.

Спроецируем обе силы по направлению касательной к траектории движения точки.

$$\begin{aligned}
 F_1 &= F_{ун} \\
 F_1 &= m \cdot g \cdot \sin(\phi) \\
 F_{ун} &= -\varepsilon \cdot l \cdot m = -\ddot{\phi} \cdot l \cdot m \\
 m \cdot g \cdot \sin(\phi) + \ddot{\phi} \cdot l \cdot m &= 0 \\
 \ddot{\phi} + \frac{g}{l} \cdot \sin(\phi) &= 0
 \end{aligned}$$

Пусть $\varphi \approx \sin(\varphi)$, то есть $\varphi \ll 1$

$$\ddot{\phi} + \frac{g}{l} \cdot \phi = 0$$

Пример 5. Балка переменного сечения под действием изгибающей нагрузки

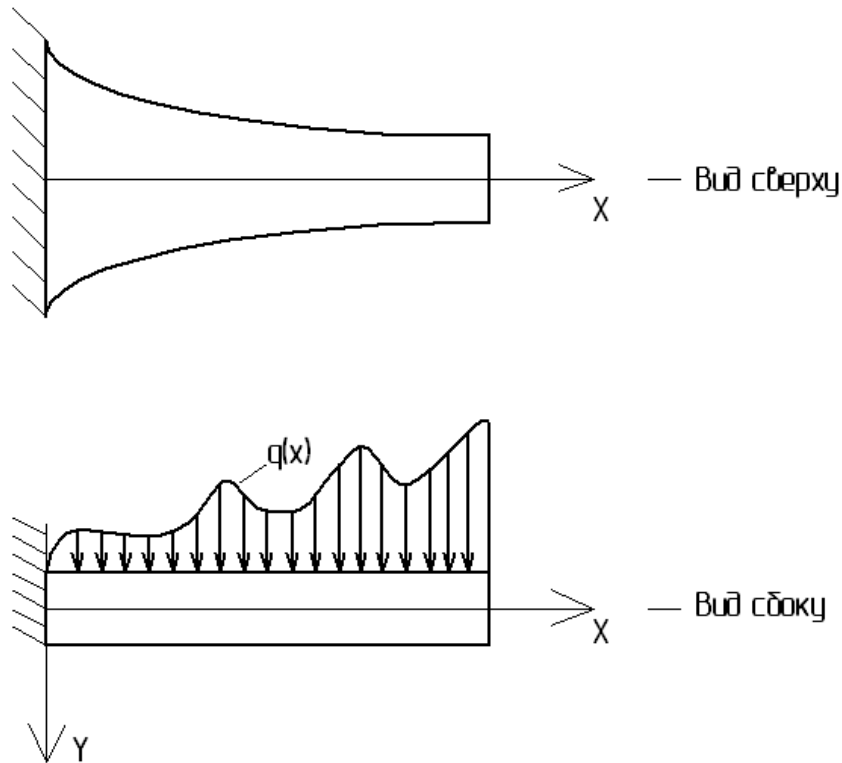


Рисунок 5.5. Балка переменного сечения

Дифференциальное уравнение изогнутой оси балки имеет вид

$$E \cdot J(x) \frac{d^2 y}{dx^2} = -M(x) \quad (5.1)$$

где E – модуль Юнга,

$J(x)$ – момент инерции сечения,

$M(x)$ – изгибающий момент.

$$\frac{dM}{dx} = Q(x) \quad (5.2)$$

$$\frac{dQ(x)}{dx} = -q(x) \quad (5.3)$$

Продифференцируем (1) два раза:

$$E \cdot J' \cdot y'' + E \cdot J \cdot y''' = -Q(x)$$

$$E \cdot (J'' \cdot y'' + 2 \cdot J' \cdot y''' + J \cdot y'''') = q(x)$$

Как видно, получено ОДУ четвертого порядка относительно неизвестной функции $y(x)$. Для того, чтобы однозначно определить это решение, необходимо сформулировать четыре условия для самой функции и её производных.

$$\begin{aligned}y(x)|_{x=0} &= y_0 \\y'(x)|_{x=0} &= y'_0 \\y''(x)|_{x=0} &= y''_0 \\y'''(x)|_{x=0} &= y'''_0\end{aligned}$$

Мы рассмотрим случай, когда значения функции и её производных заданы для начальных значений. Такая задача называется задачей с начальными условиями для ОДУ. Независимая переменная меняется на интервале наблюдения

$$x \in [0; L]$$

Если на правом конце известны какие-нибудь значения функции или её производных, то аналогичные четыре условия будут записаны соответственно на левом и на правом конце (всего четыре условия). Это пример граничной задачи.

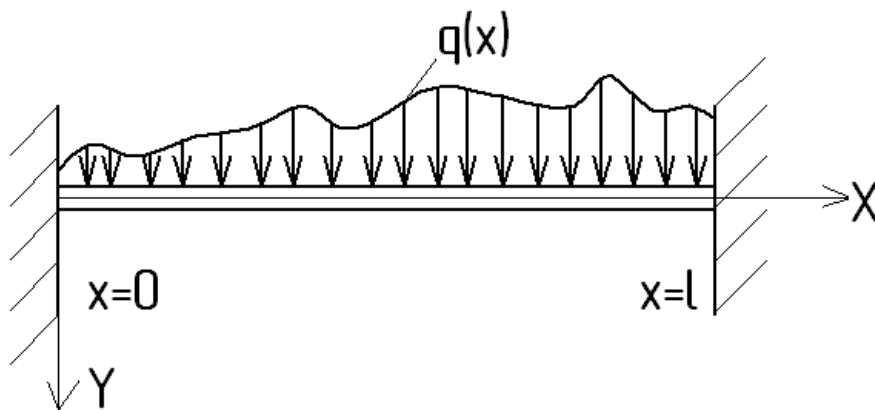


Рисунок 5.3. Схема нагружения балки для граничной задачи.

$$\begin{aligned}y(x)|_{x=0} &= 0 \\y'(x)|_{x=0} &= 0 \\y(x)|_{x=L} &= 0 \\y'(x)|_{x=L} &= 0\end{aligned}$$

Такая задача для ОДУ называется краевой или граничной.

Пример 3. Одномассовая колебательная система.

Брус связан со стеной пружиной с жесткостью k и устройством, рассеивающим энергию (демпфером) с характеристикой s . Брус движется без трения по горизонтальной поверхности под действием силы $P(t)$.

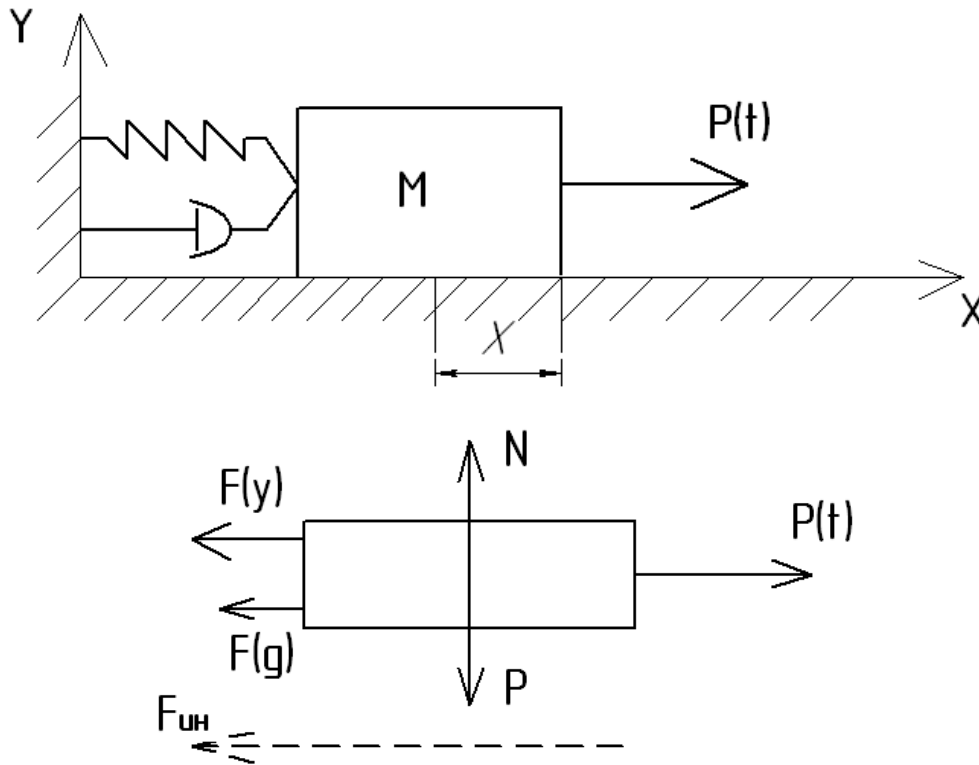


Рисунок 5.4. Колебательная одномассовая система.

Если спроецировать все действующие силы на OX, то

$$p(t) - F_y - F_d - F_u = 0.$$

$x=x(t)$ – функция от времени, описывающая смещение массы относительно положения, в котором она находилась при недеформированном состоянии пружины. Будем считать, что в этом случае наблюдение начинается в момент времени t_0 .

$t \in [t_0; t_k]$ – интервал наблюдения.

$F_y = k \cdot x$ - сила упругости.

$F_d = C \cdot \frac{dx}{dt}$ - сила демпфирования.

$F_u = m \cdot \frac{d^2x}{dt^2}$ - сила инерции.

$$m \cdot \ddot{x} + c \cdot \dot{x} + k \cdot x = p(t)$$

$$\dot{x} = \frac{dx}{dt}$$

$$\ddot{x} = \frac{d^2x}{dt^2}$$

$$x(t)|_{t=0} = x_0$$

$$\dot{x}(t)|_{t=0} = \dot{x}_0$$

Получили ОДУ второго порядка с заданными начальными условиями.

Приведение системы ОДУ произвольного порядка к каноническому виду

Задачу вида

$$\begin{cases} \frac{dz}{dx} = f(x, z(x)) \\ z(x)|_{x=x_0} = z_0 \end{cases}$$

называют задачей Коши.

Пример задачи Коши:

$$\begin{cases} \frac{dz}{dx} = \cos(x) \cdot z \\ z(x)|_{x=0} = 2 \end{cases}$$

Численные методы для решения такой задачи хорошо исследованы, разработаны соответствующие алгоритмы и программы. Такие программы присутствуют в MATHCAD имеются встроенные процедуры: **rkfixed**, **Rkadapt**.

При этом предполагается, что пользователь должен подготовить свою задачу в виде системы ОДУ первого порядка в нормализованном виде Коши.

$$\begin{cases} \frac{dz_1}{dx} = f_1(x, z_1, z_2, \dots, z_N) \\ \frac{dz_2}{dx} = f_2(x, z_1, z_2, \dots, z_N) \\ \dots\dots\dots \\ \frac{dz_N}{dx} = f_N(x, z_1, z_2, \dots, z_N) \end{cases}$$

$$z_1(x_0) = z_{10}$$

$$\dots\dots\dots$$

$$z_n(x_0) = z_{n0}$$

Для двух рассмотренных примеров покажем, как можно свести их к нормализованной системе.

Для примера 2 из предыдущего параграфа:

$$E \cdot (J'' \cdot y'' + 2 \cdot J' \cdot y''' + J \cdot y'''') = q(x)$$

$$y(x_0) = y_0$$

$$y'(x_0) = y_0'$$

$$y''(x_0) = y_0''$$

$$y'''(x_0) = y_0'''$$

Выразим старшую производную из этого уравнения:

$$y^{(4)} = \frac{1}{J} \cdot \left(\frac{q(x)}{E} - J^{(2)} \cdot y^{(2)} - 2 \cdot J' \cdot y^{(3)} \right)$$

Введем новые функции:

$$\begin{aligned} z_1(x) &= y(x) \\ z_2(x) &= y'(x) \\ z_3(x) &= y''(x) \\ z_4(x) &= y'''(x) \end{aligned}$$

Тогда нормализованная система вида Коши имеет такой вид:

$$\begin{cases} \frac{dz_1}{dx} = z_2 \\ \frac{dz_2}{dx} = z_3 \\ \frac{dz_3}{dx} = z_4 \\ \frac{dz_4}{dx} = \frac{1}{J} \cdot \left(\frac{q(x)}{E} - J^{(2)} \cdot z_3 - 2 \cdot J' \cdot z_4 \right) \end{cases}$$

Если имеется система из нескольких дифференциальных уравнений, куда входят неизвестные функции, порядок старших производных которых равен k_1, k_2, \dots, k_l , то после преобразования этой системы к системе ОДУ первого порядка в нормализованном виде в этой системе окажется

$$k_1 + k_2 + \dots + k_l \quad \text{ОДУ.}$$

Для примера 3 из предыдущего параграфа:

$$\begin{aligned} m \cdot \ddot{x} + c \cdot \dot{x} + k \cdot x &= p(t) \\ x(t_0) &= x_0 \\ \dot{x}(t_0) &= \dot{x}_0 \end{aligned}$$

Выразим старшую производную:

$$\ddot{x} = \frac{p(t) - c\dot{x} - kx}{m} \quad (5.4)$$

Введём новые неизвестные функции:

$$\begin{aligned} z_1 &= x(t) \\ z_2 &= \dot{x}(t) \end{aligned}$$

Тогда уравнение можно представить в виде системы двух ОДУ:

$$\begin{cases} \frac{dz_1}{dt} = z_2 \\ \frac{dz_2}{dt} = \frac{p(t) - cz_2 - kz_1}{m} \end{cases}$$

с начальными условиями

$$\begin{aligned} z_1(t_0) &= x_0 \\ z_2(t_0) &= \dot{x}_0 \end{aligned}$$

Используем встроенную в MATHCAD процедуру **Rkadapt** для решения систем ОДУ методом Рунге-Кутты четвертого порядка для решения этой задачи. Документ MATHCAD представлен ниже.

```

w := 5
M := 1
C := 0
P(t) := sin(w·t)
K := 1
ORIGIN := 0

f(t, z) := 
$$\begin{bmatrix} z_1 \\ \frac{1 \cdot (P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix}$$

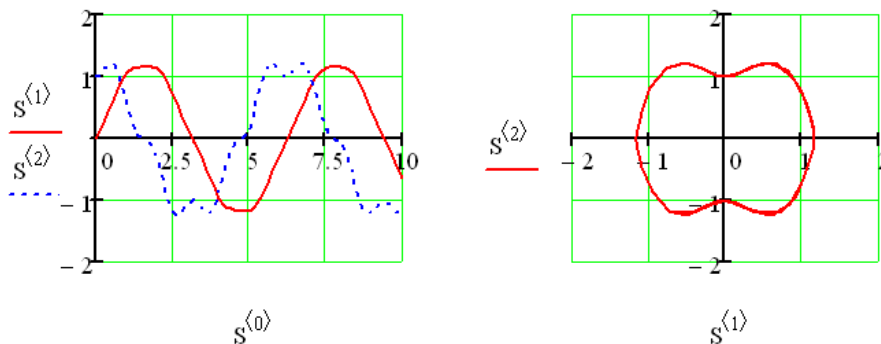
zbegin := 
$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$


tbegin := 0
tend := 10
h := 0.2

N := 
$$\frac{tend - tbegin}{h}$$


S := Rkadapt(zbegin, tbegin, tend, N, f)

```



При $p(t) = \sin(w \cdot t)$ и $w = w_{\text{собственное}}$

$$K := 1$$

$$M := 1$$

$$C := 0$$

$$w := \sqrt{\frac{1 \cdot K}{M}}$$

$$P(t) := \sin(w \cdot t)$$

$$\text{ORIGIN} := 0$$

$$f(t, z) := \begin{bmatrix} z_1 \\ 1 \cdot \frac{(P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix} \quad z_{\text{begin}} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

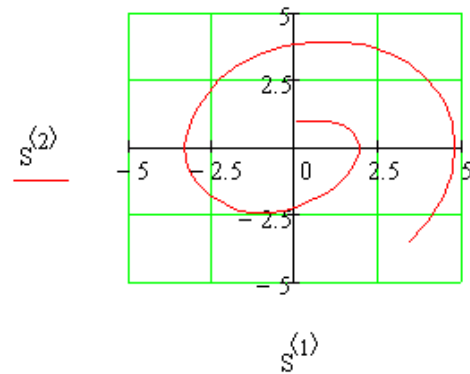
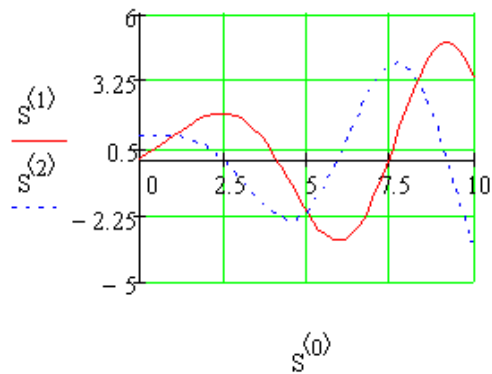
$$t_{\text{begin}} := 0$$

$$t_{\text{end}} := 10$$

$$h := 0.2$$

$$N := \frac{t_{\text{end}} - t_{\text{begin}}}{h}$$

$$S := \text{Rkadapt}(z_{\text{begin}}, t_{\text{begin}}, t_{\text{end}}, N, f)$$



При $p(t) = \text{const}$

$$K := 1$$

$$M := 1$$

$$C := 0$$

$$w := \sqrt{\frac{1-K}{M}}$$

$$P(t) := t$$

$$\text{ORIGIN} := 0$$

$$f(t, z) := \begin{bmatrix} z_1 \\ \frac{1 \cdot (P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix}$$

$$z_{\text{begin}} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

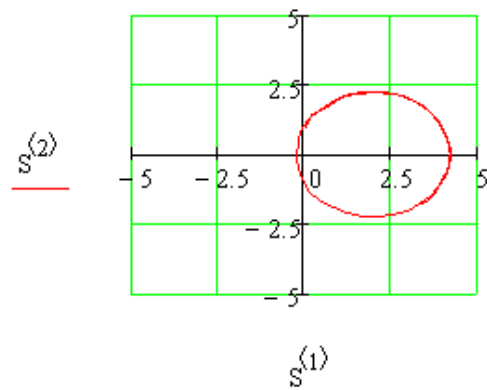
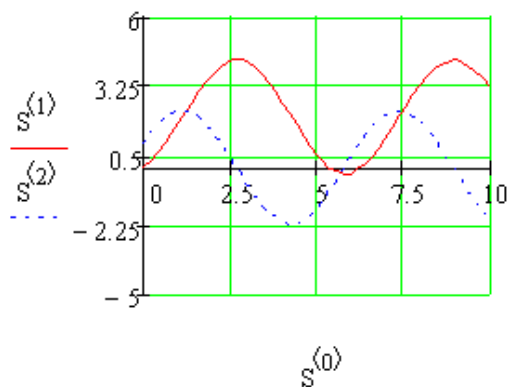
$$t_{\text{begin}} := 0$$

$$t_{\text{end}} := 10$$

$$h := 0.2$$

$$N := \frac{t_{\text{end}} - t_{\text{begin}}}{h}$$

$$S := \text{Rkadapt}(z_{\text{begin}}, t_{\text{begin}}, t_{\text{end}}, N, f)$$



При $p(t) = t$

$$K := 1$$

$$M := 1$$

$$C := 0$$

$$w := \sqrt{\frac{1 \cdot K}{M}}$$

$$P(t) := t$$

$$\text{ORIGIN} := 0$$

$$f(t, z) := \begin{bmatrix} z_1 \\ \frac{1 \cdot (P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix}$$

$$z_{\text{begin}} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

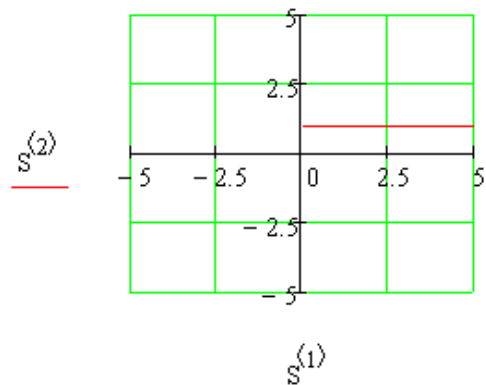
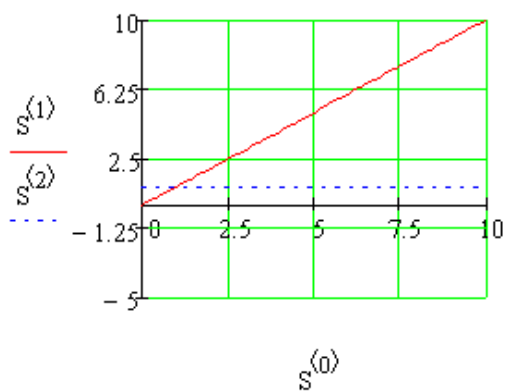
$$t_{\text{begin}} := 0$$

$$t_{\text{end}} := 10$$

$$h := 0.2$$

$$N := \frac{t_{\text{end}} - t_{\text{begin}}}{h}$$

$$S := \text{Rkadapt}(z_{\text{begin}}, t_{\text{begin}}, t_{\text{end}}, N, f)$$



При очень маленьком C:

$$K := 1$$

$$M := 1$$

$$C := 0.0001$$

$$w := \sqrt{\frac{1 \cdot K}{M}}$$

$$P(t) := 0$$

$$\text{ORIGIN} := 0$$

$$f(t, z) := \begin{bmatrix} z_1 \\ \frac{1 \cdot (P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix}$$

$$z_{\text{begin}} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

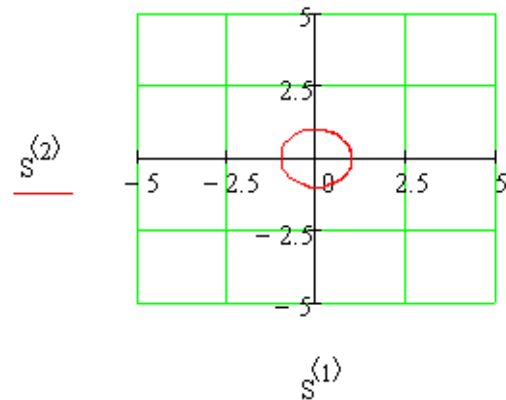
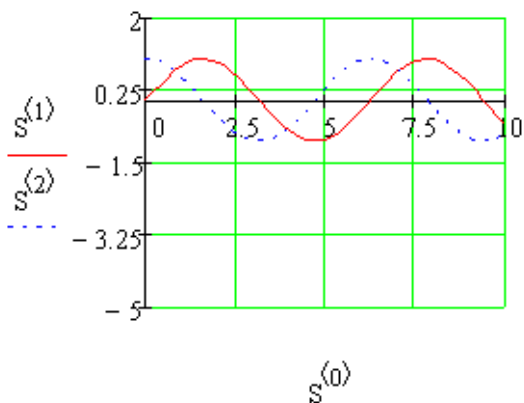
$$t_{\text{begin}} := 0$$

$$t_{\text{end}} := 10$$

$$h := 0.2$$

$$N := \frac{t_{\text{end}} - t_{\text{begin}}}{h}$$

$$S := \text{Rkadapt}(z_{\text{begin}}, t_{\text{begin}}, t_{\text{end}}, N, f)$$



При очень большом C:

$$K := 1$$

$$M := 1$$

$$C := 100000$$

$$w := \sqrt{\frac{1 \cdot K}{M}}$$

$$P(t) := 0$$

$$\text{ORIGIN} := 0$$

$$f(t, z) := \begin{bmatrix} z_1 \\ \frac{1 \cdot (P(t) - C \cdot z_1 - K \cdot z_0)}{M} \end{bmatrix}$$

$$z_{\text{begin}} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

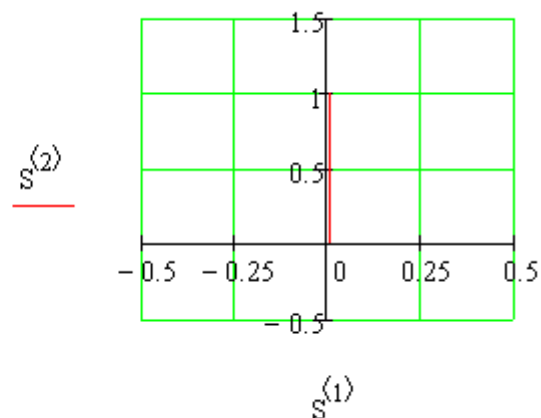
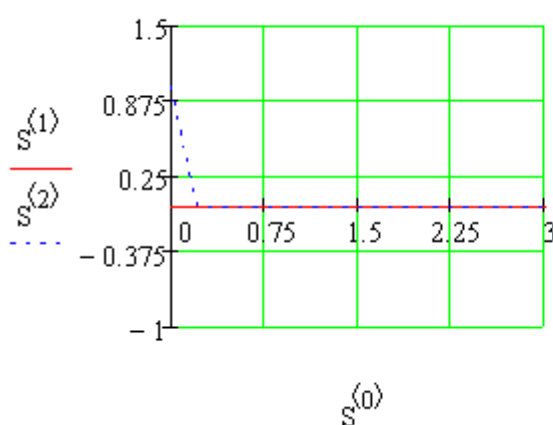
$$t_{\text{begin}} := 0$$

$$t_{\text{end}} := 10$$

$$h := 0.2$$

$$N := \frac{t_{\text{end}} - t_{\text{begin}}}{h}$$

$$S := \text{Rkadapt}(z_{\text{begin}}, t_{\text{begin}}, t_{\text{end}}, N, f)$$



Некоторые функции MATHCAD для решения обыкновенных дифференциальных уравнений

В Mathcad имеются встроенные функции, которые позволяют решать задачу Коши различными численными методами.

- $\text{rkfixed}(y_0, t_0, t_1, M, D)$ — метод Рунге-Кутты с фиксированным шагом,
- $\text{Rkadapt}(y_0, t_0, t_1, M, D)$ — метод Рунге-Кутты с переменным шагом;
- $\text{Buistoer}(y_0, t_0, t_1, M, D)$ — метод Булирша-Штера;
 - y_0 — вектор начальных значений в точке t_0 размера $N \times 1$;
 - t_0 — начальная точка расчета,
 - t_1 — конечная точка расчета,
 - M — число шагов, на которых численный метод находит решение;
 - D — векторная функция размера $N \times 1$ двух аргументов — скалярного t и векторного y . При этом y — искомая векторная функция аргумента t того же размера $N \times 1$.

Соблюдайте регистр первой буквы рассматриваемых функций, поскольку это влияет на выбор алгоритма счета, в отличие от многих других встроенных функций Mathcad.

Каждая из приведенных функций выдает решение в виде матрицы размера $(M+1) \times (N+1)$. В ее левом столбце находятся значения аргумента t , делящие интервал на равномерные шаги, а в остальных N столбцах — значения искомого функций $y_0(t)$, $y_1(t)$, .., $y_{N-1}(t)$, рассчитанные для этих значений аргумента. Поскольку всего точек (помимо начальной) M , то строк в матрице решения будет всего $M+1$

В подавляющем большинстве случаев достаточно использовать первую функцию rkfixed ,

$\text{rkfixed}(y, x_1, x_2, N, F)$ – выдает таблицу результатов решения системы обыкновенных дифференциальных уравнений методом Рунге-Кутты четвертого порядка точности с фиксированным шагом интегрирования $\frac{x_2 - x_1}{N}$

y – вектор начальных значений искомого решения;

x_1 – начальное значение независимой переменной;

x_2 – конечное значение независимой переменной;

N – фиксированное число шагов интегрирования на отрезке от x_1 до x_2 :

F – правые части системы уравнений, записанные в символьном виде.

Пример использования функции rkfixed приведен ниже:

$$F(x, Y) := \begin{bmatrix} 1 + Y_0 + x^2 \\ \sin(Y_1 + x^2) + Y_2 \\ 2 \cdot x \cdot (Y_1)^2 \end{bmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \begin{array}{l} x1 := 0 \\ x2 := 1 \\ N := 5 \end{array}$$

$$\text{rkfixed}(Y, x1, x2, N, F) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.2 & 0.224 & 1.181 & 0.05 \\ 0.4 & 0.515 & 1.404 & 0.254 \\ 0.6 & 0.906 & 1.689 & 0.736 \\ 0.8 & 1.437 & 2.062 & 1.725 \\ 1 & 2.155 & 2.583 & 3.664 \end{pmatrix}$$

Результатом решения функции `rkfixed` является матрица. Первый столбец этой матрицы – это независимая переменная x с $N=5$ равномерными интервалами разбиения заданного отрезка $x1-x2$. Второй и так далее столбцы – это решения для соответствующих функций $Y_i(x)$ с заданными начальными значениями.

Rkadapt(y,x1,x2,N,F) - выдает таблицу результатов решения системы обыкновенных дифференциальных уравнений методом Рунге-Кутты с адаптивным шагом интегрирования. Список параметров аналогичен функции `rkfixed`.

Пример использования функции `Rkadapt` приведен ниже.

Результирующая матрица имеет такой же вид, что и в функции `rkfixed`. Как видно, для одного и того же примера, приведенного для `rkfixed` и для `Rkadapt` решения немного расходятся. Следует ожидать, что решение по функции `Rkadapt` будет более точным.

$$F(x, Y) := \begin{bmatrix} 1 + Y_0 + x^2 \\ \sin(Y_1 + x^2) + Y_2 \\ 2 \cdot x \cdot (Y_1)^2 \end{bmatrix} \quad Y := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \begin{array}{l} x1 := 0 \\ x2 := 1 \end{array}$$

$$Rkadapt(Y, x1, x2, 5, F) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0.2 & 0.224 & 1.181 & 0.05 \\ 0.4 & 0.515 & 1.404 & 0.253 \\ 0.6 & 0.906 & 1.689 & 0.735 \\ 0.8 & 1.437 & 2.063 & 1.724 \\ 1 & 2.155 & 2.584 & 3.662 \end{pmatrix}$$

Во-первых, функция D, входящая в число параметров встроенных функций для решения ОДУ, должна быть функцией обязательно двух аргументов. Во-вторых, второй ее аргумент должен быть вектором того же размера, что и сама функция D. В-третьих, точно такой же размер должен быть и у вектора начальных значений.

Не забывайте, что векторную функцию D(t,y) следует определять через компоненты вектора y с помощью кнопки нижнего индекса (Subscript) с наборной панели Calculator (Калькулятор) или нажатием клавиши <[>. В третьей строке листинга определено число шагов, на которых рассчитывается решение, а его последняя строка присваивает матричной переменной и результат действия функции rkfixed.

При решении систем ОДУ многие проблемы могут быть устранены при помощи простой попытки увеличить число шагов численного метода. В частности, сделайте так при неожиданном возникновении ошибки "Found a number with a magnitude greater than 10^307" (Найдено число, превышающее значение 10307). Данная ошибка может означать не то, что решение в действительности расходится, а просто недостаток шагов для корректной работы численного алгоритма.

В заключение следует сказать несколько слов об особенностях различных численных методов. Все они основаны на аппроксимации дифференциальных уравнений разностными аналогами. В зависимости от конкретной формы аппроксимации, получаются алгоритмы различной точности и быстродействия. В Mathcad использован наиболее популярный алгоритм Рунге-Кутты четвертого

порядка, описанный в большинстве книг по методам вычислений. Он обеспечивает малую погрешность для широкого класса систем ОДУ за исключением жестких систем. Поэтому в большинстве случаев стоит применять функцию `rkfixed`. Если по различным причинам время расчетов становится критичным или точность неудовлетворительна, стоит попробовать вместо `rkfixed` другие функции, благо сделать это очень просто, благодаря одинаковому набору параметров. Для этого нужно только поменять имя функции в программе.

Функция `Rkadapt` может быть полезна в случае, когда известно, что решение на рассматриваемом интервале меняется слабо, либо существуют участки медленных и быстрых его изменений. Метод Рунге-Кутты с переменным шагом разбивает интервал не на равномерные шаги, а более оптимальным способом. Там, где решение меняется слабо, шаги выбираются более редкими, а в областях его сильных изменений — частыми. В результате, для достижения одинаковой точности требуется меньшее число шагов, чем для `rkfixed`. Метод Булирша-Штера `Vuistoer` часто оказывается более эффективным для поиска гладких решений.

Получение уравнений математической модели механической системы с сосредоточенными параметрами для в общем случае

Большое количество механических систем при выполнении предположений о том, что массы сосредоточены в отдельных точках, а податливыми являются лишь некоторые элементы (остальные являются бесконечно жёсткими) можно представить состоящими из разнотипных элементов, которые попарно не пересекаются. Такими элементами являются:

Упругие элементы (пружины)

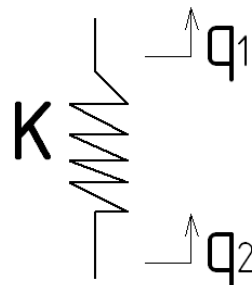


Рисунок 5.5. Упругий элемент

Диссипативный элемент (демпфер)

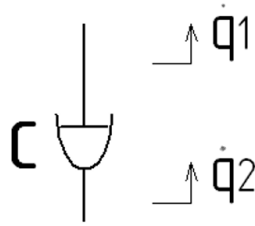


Рисунок 5.6. Демпфирующий элемент

Инерционный элемент (масса)

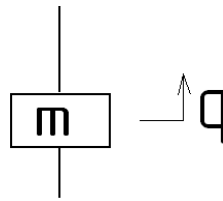


Рисунок 5.7. Инерционный элемент

Представленные схемы элементов соответствуют поступательному движению. Аналогичные же элементы могут быть введены и для случая вращательного движения:

- 1). крутильная пружина 2). крутильный демпфер 3). момент инерции

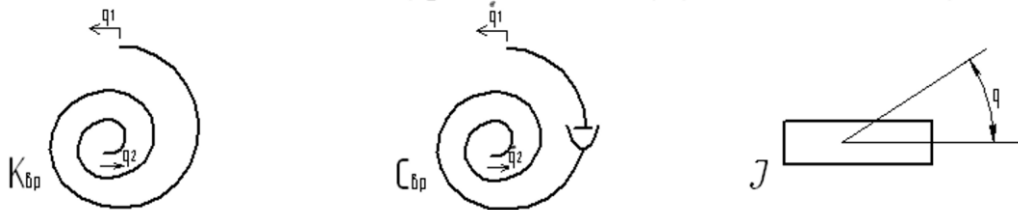


Рисунок 5.8. Аналогичные для случая вращательного движения

Элемент типа пружина накапливает потенциальную энергию. При этом будем считать, что при наличии перемещений концов q_1 , q_2 , когда удлинение составляет

$$\Delta = q_1 - q_2,$$

его действия на систему описывают силы упругости, пропорциональные удлинению:

$$F_y = k \cdot \Delta.$$

При этом необходимо ввести правило знаков: будем считать, что удлинению пружины соответствует положительная сила упругости.

На элементе типа демпфер механическая энергия преобразуется в тепловую. При этом считают, что сила демпфирования пропорциональна скорости удлинения

$$F_{\delta} = c \cdot \dot{\Delta} = c \cdot (\dot{q}_1 - \dot{q}_2).$$

Элемент типа масса «накапливает» кинетическую энергию. Сила инерции при этом имеет выражение

$$F_{инерции} = m \cdot \ddot{q}.$$

В зависимости от природы упругих элементов и демпферов соответствующие выражения сил упругости и затухания могут иметь и нелинейный вид.

Выражение кинетической энергии:

- При поступательном движении: $T = \frac{1}{2} \cdot m \cdot v^2$.
- При вращательном движении: $T = \frac{1}{2} \cdot J \cdot \dot{\omega}^2$.

На рисунках линейные и угловые перемещения обозначены буквой q .

Как известно, для получения уравнений движения такой системы могут быть использованы уравнения Лагранжа 2-го рода:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = Q_i \quad i = \overline{1, n}$$

$q_i = q_i(t)$ - перемещение по i -той обобщённой степени свободы;

T – кинетическая энергия системы;

$Q_i = Q_i^* - \frac{\partial \Phi}{\partial \dot{q}_i} - \frac{\partial \Pi}{\partial q_i}$ - обобщённая сила, действующая по i -тому направлению.

Φ – диссипативная функция системы;

Π – потенциальная энергия системы;

Q_i^* - внешняя сила, действующая по i -тому обобщённому направлению.

Таким образом, последовательность действий для записей уравнений движения механической системы с сосредоточенными параметрами следующая:

- необходимо мысленно разбить систему на попарно непересекающиеся элементы стандартного типа (формируется расчетная схема системы);
- вводятся обобщенные перемещения q_1, q_2, \dots, q_n . Эти обобщенные перемещения должны полностью описывать положение системы в произвольный момент времени и никакое из них не должно выражаться через остальные (независимые);
- при заданных величинах свободных перемещений записываются выражения для абсолютных перемещений инерционных элементов;
- записывают выражения для удлинений упругих элементов;

- записывают выражения для скоростей удлинений диссипативных элементов;
- составляют выражение для кинетической энергии T , диссипативной функции Φ и потенциальной энергии Π всей системы в целом;
- вычисляют частные производные, присутствующие в уравнении Лагранжа:

$$\frac{\partial T}{\partial \dot{q}_i}; \frac{\partial T}{\partial q_i}; \frac{\partial \Phi}{\partial \dot{q}_i}; \frac{\partial \Pi}{\partial q_i};$$

- получают выражение для величины Q_i^* . Для этого
 - изображают все активные силы, действующие на систему при заданных положительных направлениях обобщенных перемещений; вычислить значения работы этих сил на заданных перемещениях;
 - полагают:

$$\delta q_1 \neq 0, \delta q_2 = \delta q_3 = \dots = \delta q_n = 0,$$

то есть задают возможные перемещения только по первому направлению, а по всем остальным запрещают. На заданном возможном перемещении вычисляют работу всех активных сил и реакций связей. Если в полученном выражении привести подобные и представить его в виде $\delta q_1 \cdot (\dots)$, то коэффициент при этом δq_1 будет Q_1^* .

- теперь полагают:

$\delta q_2 \neq 0, \delta q_1 = \delta q_3 = \dots = \delta q_n = 0$. Выполняем то же самое и получаем выражение в следующем виде $\delta q_2 \cdot (\dots)$, коэффициент при δq_2 будет Q_2^* .

И так далее для всех степеней свободы.

Пример 1.

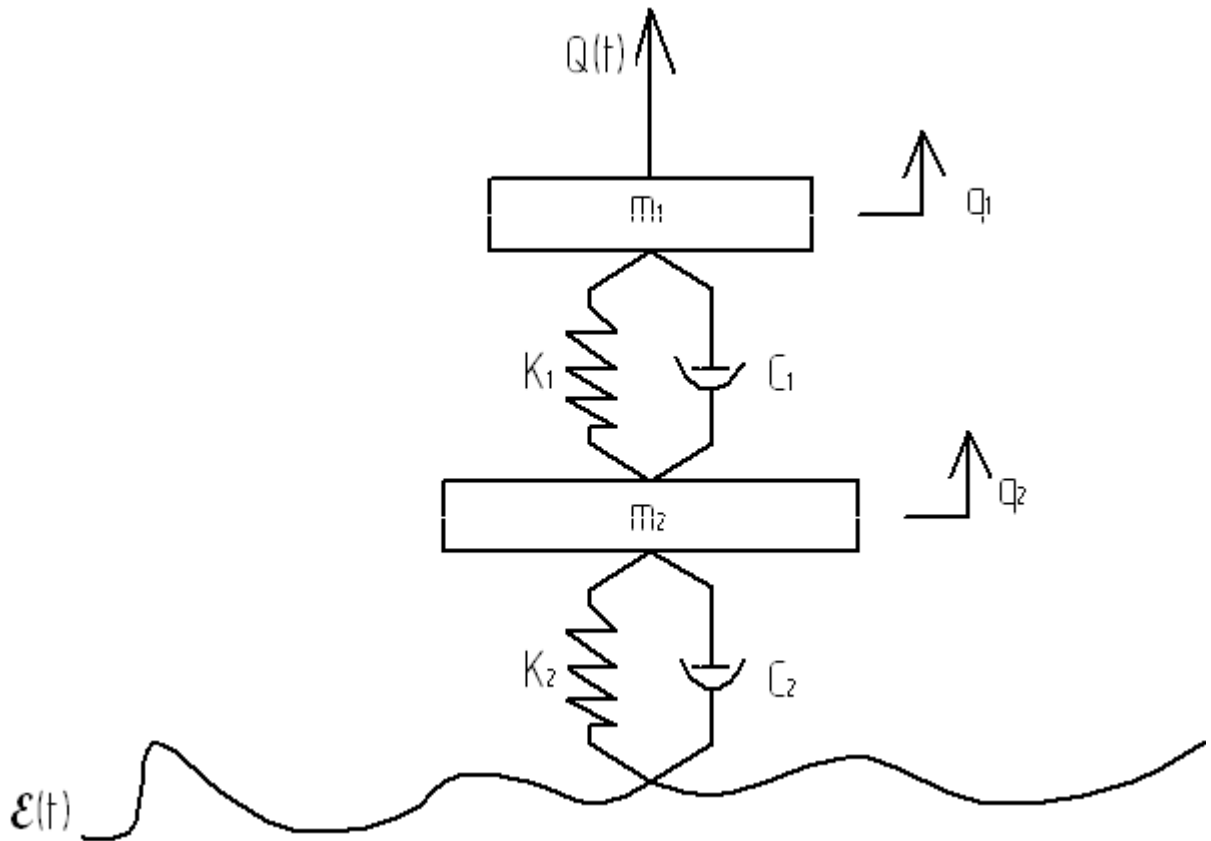


Рисунок 5.9. Колебательная двухмассовая система

Получим уравнения движения для двухмассовой системы. Движение происходит по дороге со скоростью v .

$\xi(t)$ – закон дорожного возмущения (кинематическое возмущение).

m_1, m_2 – массы;

k_1, k_2 – жесткости упругих элементов;

c_1, c_2 – характеристики демпфирующих элементов.

Введены обобщенные перемещения

$q_1 = q_1(t)$ – смещение верхней массы относительно дороги.

$q_2 = q_2(t)$ – смещение нижней массы относительно дороги.

При этом будем считать, что эти смещения соответствуют недеформированному состоянию пружины. Очевидно, что введенных обобщенных степеней достаточно для однозначного определения положения системы в любой момент времени. При этом ни одно из них не выражается через другое.

Составим выражения для абсолютных перемещений:

$$u_1(t) = q_1(t)$$

$$u_2(t) = q_2(t)$$

Запишем уравнения для удлинений пружин:

$$\Delta_2 = q_2 - \xi(t)$$

$$\Delta_1 = q_1 - q_2$$

Запишем скорости удлинений:

$$\dot{\Delta}_2 = \dot{q}_2 - \dot{\xi}(t)$$

$$\dot{\Delta}_1 = \dot{q}_1 - \dot{q}_2$$

Записываем выражение для кинетической энергии:

$$T = \frac{1}{2} \cdot m_1 \cdot \dot{u}_1^2 + \frac{1}{2} \cdot m_2 \cdot \dot{u}_2^2 = \frac{1}{2} \cdot m_1 \cdot \dot{q}_1^2 + \frac{1}{2} \cdot m_2 \cdot \dot{q}_2^2$$

Записываем выражение для диссипативной функции:

$$\Phi = \frac{1}{2} \cdot c_1 \cdot \dot{\Delta}_1^2 + \frac{1}{2} \cdot c_2 \cdot \dot{\Delta}_2^2 = \frac{1}{2} \cdot c_1 \cdot (\dot{q}_1 - \dot{q}_2)^2 + \frac{1}{2} \cdot c_2 \cdot (\dot{q}_2 - \dot{\xi})^2$$

Выражение для потенциальной энергии:

$$\Pi = \frac{1}{2} \cdot k_1 \cdot \Delta_1^2 + \frac{1}{2} \cdot k_2 \cdot \Delta_2^2 = \frac{1}{2} \cdot k_1 \cdot (q_1 - q_2)^2 + \frac{1}{2} \cdot k_2 \cdot (q_2 - \xi)^2$$

Выражаем Q_1^* :

$$A_1 = Q(t) \cdot \delta q_1 \text{ поэтому } Q_1^* = Q(t)$$

$$Q_2^* = 0$$

Вычислим нужные частные производные:

$$\frac{\partial T}{\partial \dot{q}_1} = m_1 \cdot \dot{q}_1 \qquad \frac{\partial T}{\partial \dot{q}_2} = m_2 \cdot \dot{q}_2$$

$$\frac{\partial T}{\partial q_1} = 0 \qquad \frac{\partial T}{\partial q_2} = 0$$

$$\frac{\partial \Pi}{\partial q_1} = k_1 \cdot (q_1 - q_2)$$

$$\frac{\partial \Pi}{\partial q_2} = -k_1 \cdot (q_1 - q_2) + k_2 \cdot (q_2 - \xi)$$

$$\frac{\partial \Phi}{\partial \dot{q}_1} = c_1 \cdot (\dot{q}_1 - \dot{q}_2)$$

$$\frac{\partial \Phi}{\partial \dot{q}_2} = -c_1 \cdot (\dot{q}_1 - \dot{q}_2) + c_2 \cdot (\dot{q}_2 - \dot{\xi})$$

Окончательно система уравнений имеет вид:

$$\begin{cases} m_1 \cdot \ddot{q}_1 = Q(t) - c_1 \cdot (\dot{q}_1 - \dot{q}_2) - k_1 \cdot (q_1 - q_2) \\ m_2 \cdot \ddot{q}_2 = c_1 \cdot (\dot{q}_1 - \dot{q}_2) - c_2 \cdot (\dot{q}_2 - \dot{\xi}) + k_1 \cdot (q_1 - q_2) - k_2 \cdot (q_2 - \xi) \end{cases}$$

Пример 5.

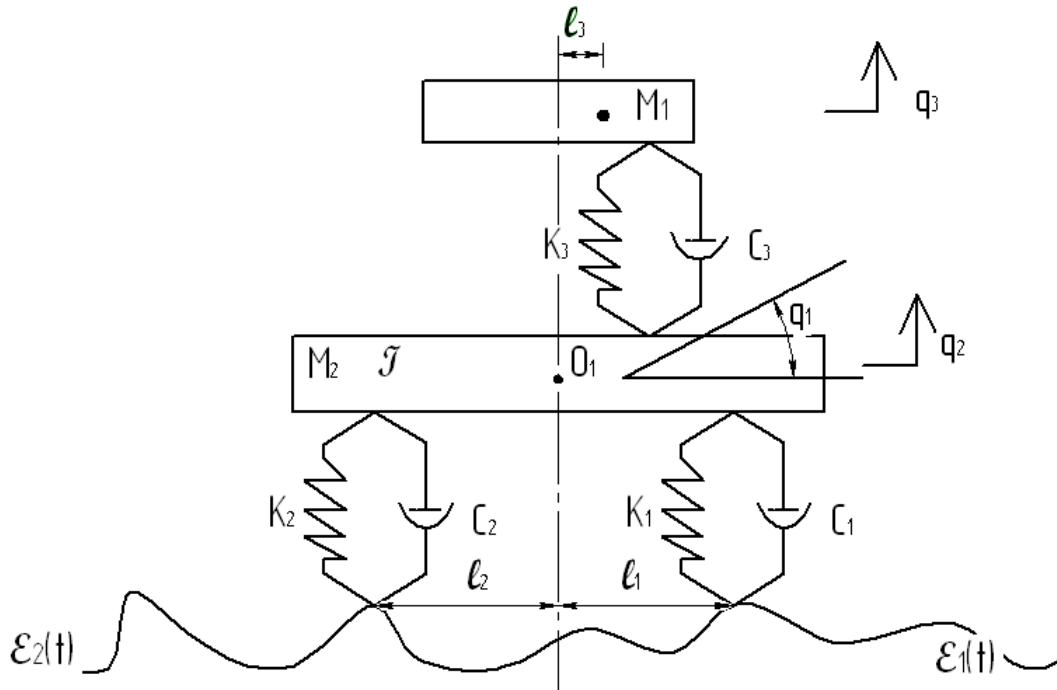


Рисунок 5.10. Система с тремя степенями свободы

$q_1 = q_1(t)$ - перемещение нижней платформы относительно грунта;

$q_2 = q_2(t)$ - угол поворота платформы вокруг точки O_1 относительно грунта;

$q_3 = q_3(t)$ - перемещение массы M_1 относительно точки крепления к платформе.

Абсолютные перемещения:

$$u_1 = q_1$$

$$u_2 = q_2$$

$$u_3 = q_1 + q_2 \cdot l_3 + q_3$$

$$\dot{u}_1 = \dot{q}_1$$

$$\dot{u}_2 = \dot{q}_2$$

$$\dot{u}_3 = \dot{q}_1 + \dot{q}_2 \cdot l_3 + \dot{q}_3$$

Удлинения пружин и скорости удлинений:

$$k_3: \Delta_3 = q_3$$

$$\dot{\Delta}_3 = \dot{q}_3$$

$$k_1: \Delta_1 = q_1 + q_2 \cdot l_1 - \xi_1(t)$$

$$\dot{\Delta}_1 = \dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)$$

$$k_2: \Delta_2 = q_1 - q_2 \cdot l_2 - \xi_2(t)$$

$$\dot{\Delta}_2 = \dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t)$$

Кинетическая энергия:

$$T = \frac{1}{2} \cdot M_2 \cdot \dot{q}_1^2 + \frac{1}{2} \cdot J \cdot \dot{q}_2^2 + \frac{1}{2} \cdot M_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_3 + \dot{q}_3)^2$$

Потенциальная энергия:

$$\Pi = \frac{1}{2} \cdot (k_1 \cdot \Delta_1^2 + k_2 \cdot \Delta_2^2 + k_3 \cdot \Delta_3^2)$$

Для диссипативной функции:

$$\Phi = \frac{1}{2} \cdot (c_1 \cdot \dot{\Delta}_1^2 + c_2 \cdot \dot{\Delta}_2^2 + c_3 \cdot \dot{\Delta}_3^2)$$

Уравнение Лагранжа:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = Q_i \quad i = \overline{1, n}$$

$$i = \overline{1, n}, \quad n=3.$$

Вычислим нужные частные производные:

$$\frac{\partial T}{\partial \dot{q}_1} = M_2 \cdot \dot{q}_1 + M_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_3 + \dot{q}_3)$$

$$\frac{\partial T}{\partial q_1} = 0$$

$$\frac{\partial T}{\partial \dot{q}_2} = J \cdot \dot{q}_2 + M_1 \cdot l_3 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_3 + \dot{q}_3)$$

$$\frac{\partial T}{\partial q_2} = 0$$

$$\frac{\partial T}{\partial \dot{q}_3} = M_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_3 + \dot{q}_3)$$

$$\frac{\partial T}{\partial q_3} = 0$$

$$\frac{\partial \Pi}{\partial q_1} = k_1 \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) + k_2 \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t))$$

$$\frac{\partial \Pi}{\partial q_2} = k_1 \cdot l_1 \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) - k_2 \cdot l_2 \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t))$$

$$\frac{\partial \Pi}{\partial q_3} = k_3 \cdot q_3$$

$$\frac{\partial \Phi}{\partial \dot{q}_1} = c_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) + c_2 \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t))$$

$$\frac{\partial \Phi}{\partial \dot{q}_2} = c_1 \cdot l_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) - l_2 \cdot c_2 \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t))$$

$$\frac{\partial \Phi}{\partial \dot{q}_3} = c_3 \cdot \dot{q}_3$$

Окончательно запишем систему трёх ДУ второго порядка:

$$\begin{cases} M_2 \cdot \ddot{q}_1 + M_1 \cdot (\ddot{q}_1 + \dot{q}_2 \cdot l_3 + \ddot{q}_3) = -k_1 \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) - k_2 \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t)) - \\ -c_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) - c_2 \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t)) \\ J \cdot \ddot{q}_2 + M_1 \cdot l_3 \cdot (\ddot{q}_1 + \dot{q}_2 \cdot l_3 + \ddot{q}_3) = -k_1 \cdot l_1 \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) + k_2 \cdot l_2 \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t)) - \\ -c_1 \cdot l_1 \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) + l_2 \cdot c_2 \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t)) \\ M_1 \cdot (\ddot{q}_1 + \dot{q}_2 \cdot l_3 + \ddot{q}_3) = -k_3 \cdot q_3 - c_3 \cdot \dot{q}_3 \end{cases} \quad (5.5)$$

Эта система должна быть снабжена шестью начальными условиями:

$$q_1(0) = q_{10}$$

$$\dot{q}_1(0) = \dot{q}_{10}$$

$$q_2(0) = q_{20}$$

$$\dot{q}_2(0) = \dot{q}_{20}$$

$$q_3(0) = q_{30}$$

$$\dot{q}_3(0) = \dot{q}_{30}$$

Приведём пример возможного упрощения уравнения.

Рассмотрим ситуацию, когда верхняя масс значительно меньше нижней:

$$\frac{M_1}{M_2} \ll 1 \quad (5.6)$$

$$\frac{M_1 \cdot l_3}{J} \ll 1$$

Тогда после деления первого уравнения на M_2 , а второго уравнения на J , в левой части образуются слагаемые с коэффициентом вида (5.6), которыми можно пренебречь по сравнению со слагаемыми, содержащими \ddot{q}_1 в первом уравнении и \ddot{q}_2 - во втором уравнении. Тогда уравнения, аппроксимирующее исходную систему будет иметь вид:

$$\begin{cases} \ddot{q}_1 = -\frac{k_1}{M_2} \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) - \frac{k_2}{M_2} \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t)) - \\ -\frac{c_1}{M_2} \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) - \frac{c_2}{M_2} \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t)) \\ \ddot{q}_2 = -\frac{k_1 \cdot l_1}{J} \cdot (q_1 + q_2 \cdot l_1 - \xi_1(t)) + \frac{k_2 \cdot l_2}{J} \cdot (q_1 - q_2 \cdot l_2 - \xi_2(t)) - \\ -\frac{c_1 \cdot l_1}{J} \cdot (\dot{q}_1 + \dot{q}_2 \cdot l_1 - \dot{\xi}_1(t)) + \frac{l_2 \cdot c_2}{J} \cdot (\dot{q}_1 - \dot{q}_2 \cdot l_2 - \dot{\xi}_2(t)) \end{cases} \quad (5.7)$$

Как видно из системы уравнений (5.7) q_1 и q_2 не зависят от q_3 , т.е. движение платформы не зависит от движения груза на ней, поэтому для получения закона движения груза q_3 , можно предварительно решить систему (5.7), а затем подставить эти известные функции в последнее уравнение системы (5.5) и решить его относительно q_3 .

Сведём систему (5.5) к системе ОДУ 1-го порядка в каноническом виде Коши. Очевидно, (5.5) можно записать в следующем виде:

$$\begin{cases} a_{11} \cdot \ddot{q}_1 + a_{12} \cdot \ddot{q}_2 + a_{13} \cdot \ddot{q}_3 = b_1 \\ a_{21} \cdot \ddot{q}_1 + a_{22} \cdot \ddot{q}_2 + a_{23} \cdot \ddot{q}_3 = b_2 \\ a_{31} \cdot \ddot{q}_1 + a_{32} \cdot \ddot{q}_2 + a_{33} \cdot \ddot{q}_3 = b_3 \end{cases} \quad (5.8)$$

$$a_{11} = M_1 + M_2$$

$$a_{12} = M_1 \cdot l_3$$

$$a_{13} = M_1$$

Здесь b_1, b_2, b_3 - правые части системы (4), например

$$b_1 = -k_1 \cdot (q_1 + l_1 \cdot q_2 - \xi_1(t)) - \dots$$

Как видно, если формально считать \ddot{q}_1, \ddot{q}_2 и \ddot{q}_3 неизвестными, то (4) представляет систему линейных алгебраических уравнений. Её можно решить, например, по формуле Крамера:

$$\ddot{q}_1 = \frac{\Delta_1}{\Delta} \quad \ddot{q}_2 = \frac{\Delta_2}{\Delta} \quad \ddot{q}_3 = \frac{\Delta_3}{\Delta}$$

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad \Delta_1 = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix} \quad \text{и так далее.}$$

Введем новые функции

$$z_1 = q_1$$

$$z_2 = \dot{q}_1$$

$$z_3 = q_2$$

$$z_4 = \dot{q}_2$$

$$z_5 = q_3$$

$$z_6 = \dot{q}_3.$$

Тогда эквивалент системы (1) примет вид:

$$\begin{cases} \frac{dz_1}{dt} = z_2 \\ \frac{dz_2}{dt} = \frac{\Delta_1}{\Delta} \\ \frac{dz_3}{dt} = z_4 \\ \frac{dz_4}{dt} = \frac{\Delta_2}{\Delta} \\ \frac{dz_5}{dt} = z_6 \\ \frac{dz_6}{dt} = \frac{\Delta_3}{\Delta} \end{cases}$$

Начальные условия теперь запишутся в виде:

$$z_1(0) = q_{10}$$

$$z_2(0) = \dot{q}_{10}$$

$$z_3(0) = q_{20}$$

$$z_4(0) = \dot{q}_{20}$$

$$z_5(0) = q_{30}$$

$$z_6(0) = \dot{q}_{30}$$

Шаговые методы решения задачи Коши

Пусть интервал наблюдений (x_0, x_n) разбит на подинтервалы с равным шагом (рисунок 5.11).

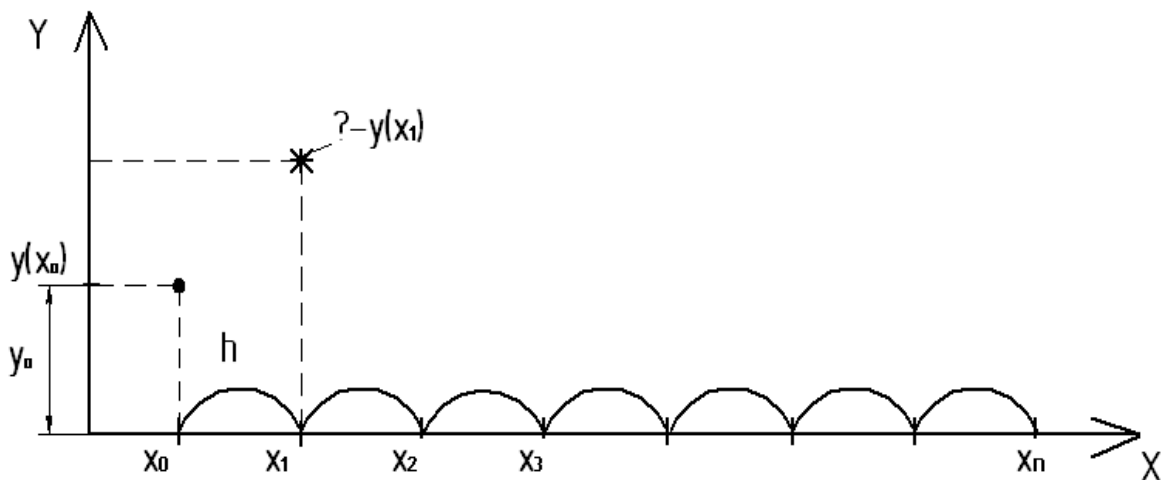


Рисунок 5.11. Интервал наблюдений, разбитый на подинтервалы

Предположим, что нам известно решение ДУ

$$\frac{dy}{dx} = f(x, y) \quad (5.9)$$

с начальным условием

$$y(x_0) = y_0 \quad (5.10)$$

Если каким-то способом удастся построить решение в точке x , на конце шага h , то используя этот же способ для последующих шагов, мы найдем решение и в остальных узлах. Если решается не одно ДУ, а их система

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, \dots, y_k) \\ \dots\dots\dots \\ \frac{dy_k}{dx} = f_k(x, y_1, \dots, y_k) \end{cases} \quad (5.11)$$

То можно использовать такой же способ для решения всех уравнений на каждом шаге. Таким образом решение системы (5.11) с начальными условиями:

$$y_1(x_0) = y_{10}$$

$$\dots\dots\dots$$

$$y_k(x_0) = y_{k0} \quad (5.12)$$

сводится к проблеме решения одного ДУ (5.9) с начальными условиями (5.10) на отдельном шаге h .

Выполним интегрирование обеих частей ДУ (5.9) на интервале $[x_0, x_0 + h]$.

$$\int_{x_0}^{x_0+h} \frac{dy}{dx} dx = \int_{x_0}^{x_0+h} f(x, y) dx$$

$$y(x)|_{x_0}^{x_0+h} = \int_{x_0}^{x_0+h} f(x, y) dx$$

$$y(x_0 + h) - y(x_0) = \int_{x_0}^{x_0+h} f(x, y) dx$$

$$y(x_0 + h) = y(x_0) + \int_{x_0}^{x_0+h} f(x, y) dx$$

$$y_1 = y_0 + \Delta y$$

где $\Delta y = \int_{x_0}^{x_0+h} f(x, y) dx$.

(5) – точное равенство. Нижний индекс здесь означает номер текущей точки на интервале наблюдения.

Приближенные вычисления начинаются, только тогда, когда мы заменяем значение интеграла его приближенным значением.

Разные методы интегрирования ОДУ различаются формулами для вычисления интеграла.

Метод Эйлера решения задачи Коши

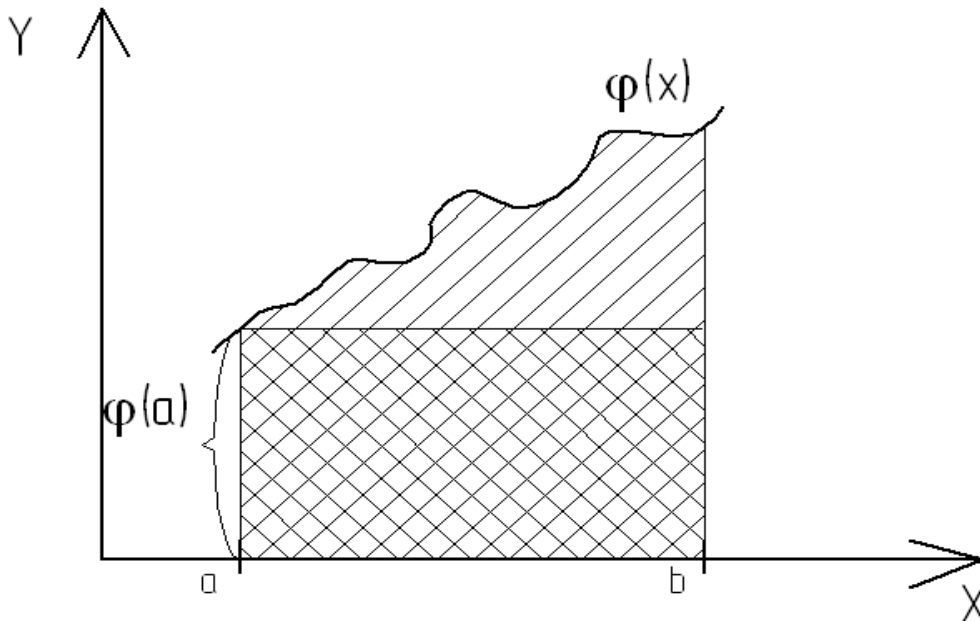


Рисунок 5.15. Приближенное нахождение интеграла по формуле левых прямоугольников.

$$\int_a^b \phi(x) dx \approx \phi(a) \cdot (b - a)$$

$$\int_{x_0}^{x_0+h} f(x) dx = f(x_0) \cdot h$$

$$y_1 \approx y_0 + f(x_0) \cdot h$$

То есть мы приближенно заменили $\Delta y \approx \tilde{\Delta} y$. Здесь $\tilde{\Delta} y = f(x_0) \cdot h$
Тогда формула метода Эйлера для произвольного шага примет вид:

$$y_{i+1} \approx y_i + f(x_i) \cdot h$$

Заметим, что можно следующим образом интерпретировать формулу Эйлера.

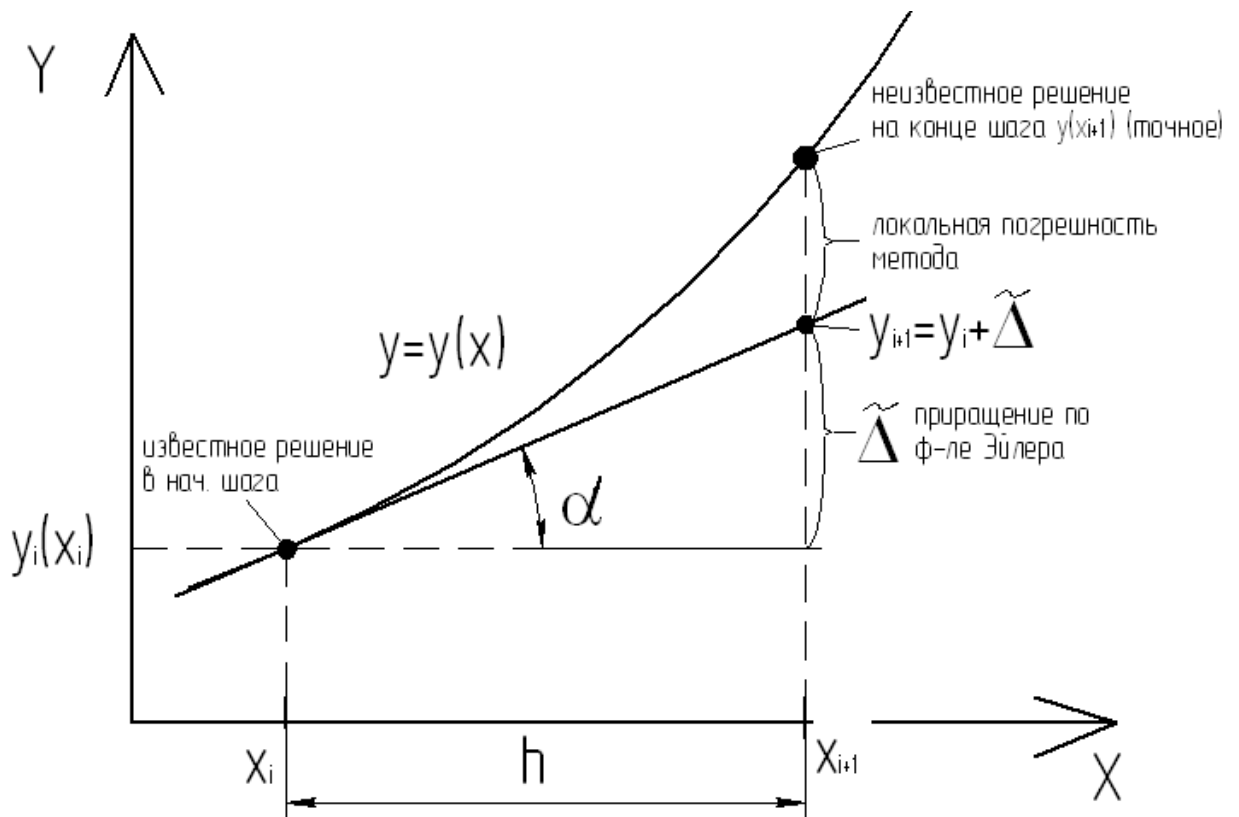


Рисунок 5.13. Графическая интерпретация метода Эйлера

$$tg(\alpha) = \left. \frac{dy}{dx} \right|_{x=x_i}$$

Пусть у нас имеется решаемое ДУ:

$$\frac{dy}{dx} = f(x, y) \quad (5.12)$$

Из ДУ (5.12)

$$tg(\alpha) = f(x, y)|_{x=x_i} = f_i$$

$$\tilde{\Delta}y = h \cdot tg(\alpha)$$

Локальная погрешность метода: $(y(x_{i+1}) - y_{i+1})$, где $y(x_{i+1})$ - точно неизвестное, а y_{i+1} - вычислено по приближенной формуле.

Локальная погрешность – отклонение приближенного значения от точного на данном шаге интегрирования.

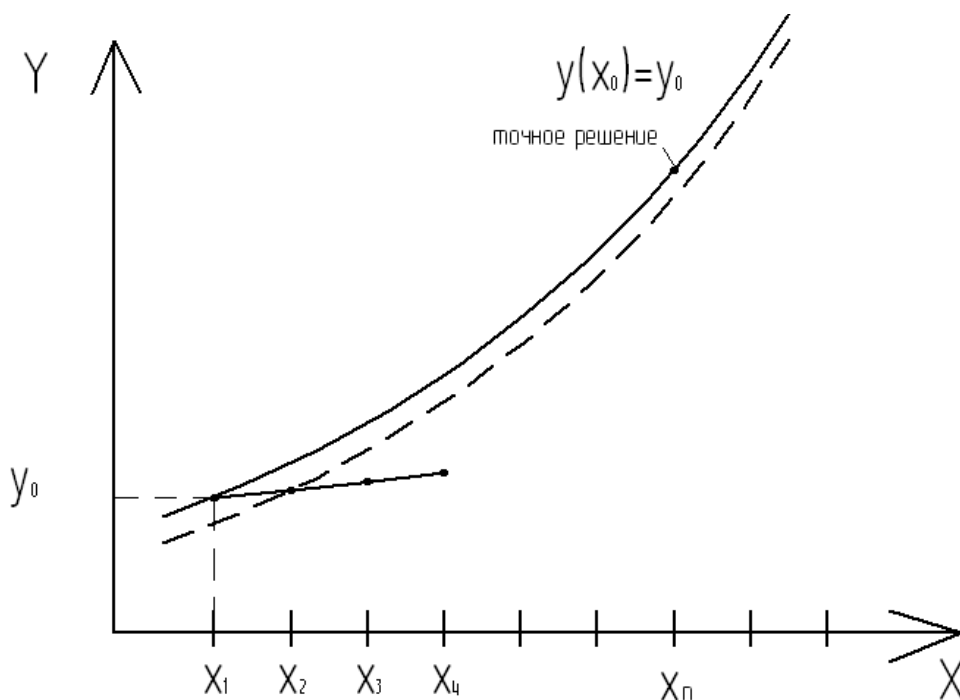


Рисунок 5.14. Иллюстрация «перескакивания» из одной интегрируемой прямой на другую на каждом шаге.

Как видно, для некоторых видов ДУ может возникнуть ситуация, когда ошибка будет бесконечно нарастать при продвижении по независимой переменной. Такой процесс носит название неустойчивости численного решения. Например, если решением ДУ является $y = c \cdot e^x$, то погрешность, допущенная на каком-либо шаге, будет возрастать в дальнейшем по экспоненте. Для уравнения $\frac{dy}{dx} = -y$, решением которого является $y = c \cdot e^{-x}$, погрешность будет стремиться к нулю.

Таким образом следует различать неустойчивость решения, связанную с природой самого ДУ, и неустойчивость, связанную с численным методом решения. То есть может оказаться так, что несмотря на то, что само ДУ «устойчиво», численный метод приносит такую ошибку, что

приближенное решение отклоняется от точного при $x \rightarrow \infty$ на бесконечно большую величину.

$$y_{i+1} \approx y_i + f(x_i) \cdot h$$

$$y_{i+1} = y_i + f(x_i) \cdot h + O(h^{k+1}),$$

где $O(h^{k+1})$ - локальная погрешность метода,

Число k называют порядком метода интегрирования.

Если $h=0,01$, $k=1$, то

$$O(h^{k+1}) = 0,01^2$$

Чем больше k , тем меньше локальная погрешность, тем точнее метод.

Метод Эйлера имеет локальную погрешность $O(h^2)$; $k=1$ – метод 1-го порядка точности.

Усовершенствованный метод Эйлера решения задачи Коши

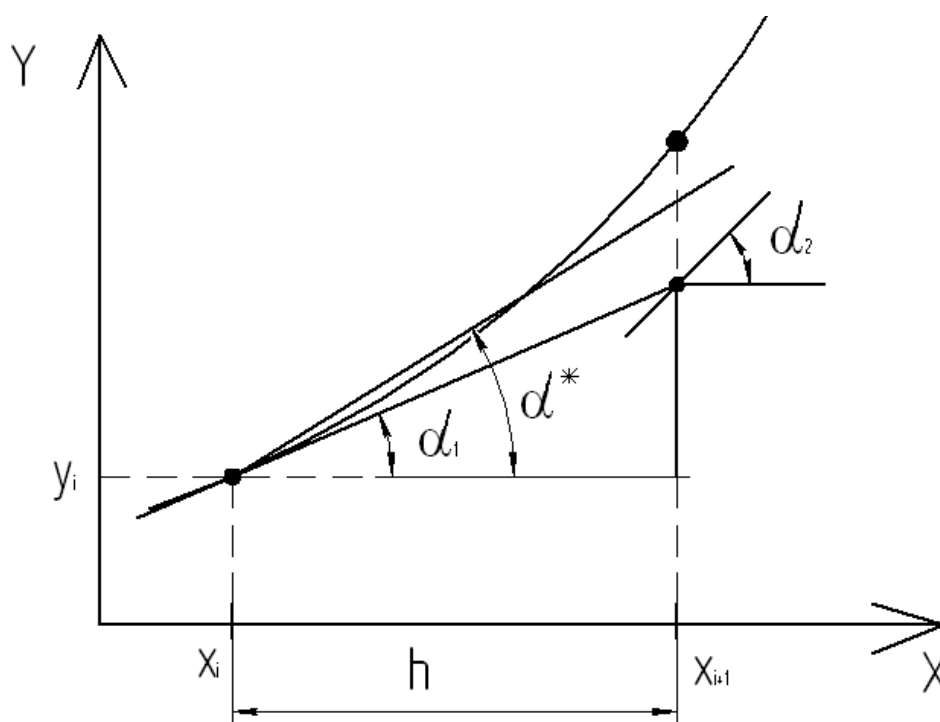


Рисунок 5.15. Усовершенствованный метод Эйлера. Графическая интерпретация

Основная идея этого метода заключается в том, что результирующий шаг будет выполняться под уточненным углом наклона α^* , удовлетворяющим соотношению:

$$tg(\alpha^*) = \frac{1}{2} \cdot (tg(\alpha_1) + tg(\alpha_2)),$$

где α_1 - угол, под которым наклонено решение в начальной точке интервала,

α_2 - приближенный угол, под которым наклонено решение в конечной точке интервала.

Последовательность шагов при реализации усовершенствованного метода Эйлера:

1. По формуле Эйлера вычислить

$$y_{i+1} \approx y_i + f(x_i) \cdot h$$
2. Используя правую часть ДУ, вычислить

$$\tilde{f}_{i+1} = f(x_{i+1}, y_{i+1})$$

3. Вычислить

$$\operatorname{tg}(\alpha^*) = \frac{1}{2} \cdot (\operatorname{tg}(\alpha_1) + \operatorname{tg}(\alpha_2))$$

$$f^* = \frac{1}{2} \cdot (f_i + \tilde{f}_{i+1})$$

4. Вернувшись в начало шага, выполнить окончательный расчет для решения на конце шага

$$y_{i+1} \approx y_i + f_i^* \cdot h.$$

Для усовершенствованного метода Эйлера:

$$y_{i+1} = y_i + f_i^* \cdot h + O(h^3),$$

$k=2$ – это метод 2-го порядка точности. Эта точность достигается двукратным вычислением правой части ДУ.

Модифицированный метод Эйлера решения задачи Коши

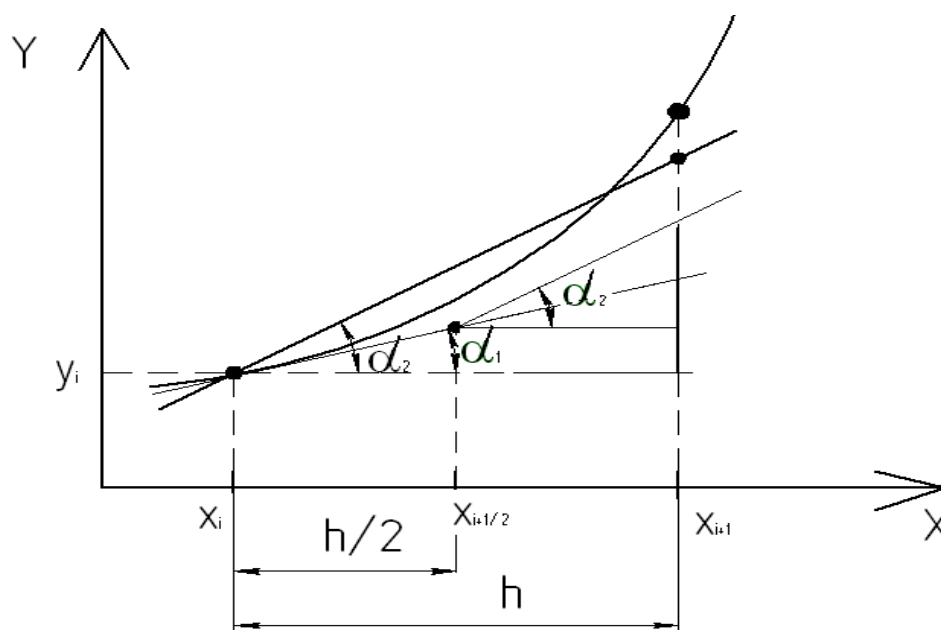


Рисунок 5.16. Модифицированный метод Эйлера. Графическая интерпретация

Последовательность шагов при реализации модифицированного метода Эйлера:

1. В этом методе сначала выполняется поиск решения по методу Эйлера на половине шага:

$$y_{i+\frac{1}{2}} \approx y_i + f_i \cdot \frac{h}{2}$$

2. Затем вычисляется приближенное значение производной в середине интервала

$$f_{i+\frac{1}{2}}^* \approx f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$$

3. И только после этого, вернувшись в начало шага, выполняется окончательный расчет для решения на конце шага

$$y_{i+1} \approx y_i + f_{i+\frac{1}{2}}^* \cdot h.$$

Этот метод имеет локальную погрешность $O(h^3)$, $k=2$, это метод 2-го порядка точности.

Формулы для приближенного решения задачи Коши методами Рунге-Кутты

Как отмечено выше, все приближенные формулы для решения ОДУ имеют вид

$$y_{i+1} \approx y_i + \tilde{\Delta}y$$

и отличаются лишь вычислением $\tilde{\Delta}y$.

Заметим, что для всех формул Рунге-Кутты необходимо вычислять следующие значения:

$$\phi_0 = h \cdot f(x_i, y_i)$$

$$\phi_1 = h \cdot f(x_i + \alpha_1 \cdot h, y_i + \beta_{10} \cdot \phi_0)$$

$$\phi_2 = h \cdot f(x_i + \alpha_2 \cdot h, y_i + \beta_{20} \cdot \phi_0 + \beta_{21} \cdot \phi_1)$$

.....

$$\phi_q = h \cdot f(x_i + \alpha_q \cdot h, y_i + \beta_{q0} \cdot \phi_0 + \beta_{q1} \cdot \phi_1 + \dots + \beta_{q(q-1)} \cdot \phi_{q-1})$$

Из этих величин составить линейную комбинацию, аппроксимирующую «добавку» $\tilde{\Delta}$

$$\tilde{\Delta}y = A_0 \cdot \phi_0 + A_1 \cdot \phi_1 + \dots + A_q \cdot \phi_q$$

Приведем формулы Рунге-Кутта 2-го порядка точности.

Для метода 2-го порядка точности константы A, α, β должны удовлетворять следующим условиям:

$$\begin{cases} A_0 + A_1 = 1 \\ A_1 \cdot \alpha_1 = \frac{1}{2} \\ A_1 \cdot \beta_{10} = \frac{1}{2} \end{cases}$$

Это системы трех уравнений относительно четырех неизвестных. Она имеет бесконечное множество решений. Если

$$\text{Для } A_1 = \frac{1}{2}, \text{ то } A_0 = \frac{1}{2}, \alpha_1 = 1, \beta_{10} = \frac{1}{2}, \text{ и } \tilde{\Delta}y = \frac{1}{2} \cdot (\phi_0 + \phi_1).$$

$$\text{Для } A_1 = 1, \text{ то } A_0 = 0, \alpha_1 = 1, \beta_{10} = \frac{1}{2}, \text{ и } \tilde{\Delta}y = \phi_1.$$

Таким образом эти формулы соответствуют формулам усовершенствованного и модифицированного методов Эйлера. Значит эти методы являются частными случаями методов Рунге-Кутты.

Методы 4-го порядка

Чаще всего используют следующий метод четвертого порядка:

$$\tilde{\Delta}y = \frac{1}{6} \cdot (\phi_0 + 2 \cdot \phi_1 + 2 \cdot \phi_2 + \phi_3)$$

$$\phi_0 = h \cdot f(x_i, y_i)$$

$$\phi_1 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{\phi_0}{2}\right)$$

$$\phi_2 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{\phi_1}{2}\right)$$

$$\phi_3 = h \cdot f(x_i + h, y_i + \phi_2)$$

Приведем возможную процедуру для реализации последних формул в среде MATHCAD. Построим её так, чтобы список её параметров соответствовал списку параметров функций встроенных функций **rkfixed**, **Rkadapt**.

```

Rgk4(Y0, t0, t1, N, f) :=
  z⟨0⟩ ← Y0
  h ← (t1 - t0) / N
  x ← t0
  h2 ← h / 2
  for i ∈ 0.. N - 1
    rez0,i ← x
    s0 ← h · f(x, z⟨i⟩)
    s1 ← h · f(x + h2, z⟨i⟩ + s0 / 2)
    s2 ← h · f(x + h2, z⟨i⟩ + s1 / 2)
    s3 ← h · f(x + h, z⟨i⟩ + s2)
    x ← x + h
    z⟨i+1⟩ ← z⟨i⟩ + (s0 + 2 · s1 + 2 · s2 + s3) / 6
  for j ∈ 1.. length(Y0)
    for i ∈ 0.. N - 1
      rezj,i ← zj-1,i
  rezT

```

Для того, чтобы переделать её для любого другого метода, достаточно изменить в ней только несколько строк.

Другой метод четвертого порядка имеет следующие формулы:

$$\tilde{\Delta}y = \frac{1}{8} \cdot (\phi_0 + 3 \cdot \phi_1 + 3 \cdot \phi_2 + \phi_3)$$

$$\phi_0 = h \cdot f(x, y)$$

$$\phi_1 = h \cdot f\left(x + \frac{h}{3}, y + \frac{\phi_0}{3}\right)$$

$$\phi_2 = h \cdot f\left(x + \frac{h}{3}, y - \frac{\phi_0}{3} + \phi_1\right)$$

$$\phi_3 = h \cdot f(x + h, y + \phi_0 - \phi_1 + \phi_2)$$

Методы 3-го порядка:

$$\begin{cases} A_0 + A_1 + A_2 = 1 \\ A_1 \cdot \alpha_1 + A_2 \cdot \alpha_2 = \frac{1}{2} \\ A_1 \cdot \alpha_1^2 + A_2 \cdot \alpha_2^2 = \frac{1}{3} \\ A_2 \cdot \alpha_1 \cdot \beta_{21} = \frac{1}{6} \\ \beta_{20} + \beta_{21} = \alpha_2 \\ \beta_{10} = \alpha_1 \end{cases}$$

$$\tilde{\Delta}y = \frac{1}{6} \cdot (\phi_0 + 4 \cdot \phi_1 + \phi_2)$$

$$\phi_0 = h \cdot f(x, y)$$

$$\phi_1 = h \cdot f\left(x + \frac{h}{2}, y + \frac{\phi_0}{2}\right)$$

$$\phi_2 = h \cdot f(x + h, y - \phi_0 + 2 \cdot \phi_1)$$

Метод 5-го порядка:

$$\tilde{\Delta}y = \frac{1}{192} \cdot (23 \cdot \phi_0 + 125 \cdot \phi_2 + 81 \cdot \phi_4 + 125 \cdot \phi_5)$$

$$\phi_0 = h \cdot f(x, y)$$

$$\phi_1 = h \cdot f\left(x + \frac{h}{3}, y + \frac{\phi_0}{3}\right)$$

$$\phi_2 = h \cdot f\left(x + \frac{2 \cdot h}{5}, y + \frac{4 \cdot \phi_0 + 6 \cdot \phi_1}{25}\right)$$

$$\phi_3 = h \cdot f\left(x + h, y + \frac{4 \cdot \phi_0 + 12 \cdot \phi_1 + 15 \cdot \phi_2}{4}\right)$$

$$\phi_4 = h \cdot f\left(x + \frac{2 \cdot h}{3}, y + \frac{6 \cdot \phi_0 + 90 \cdot \phi_1 + 50 \cdot \phi_2 + 8 \cdot \phi_3}{81}\right)$$

$$\phi_5 = h \cdot f\left(x + \frac{4 \cdot h}{5}, y + \frac{6 \cdot \phi_0 + 36 \cdot \phi_1 + 10 \cdot \phi_2 + 8 \cdot \phi_3}{75}\right)$$

**Пример получения уравнений для формул первого порядка
точности**

Как отмечено выше, решение на отдельном шаге вычисляется по формуле

$$y_{n+1} \approx y_n + \tilde{\Delta}y$$

И вся задача заключается в том, чтобы вычислить $\tilde{\Delta}y$.

Предложение Рунге-Кутты заключается в следующем. Аппроксиматор представим в виде

$$\tilde{\Delta}y = \sum_{i=0}^q A_i \cdot \phi_i \quad (5.13)$$

При этом вводятся три набора констант:

$$1. \quad A_0, A_1, \dots, A_q \quad (A)$$

$$2. \quad \alpha_1, \alpha_2, \dots, \alpha_q \quad (\alpha)$$

$$3. \quad \begin{array}{l} \beta_{10} \\ \beta_{20} \quad \beta_{21} \\ \dots\dots\dots \\ \beta_{q0} \quad \beta_{q1} \quad \beta_{q2} \quad \dots\dots \quad \beta_{q(q-1)} \end{array} \quad (\beta)$$

При этом величины ϕ_i последовательно вычисляются следующим образом:

$$\phi_0 = h \cdot f(x_i, y_i)$$

$$\phi_1 = h \cdot f(x_i + \alpha_1 \cdot h, y_i + \beta_{10} \cdot \phi_0)$$

$$\phi_2 = h \cdot f(x_i + \alpha_2 \cdot h, y_i + \beta_{20} \cdot \phi_0 + \beta_{21} \cdot \phi_1)$$

.....

$$\phi_q = h \cdot f(x_i + \alpha_q \cdot h, y_i + \beta_{q0} \cdot \phi_0 + \beta_{q1} \cdot \phi_1 + \dots + \beta_{q(q-1)} \cdot \phi_{q-1})$$

Очередные ϕ вычисляются через предыдущие.

Если ввести величину:

$$r_q = \Delta y - \sum_{i=0}^q A_i \cdot \phi_i,$$

то это и будет локальная погрешность метода.

Считая, что h достаточно мал, разложим эту погрешность в окрестности нуля по формуле Тейлора:

$$r_q = \sum_{j=0}^k \frac{h^j}{j!} r_q^{(j)}(h) \Big|_{h=0} + \frac{h^{k+1}}{(k+1)!} r_q^{(k+1)}(\theta h), \quad \theta \in [0; 1]$$

Основная идея метода Рунге-Кутты заключается в том, чтобы обеспечить совпадение неизвестной добавки Δy с её приближенным представлением по формуле (8) до члена с возможно более высокими степенями h .

Это означает, что $r_q^{(j)}(h) \Big|_{h=0}$ следует сделать равной нулю для $j=0, 1, \dots$ до возможно большего k . А обеспечить это можно выбором соответствующего набора A, α, β . Чем больший набор A, α, β , тем большей точности можно достичь. Число k – порядок точности метода.

Рассмотрим ситуацию для построения методов 1-го порядка точности: $k=1$.

$$\Delta y = A_0 \cdot \phi_0$$

$$\phi_0 = h \cdot f(x, y)$$

$$r_0(h) = y(x+h) - y(x) - h \cdot A_0 \cdot f(x, y)$$

$$r_0'(h) = y'(x+h) - A_0 \cdot f(x, y)$$

$$r_0''(h) = y''(x + h)$$

Последняя производная не зависит от A, α, β , то равенства нулю этого выражения при произвольной функции $f(x, y)$ мы добиться не сможем, значит

$$\sum_{i=0}^q A_i \cdot \phi_i = A_0 \cdot \phi_0$$

Приравняем к нулю первую производную:

$$y'(x + h) - A_0 \cdot f(x, y) = 0.$$

С учетом

$$\frac{dy}{dx} = f(x, y)$$

$$(1 - A_0) \cdot f(x, y) = 0$$

$$A_0 = 1$$

$$\Delta y \approx \phi_0 = h \cdot f(x, y)$$

Окончательно

$$y(x + h) \approx y(x) + h \cdot f(x, y)$$

Как видно метод Эйлера – частный случай метода Рунге-Кутты, т.е. метод 1-го порядка. Аналогично можно показать, что модифицированный и усовершенствованный методы Эйлера частные случаи метода Рунге-Кутты, 2-го порядка.

Оценка погрешности в решении дифференциальных уравнений

Основная задача приближённого вычисления заключается в том, чтобы получить результат с нужной точностью за экономически обоснованное время расчета. Точность зависит от величины шага. Чем меньше шаг, тем обычно выше точность. Конечно же, при измельчении шага может накапливаться и погрешность, связанная с ограничением разрядной сетки компьютера.

Предположим, что на заданном количестве шагов выполняется решение ДУ. Возникает вопрос, как оценить точность решения в последний момент времени наблюдения. Рекомендации:

1. Применение более грубой расчетной схемы и качественная оценка решения.
2. Повторное решение уравнения другим методом и сравнение решений.
3. Метод Рунге.

Метод Рунге

Рассмотрим этот метод подробнее. Предположим, что на отдельном шаге погрешность пропорциональна некоторой величине с заданным коэффициентом A и на каждом шаге одна и та же:

$$A \cdot h^{m+1}$$

A – заданное число, m – порядок метода.

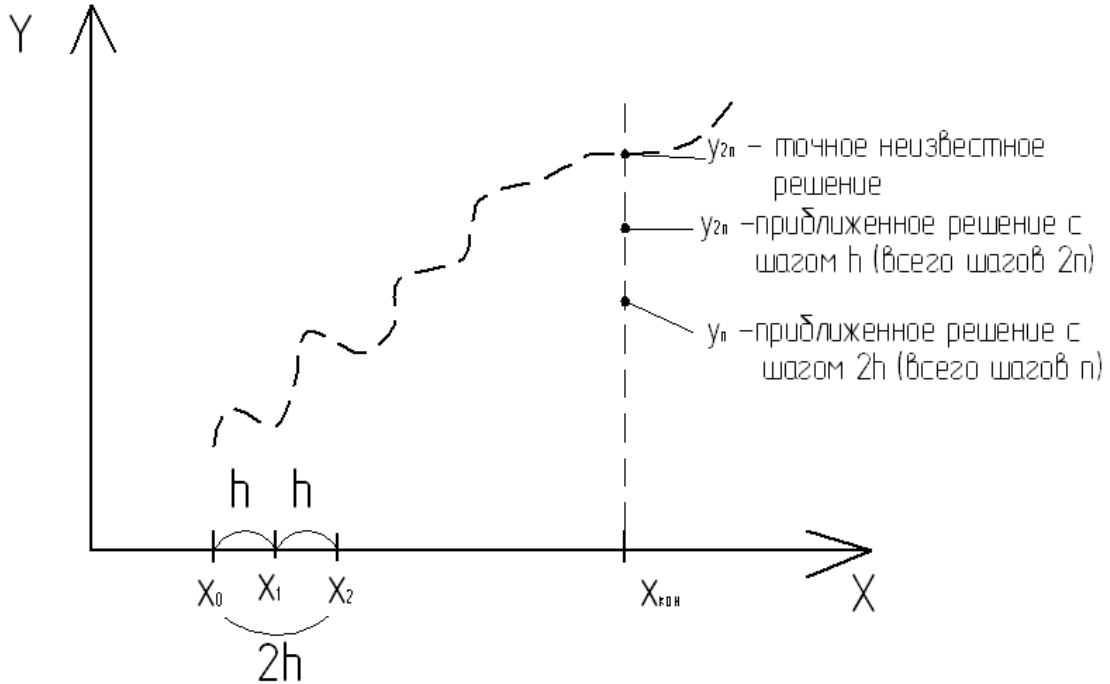


Рисунок 5.17. Иллюстрация метода Рунге.

После получения численного решения при прохождении интервала с шагом h через $2n$ шагов приближенное решение \tilde{y}_{2n}^* отклоняется от точного y_{2n} на величину, равную

$$\tilde{y}_{2n}^* - y_{2n} = 2 \cdot n \cdot A \cdot h^{m+1} \quad (5.14)$$

Если теперь выполнить решение с удвоенным шагом $2h$ (при этом шагов n), то полученное приближенное решение \tilde{y}_n^* отклоняется от точного на величину

$$\tilde{y}_n^* - y_n = n \cdot A \cdot (2 \cdot h)^{m+1} \quad (5.15)$$

(5.14) и (5.15) представляют собой систему линейных уравнений относительно двух неизвестных A , и y_{2n} .

Решая эту систему, получаем

$$A = \frac{\tilde{y}_n^* - \tilde{y}_{2n}^*}{2 \cdot n \cdot (2^m - 1) \cdot h^{m+1}}$$

$$y_{2n} = \tilde{y}_{2n}^* - \frac{\tilde{y}_n^* - \tilde{y}_{2n}^*}{2^m - 1} \quad (5.16)$$

Таким образом погрешность решения можно написать так

$$E_{2n}^* = \frac{|\tilde{y}_n^* - \tilde{y}_{2n}^*|}{2^m - 1} \quad (5.17)$$

Например, для метода 4-го порядка точности формула примет вид

$$E_{2n}^* = \frac{1}{15} \cdot |\tilde{y}_n^* - \tilde{y}_{2n}^*| \quad (5.18)$$

Последовательность этапов для оценки погрешности решения на основе предположения Рунге:

1. Выбрать n – количество подинтервалов на интервале наблюдения.
2. Вычисляем шаг:

$$h = \frac{x_{\text{кон}} - x_{\text{нач}}}{n}$$
3. С помощью выбранного метода m -того порядка точности получить численное решение на всём интервале наблюдения y_n^* .
4. Для шага $\frac{h}{2}$ выполнить решение на $2n$ шагах, получим \tilde{y}_{2n}^* .
5. Воспользоваться формулой (5.17) для оценки погрешностей.

Понятие о «жестких» ДУ

Рассмотрим следующую систему дифференциальных уравнений

$$\begin{cases} \frac{du}{dt} = 998 \cdot u + 1998 \cdot v \\ \frac{dv}{dt} = -999 \cdot u - 1998 \cdot v \end{cases}$$

$$u(0) = 1$$

$$v(0) = 1$$

Точным решением этой системы является:

$$\begin{cases} u = 4 \cdot e^{-t} - 3 \cdot e^{-1000 \cdot t} \\ v = -2 \cdot e^{-t} + 3 \cdot e^{-1000 \cdot t} \end{cases}$$

Очевидно, что после весьма короткого промежутка времени вторые слагаемые практически не сказываются на решении:

$$\begin{cases} u \approx 4 \cdot e^{-t} \\ v \approx -2 \cdot e^{-t} \end{cases}$$

Это пример уравнений, в решении которых присутствуют слагаемые с разными постоянными времени (то есть некоторые слагаемые убывают гораздо быстрее, чем другие). Если использовать метод Эйлера для решения этой системы, то получится

$$u_{n+1} = u_n + h \cdot (998 \cdot u_n + 1998 \cdot v_n)$$

$$v_{n+1} = v_n + h \cdot (-999 \cdot u_n - 1998 \cdot v_n)$$

Используя их для шага $h=0,01$ на конце первого шага, получим

$$u_1 = 1 + 0,01 \cdot (998 + 1998) = 30,96$$

$$v_1 = 1 + 0,01 \cdot (-999 - 1998) = -28,96$$

Эти значения не имеют ничего общего с точным решением, и дальнейшие итерации уведут ещё дальше. Это происходит из-за быстро убывающих слагаемых. На этом примере видно, что существуют такие

уравнения, которые накладывают очень жёсткие требования на шаг интегрирования (чтобы достаточно точно воспроизвести слагаемые $3 \cdot e^{-1000 \cdot t}$ в решении потребуется очень мелкий шаг $h = 10^{-5}$). Отсюда происхождение термина «жёсткие» ДУ.

В то же время понятно, что воспроизводить эти слагаемые в решении совершенно необязательно. Таким образом для решения жёстких уравнений стандартные методы непригодны, так как они потребуют очень большое количество машинного времени или вовсе не приведут к приемлемому результату из-за накопления локальной погрешности. Для решения таких классов уравнений разработаны специальные методы Бурлиша-Штерна (в MATHCAD процедура **Stiff**).

Методы прогноза и коррекции для решения ОДУ

В этих методах для определения решения на конце очередного шага используются данные о решении на нескольких предшествующих шагах. Такие методы называются несамостартующими или несаморазгоняющимися, то есть для их запуска необходимо выполнить несколько шагов с помощью другого метода, самостартующего.

В методах прогноза и коррекции используются две формулы:

- формула прогноза решения;
- формула коррекции решения.

Вычисления происходят следующим образом. Сначала по формуле прогноза и исходным значениям, входящих в правую часть этой формулы, определяют $y_{n+1}^{(0)}$ (приближение нулевое). Индекс в круглых скобках обозначает номер приближения в пределах данного шага.

Прогнозируемое значение является одним из последовательности $y_{n+1}^{(0)}, y_{n+1}^{(1)}, \dots$, располагающихся в порядке возрастания точности. По этому значению с использованием правой части ДУ вычисляется $(y_{n+1}^{(0)})' = f(x_{n+1}, y_{n+1}^{(0)})$. Вычисленная величина подставляется в формулу коррекции для вычисления уточнённого значения $y_{n+1}^{(j+1)}$ и т.д.

Если очередное значение производной недостаточно близко к предыдущему, то оно вводится в формулу коррекции и итерационный процесс продолжается. Если это значение достаточно близко к предыдущему, то оно используется для окончательного вычисления.

Метод предиктор-корректор Адамса-Баушфорта

В этом методе используется следующая формула прогноза:

$$y_{n+1} = y_n + \frac{1}{24} \cdot h \cdot (55 \cdot y'_n - 59 \cdot y'_{n-1} + 37 \cdot y'_{n-2} - 9 \cdot y'_{n-3}) + \frac{251}{700} \cdot h^5 \cdot y^{(5)}$$

Формула коррекции:

$$y_{n+1} = y_n + \frac{1}{24} \cdot h \cdot (9 \cdot y'_{n+1} - 19 \cdot y'_n - 19 \cdot y'_{n-1} + y'_{n-2}) + \frac{19}{720} \cdot h^5 \cdot y^{(5)}$$

Последние слагаемые в этих формулах в вычислениях не участвуют и приведены только для оценки точности решения. И та и другая формулы имеют четвертый порядок точности. y'_{n+1} во второй формуле становится известным только после применения формулы прогноза.

Метод стрельбы (пристрелки) решения граничной задачи для ОДУ

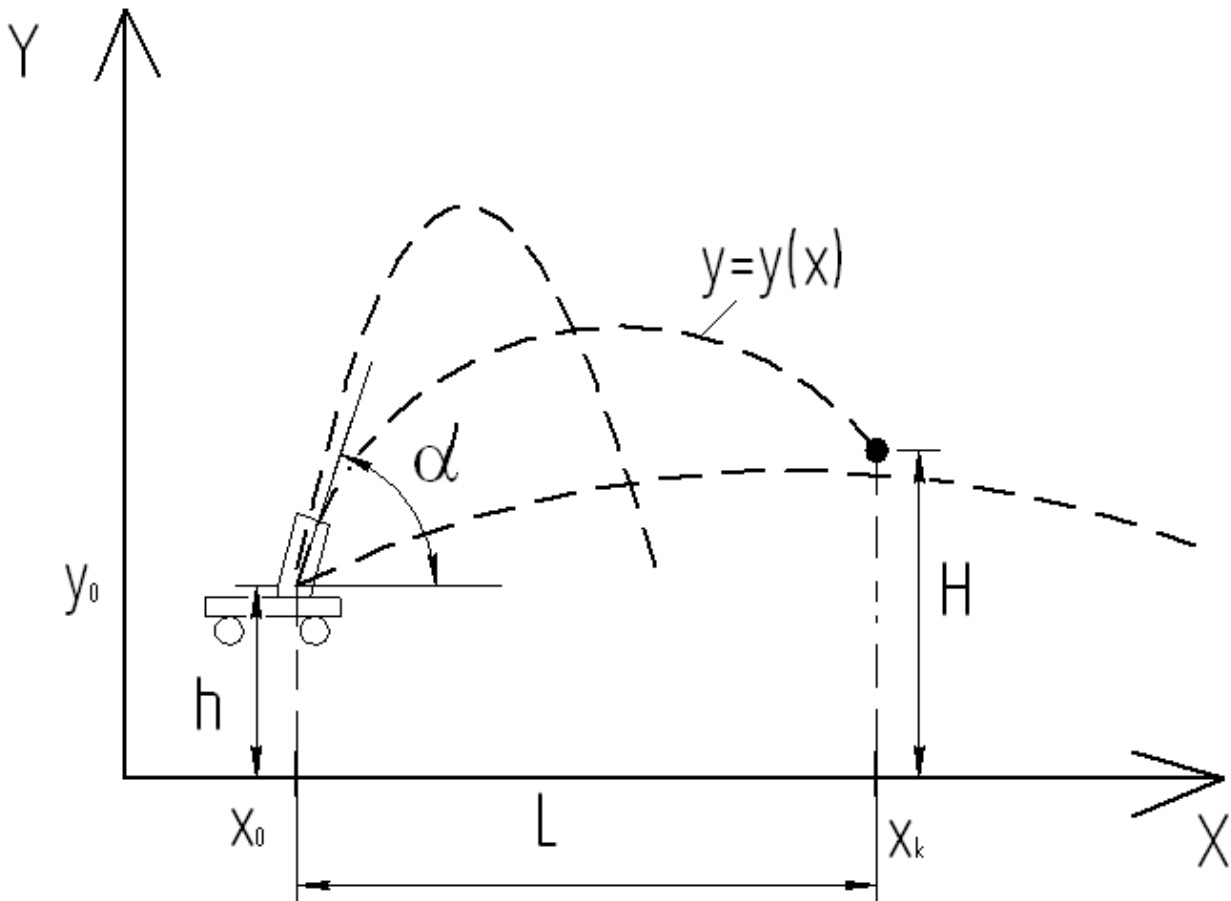


Рисунок 5.18. Иллюстрация метода стрельбы для решения граничной задачи для ОДУ

Если считать, что траектория движения имеет вид $y = y(x)$ и является решением ДУ:

$$\frac{d^2y}{dx^2} = f(x, y) \quad (5.19)$$

с начальным условием:

$$y(x_0) = h, \quad (5.20)$$

то, как видно из рисунка 5.18, это решение должно удовлетворять ещё и условию:

$$y(x_k) = H \quad (5.21)$$

Таким образом для решения заданы два условия на *границах (краях)* $[x_0, x_k]$ (5.20) и (5.21). Такая задача называется *граничной (или краевой)*.

Понятно, что можно подобрать угол наклона орудия α таким образом, чтобы удовлетворить граничному условию (5.21) на правом конце.

Иначе говоря, нужно найти такое решение задачи (5.19), (5.20), чтобы удовлетворить *дополнительному начальному условию*

$$y'(x_0) = tg(\alpha),$$

где α - требуемый угол.

Поэтому такой метод называется методом стрельбы (пристрелки).

Значит решить такую задачу можно многократно решая задачу (5.19), (5.20) и, подбирая решение так, чтобы выполнялось условие (5.21). Таким образом, вместо решения задачи (5.19), (5.20), (5.21), подбирается решение задачи:

$$\frac{d^2y}{dx^2} = f(x, y)$$

$$y(x_0) = h$$

$$y'(x_0) = tg(\alpha)$$

Решение краевой задачи сводится к многократному решению задачи с начальными условиями.

Приведем пример решения аналогичной задачи с помощью встроенных функций системы MATHCAD.

Пример 1.

$$4 \frac{d^2x}{dt^2} + x(t) = t$$

$$x(0)=5 \quad x(5)=13.5$$

$$t \in [0; 5]$$

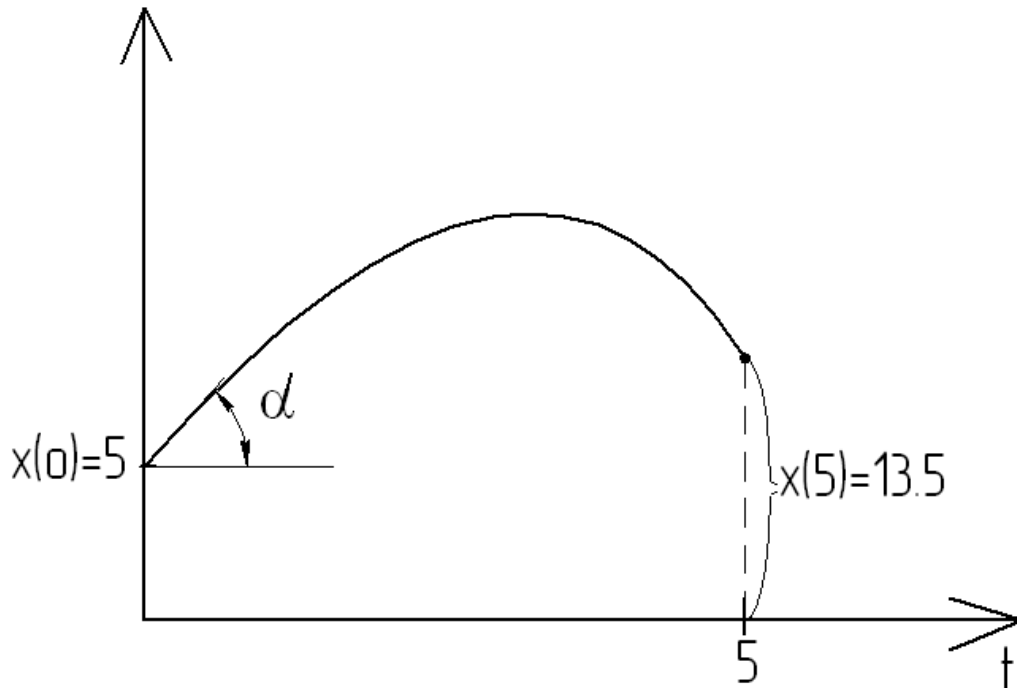


Рисунок 5.19. Вид граничных условий. Пример решения задачи по методу стрельб

1. Задаем вектор начальных значений функции
 $g_0 := 5$
2. задаем начальный и конечный моменты времени интегрирования

$$t_0 := 0$$

$$t_1 := 5$$

$$\begin{cases} \frac{dx_0}{dt} = x_1 \\ \frac{dx_1}{dt} = \frac{t - x_0}{4} \end{cases}$$

3. Записываем правые части в виде процедуры:

$$D(t, x) := \begin{pmatrix} x_1 \\ \frac{t - x_0}{4} \end{pmatrix}$$

4. Определяем функцию нагрузки для задания пробного вектора начальных условий:

$$load(t, v) := \begin{pmatrix} 5 \\ v_0 \end{pmatrix}$$

где 5 – значение первой функции в начальный момент времени;
 v_0 - подбираемое значение второй функции в начальный момент времени.

5. Определяем функцию невязки компонент пробного вектора решения и соответствующий компонент нужного вектора для ОДУ на правой границе.

$$score(t, w) := w_0 - 13.5$$

Используем встроенную функцию, которая подбирает нужные начальные условия методом стрельб.

$$IC := sbval(g, t0, t1, D, load, score)$$

$IC = (11.448)$. (Это подобранное значение первой производной.)

Построим для контроля график полученного решения (рисунок 5.20) другим способом с помощью блока Given.

Формируем вектор начальных условий:

$$Ybeg := load(0, IC)$$

$$Ybeg = \begin{pmatrix} 5 \\ 11.448 \end{pmatrix}$$

$$N := 1000$$

$$S := rkfixed(Ybeg, t0, t1, N, D)$$

Пример файла для реализации подобной задачи в MATHCAD представлен ниже:

$$4 \cdot \frac{d^2}{dt^2}x(t) + x(t) = t$$

$$x(0) = 4 \quad x(5) = 13.5$$

Given

$$4 \cdot \frac{d^2}{dt^2}x(t) + x(t) = t$$

$$x(0) = 4 \quad x(5) = 13.5$$

$$x := \text{Odesolve}(t, 5)$$

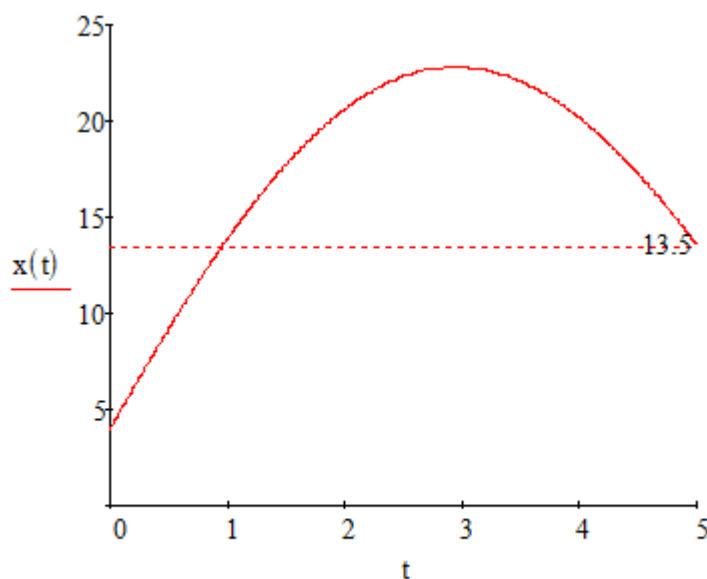


Рисунок 5.20. График полученного решения с помощью блока Given

+

$$g_0 := 4$$

$$t_0 := 0$$

$$t_1 := 5$$

$$D(t, X) := \begin{pmatrix} X_1 \\ \frac{t - X_0}{4} \end{pmatrix}$$

$$\text{load}(t, v) := \begin{pmatrix} g_0 \\ v_0 \end{pmatrix}$$

$$\text{score}(t, w) := w_0 - 13.5$$

$$\text{IC} := \text{sbval}(g, t_0, t_1, D, \text{load}, \text{score})$$

$$\text{IC} = (10.779)$$

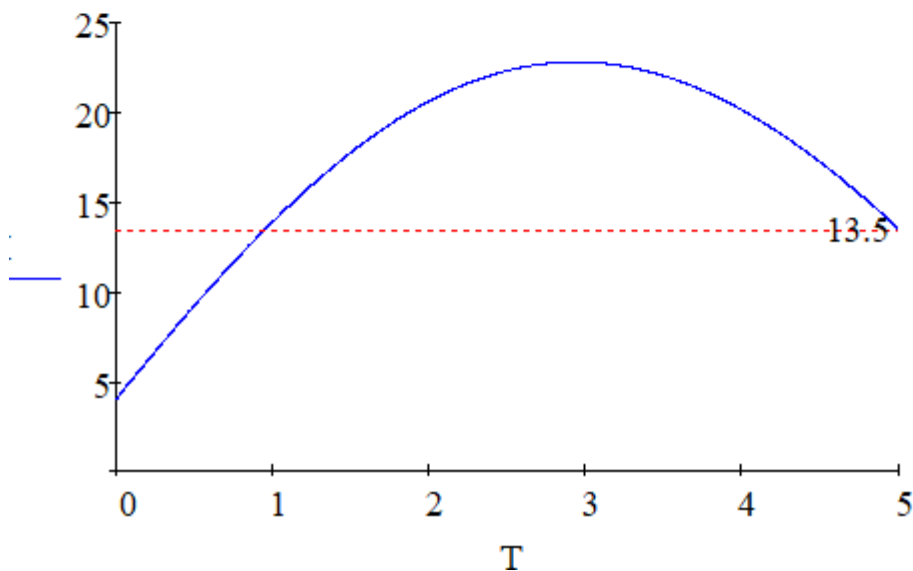
$$\text{ic} := \text{load}(0, \text{IC}) \quad \text{ic} = \begin{pmatrix} 4 \\ 10.779 \end{pmatrix}$$

$$N := 1500$$

$$S := \text{rkfixed}(\text{ic}, t_0, t_1, N, D)$$

$$T := S^{(0)}$$

$$X := S^{(1)}$$



Пример 5.

Решение задачи об изгибе балки переменного сечения под действием распределенной нагрузки.

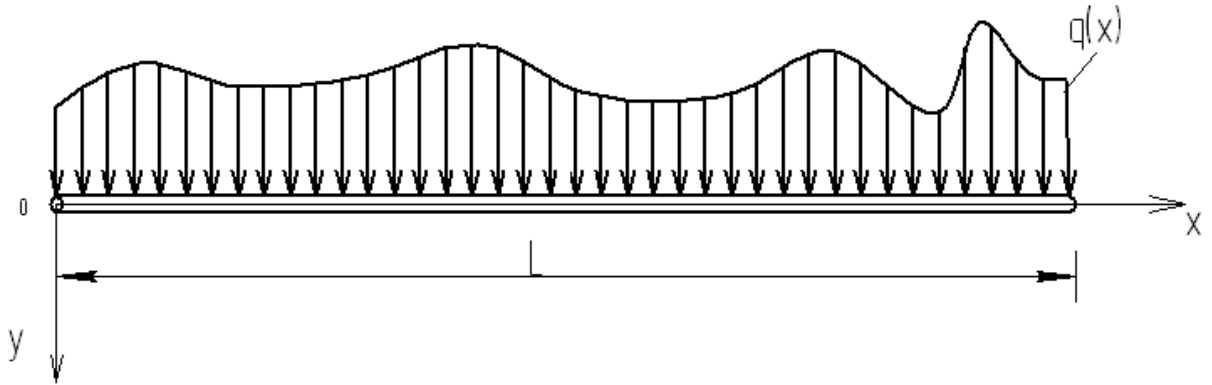


Рисунок 5.21. Балка переменного сечения под действием распределенной нагрузки

Прогиб балки в точке x описывается функцией $y(x)$, которая является решением следующего уравнения (дифференциальное уравнение изогнутой оси балки):

$$\frac{d^4 y}{dx^4} = \frac{q(x)}{E \cdot J(x)} \quad (5.22)$$

Будем считать, что на левом и на правом концах балки могут быть заданы различные виды закреплений, например

$$\begin{aligned} y(0) &= y_0 \\ y'(0) &= y'_0 \end{aligned}$$

- условия на левом конце

$$\begin{aligned} y(L) &= y_L \\ y'(L) &= y'_L \end{aligned}$$

- условия на правом конце.

Если ввести новую систему функций

$$\begin{aligned} z_0 &= y(x) \\ z_1 &= y^{(1)}(x) \\ z_2 &= y^{(2)}(x) \\ z_3 &= y^{(3)}(x), \end{aligned}$$

то уравнение (5.22) можно заменить системой ОДУ:

$$\begin{cases} \frac{dz_0}{dx} = z_1 \\ \frac{dz_1}{dx} = z_2 \\ \frac{dz_2}{dx} = z_3 \\ \frac{dz_3}{dx} = \frac{q(x)}{E \cdot J(x)} \end{cases}$$

С начальными условиями

$$\begin{aligned}
 z_0(0) &= y(0) \\
 z_1(0) &= y^{(1)}(0) \\
 z_2(0) &= y^{(2)}(0) \\
 z_3(0) &= y^{(3)}(0).
 \end{aligned}$$

Недостающие два значения на левом конце методом стрельбы нужно подобрать так, чтобы выполнить условия на правом конце.

Пример файла для реализации подобной задачи в MATHCAD на основе изложенного выше подхода представлен ниже:

Полиномы Эрмита для сравнения

$\psi_1(t) := (1 - t)^2 \cdot (1 + 2 \cdot t)$	$d\psi_1(t) := \frac{d}{dt} \psi_1(t)$	$d\psi_1(t) \text{ simplify} \rightarrow 6 \cdot t \cdot (t - 1)$
$\psi_1(0) = 1$ $\psi_1(1) = 0$		$d\psi_1(0) \rightarrow 0$ $d\psi_1(1) \rightarrow 0$
$\psi_2(t) := t^2(3 - 2 \cdot t)$	$d\psi_2(t) := \frac{d}{dt} \psi_2(t)$	$d\psi_2(t) \text{ simplify} \rightarrow -6 \cdot t \cdot (t - 1)$
$\psi_2(0) = 0$ $\psi_2(1) = 1$		$d\psi_2(0) \rightarrow 0$ $d\psi_2(1) \rightarrow 0$
$\psi_3(t) := t \cdot (1 - t)^2$	$d\psi_3(t) := \frac{d}{dt} \psi_3(t)$	$d\psi_3(t) \text{ simplify} \rightarrow (t - 1) \cdot (3 \cdot t - 1)$
$\psi_3(0) = 0$ $\psi_3(1) = 0$		$d\psi_3(0) \rightarrow 1$ $d\psi_3(1) \rightarrow 0$
$\psi_4(t) := -t^2 \cdot (1 - t)$	$d\psi_4(t) := \frac{d}{dt} \psi_4(t)$	$d\psi_4(t) \text{ simplify} \rightarrow t \cdot (3 \cdot t - 2)$
$\psi_4(0) = 0$ $\psi_4(1) = 0$		$d\psi_4(0) \rightarrow 0$ $d\psi_4(1) \rightarrow 1$

$\text{begY} := \text{load} \left[\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}, S \right]$ Формируем_вектор
 начальных_условий_на_левом_конце

$\text{begY} = \begin{pmatrix} 1 \\ 0 \\ -6 \\ 12 \end{pmatrix}$ Сформированный_вектор_начальных_условий_на_левом_конце

$N := 1500$ Число_подынтервалов

$\text{REZ} := \text{rkfixed}(\text{begY}, a, b, N, D)$ Решение_метдом_Рунге_Кутты



Рисунок 5.21. Изогнутое состояние балки.

Рассмотрим четыре конкретных случая граничных условий.

Пусть $y_0 := 1$ $y_1 := 0$ $y_2 := 0$ $y_3 := 0$
 $q(x) := 0$ $E := 20$ $J(x) := 0.5$ $a := 0$ $b := 1$
+

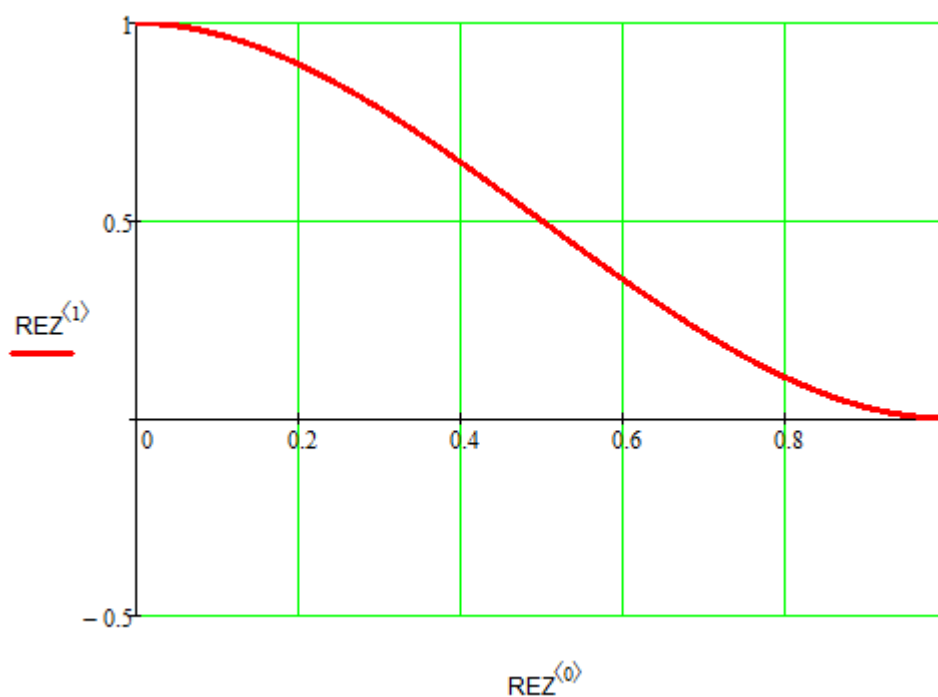
Эту задачу можно решить аналитически. Теоретически показано, что вид деформированной однородной балки для рассмотренных в этом примере граничных условий описывается одной из так называемых функций Эрмита, которая в этом случае имеет вид:

$$F1(t) = (1 - t)^2 \cdot (1 + 2 \cdot t)$$

Эта функция удовлетворяет условиям:

$$F1(0) = 1 \quad F1'(0) = 0 \quad F1(1) = 0 \quad F1'(1) = 0$$

Ниже приведен графический результат (красным цветом) решения этой задачи на основе представленной программы. Результат совпадает с аналитическим решением (соответствующей функцией Эрмита).



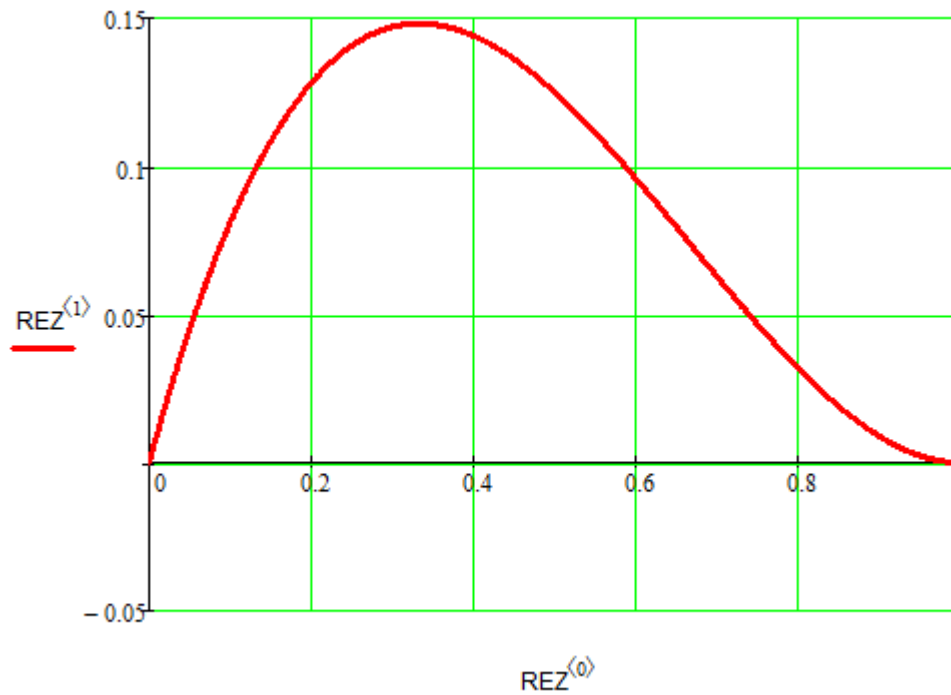
Для других граничных условий

Пусть $y_0 := 0$ $y_1 := 1$ $y_2 := 0$ $y_3 := 0$
 $q(x) := 0$ $E := 20$ $J(x) := 0.5$ $a := 0$ $b := 1$

Функций Эрмита для этого случая $F_2(t) = t^2 \cdot (3 - 2 \cdot t)$

Эта функция удовлетворяет условиям:

$$F_2(0) = 0 \quad F_2'(0) = 1 \quad F_2(1) = 0 \quad F_2'(1) = 0$$



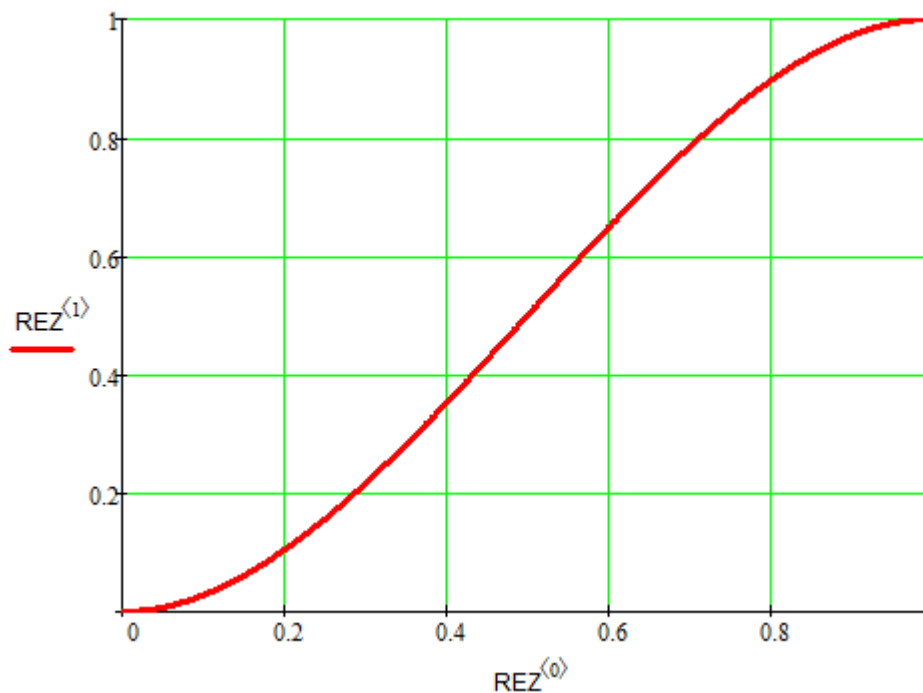
Для других граничных условий

$$\begin{array}{lllll} \text{Пусть} & y_0 := 0 & y_1 := 0 & y_2 := 1 & y_3 := 0 \\ q(x) := 0 & E := 20 & J(x) := 0.5 & a := 0 & b := 1 \end{array}$$

Функций Эрмита для этого случая $F_3(t) = t \cdot (1 - t)^2$

Эта функция удовлетворяет условиям:

$$F_3(0) = 0 \quad F_3'(0) = 0 \quad F_3(1) = 1 \quad F_3'(1) = 0$$



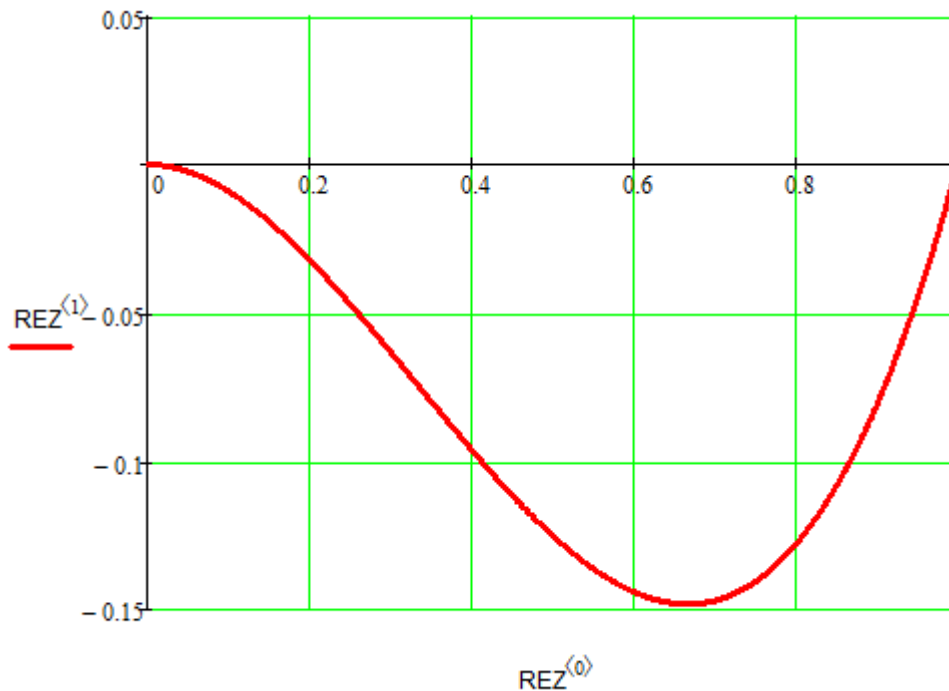
Для других граничных условий

Пусть $y_0 := 0$ $y_1 := 0$ $y_2 := 0$ $y_3 := 1$
 $q(x) := 0$ $E := 20$ $J(x) := 0.5$ $a := 0$ $b := 1$

Функций Эрмита для этого случая $F_4(t) = -t^2 \cdot (1 - t)$

Эта функция удовлетворяет условиям:

$$F_4(0) = 0 \quad F_4'(0) = 0 \quad F_4(1) = 0 \quad F_4'(1) = 1$$



Эти совпадения подтверждают адекватность предложенной программы.

Взаимодействие популяций

В 1925 г. В Италии образовался союз двух учёных: математика Вито Вольтер и зоолог Д'Анконо. Рассматривалась задача о явлении, связанном с периодическим в несколько лет возрастанием и убыванием улова промысловых рыб. Статистика привела к следующему выводу: в период первой мировой войны интенсивность рыбной ловли в средиземном море резко снизилась, что привело к возрастанию числа хищных рыб, питающихся промысловыми рыбами. В результате численность промысловых рыб резко упала, что в свою очередь привело к гибели части хищных рыб, потому что их пища стала исчезать.

Обсуждая это явление они пришли к выводу, что помимо внешних факторов (смена времён года, климата) существует причины особого характера, влияющие на популяции животных.

Получение математической модели для системы хищник-жертва

Данная математическая модель относится к задачам из области динамики популяций, для которых требуется построение нелинейной модели.

Рассмотрим модель, содержащую два вида, один вид - хищники, а другой - их добыча. Пусть x_1 , x_2 – популяции жертв и хищников соответственно. Предположим, что между особями одного вида нет соперничества.

Пусть прирост на особь $\frac{\dot{x}_1}{x_1}$ для жертв составляет $a - bx_2$, а и $b > 0$,

где a – скорость размножения жертв в отсутствии хищников в пересчете на одну особь,

где b – коэффициент уничтожения хищниками жертв в пересчете на одну особь,

$-bx_2$ - есть член, учитывающий потери от хищников в пересчете на одну особь.

Популяция хищников уменьшалась бы в отсутствии их пищи (т.е. жертв), так что

$$\frac{\dot{x}_2}{x_2} = -c, \quad c > 0 \quad \text{при } x_1 = 0.$$

Однако, наличие жертв в случае удачной охоты на них компенсирует это уменьшение с коэффициентом d (коэффициент защищенности жертв от хищников), так что при $x_1 > 0$ имеем

$$\frac{\dot{x}_2}{x_2} = -c + dx_1, \quad d > 0.$$

Таким образом, система имеет вид:

$$\begin{aligned} \dot{x}_1 &= (a - bx_2)x_1 \\ \dot{x}_2 &= (-c + dx_1)x_2 \end{aligned} \quad (3.1)$$

где $a, b, c, d > 0$.

Это и есть уравнения Вольтерра – Лотка.

В данной модели считается, что все хищники (и все жертвы) находятся в одинаковых условиях. При этом пищевые ресурсы жертв неограниченны, а хищники питаются исключительно жертвами. Обе популяции живут на ограниченной территории и не взаимодействуют с любыми другими популяциями, также отсутствуют любые другие факторы, способные повлиять на численность популяций.

Сама математическая модель «хищник – жертва» состоит из пары дифференциальных уравнений, которые описывают динамику популяций хищников и жертв в её простейшем случае, когда имеется одна популяция хищников и одна - жертв. Модель характеризуется колебаниями в размерах

обеих популяций, причём пик количества хищников немного отстаёт от пика количества жертв. С данной моделью можно ознакомиться во многих трудах по популяционной динамике или математическому моделированию. Она достаточно широко освещена и проанализирована математическими методами. Однако формулы не всегда могут дать очевидное представление о происходящем процессе.

Интересно узнать, как именно в данной модели зависит динамика популяций от начальных параметров и насколько это соответствует действительности и здравому смыслу, причём увидеть это графически, не прибегая к сложным расчётам. Для этой цели на основе модели Вольтерра была создана программа в среде Mathcad14.

Параметр	Львы	Антилопы
Начальная численность	30	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

ORIGIN := 1

$x_{begin} := \begin{pmatrix} 100 \\ 30 \end{pmatrix}$ начальная численность Антилоп
 начальная численность Львов

c := 0.02 Прирост (смертность) Львов

a := 2 Прирост (смертность) Антилоп

d := 0.001 Межвидовое взаимодействие Львы-Антилопы (коэффициент защищенности жертв от хищников)

b := 0.01 Межвидовое взаимодействие Антилопы-Львы (коэффициент уничтожения хищниками жертв).

Параметр	Львы	Антилопы
Начальная численность	30	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

ORIGIN := 1

$x_{begin} := \begin{pmatrix} 100 \\ 30 \end{pmatrix}$ начальная численность Антилоп
 начальная численность Львов

c := 0.02 Прирост (смертность) Львов

a := 2 Прирост (смертность) Антилоп

d := 0.001 Межвидовое взаимодействие Львы-Антилопы (коэффициент защищенности жертв от хищников)

b := 0.01 Межвидовое взаимодействие Антилопы-Львы (коэффициент уничтожения хищниками жертв).

tbegin := 0 Начальный момент наблюдения

tend := 300 Конечный момент наблюдения

N := 4000 Количество шагов по времени

$P(t, x) := \begin{bmatrix} x_1 \cdot (a - b \cdot x_2) \\ x_2 \cdot (-c + d \cdot x_1) \end{bmatrix}$ ⁺ Правые части системы ОДУ модели "хищник-жертва"

Rez := rkfixed(xbegin, tbegin, tend, N, P)

График изменения количества хищников и количества жертв от времени

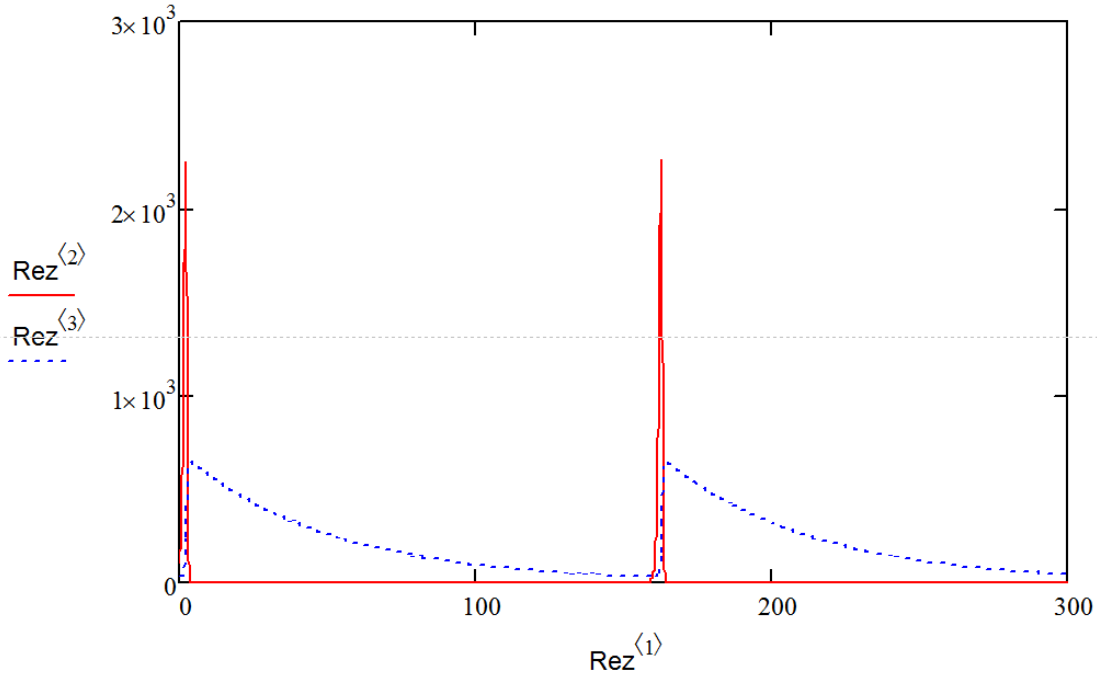
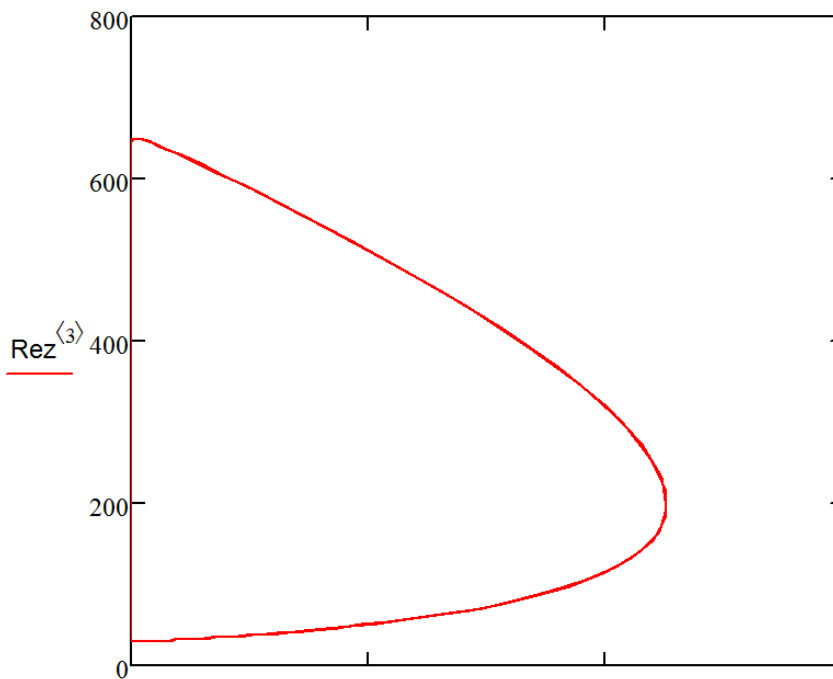


График изменения количества хищников от количества жертв (красная линия)
Это фазовый портрет процесса



Для начала проверим модель на соответствие реальным условиям. Для этого рассмотрим вырожденные случаи, когда в данных условиях обитает только одна из популяций. Теоретически было показано, что при отсутствии хищников популяция жертвы неограниченно возрастает во

времени, а популяция хищника в отсутствие жертвы вымирает, что, вообще говоря, соответствует модели и реальной ситуации (при указанной постановке задачи).

Полученные результаты отражают теоретические: хищники постепенно вымирают (рисунок 3.1), а численность жертвы неограниченно возрастает (рисунок 3.2).

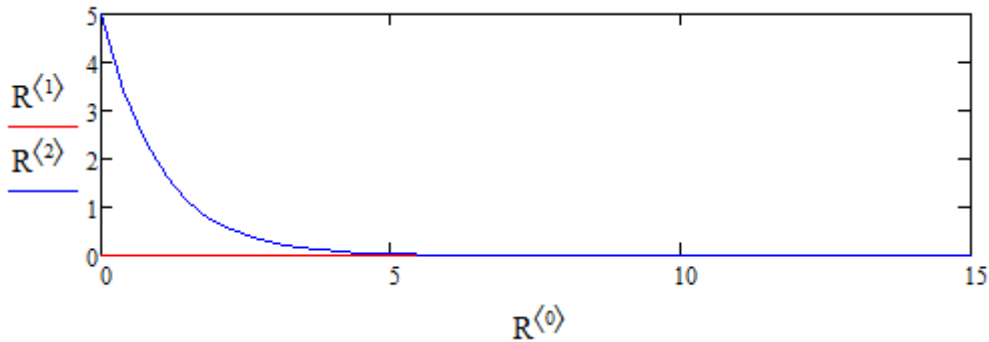


Рисунок 3.1 Зависимость числа хищников от времени при отсутствии жертвы

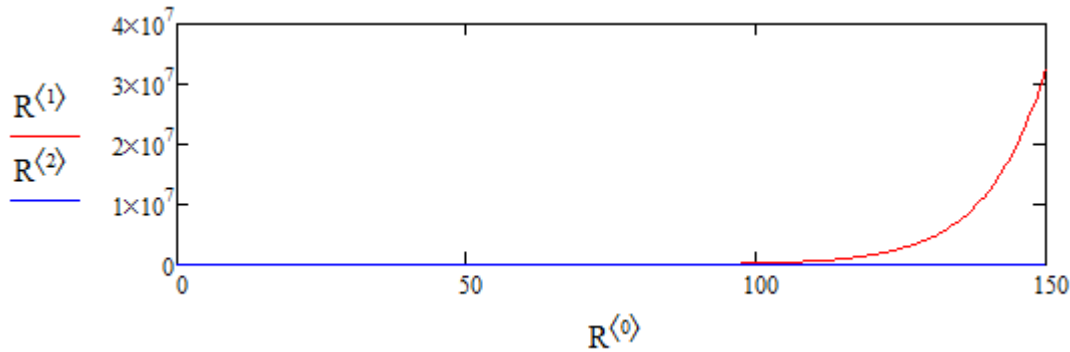


Рисунок 3.2 Зависимость числа жертв от времени при отсутствии хищников

Как видно, в данных случаях система соответствует математической модели.

Задания для самостоятельного изучения процесса на основе математической модели

Рассмотрим, как ведёт себя система при различных начальных параметрах. Пусть имеются две популяции – львы и антилопы – хищники и жертвы соответственно, и заданы начальные показатели. Тогда получаем следующие результаты (рисунок 3.3):

Таблица 3.1. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	30	100
Прирост/Смертность	0,02	2

Межвидовое взаимодействие 0,001 0,01

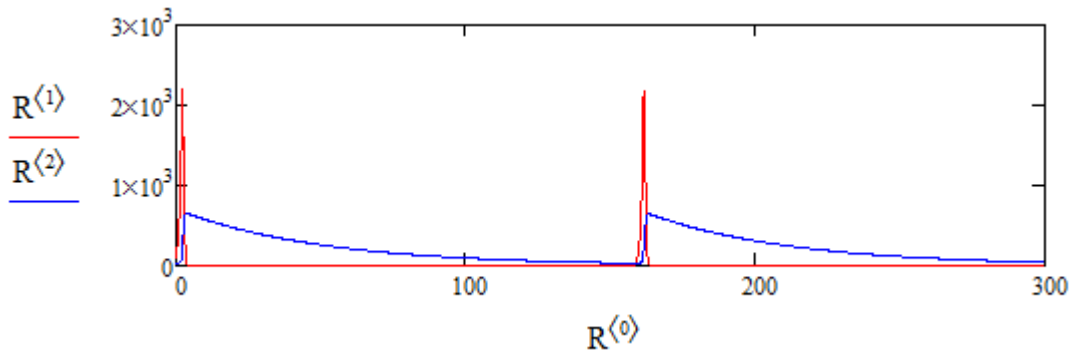


Рисунок 3.3 Система при значении параметров из Таблицы 3.1

Проанализируем полученные данные, исходя из графиков. При первоначальном возрастании популяции антилоп наблюдается прирост числа хищников. Заметим, что пик возрастания популяции хищников наблюдается позже, на спаде популяции жертв, что вполне соответствует реальным представлениям и математической модели. Действительно, рост числа антилоп означает увеличение пищевых ресурсов для львов, что влечёт за собой рост их численности. Далее активное поедание львами антилоп ведёт к стремительному уменьшению численности жертв, что неудивительно, учитывая аппетит хищника, а точнее частоту поедания хищниками жертв. Постепенное снижение численности хищника приводит к ситуации, когда популяция жертвы оказывается в благоприятных для роста условиях. Далее ситуация повторяется с определённым периодом. Делаем вывод, что данные условия не подходят для гармоничного развития особей, так как влекут резкие спады популяции жертв и резкие возрастания обеих популяций.

Положим теперь начальную численность хищника равную 200 особей при сохранении остальных параметров (рисунок 3.4).

Таблица 3.2. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

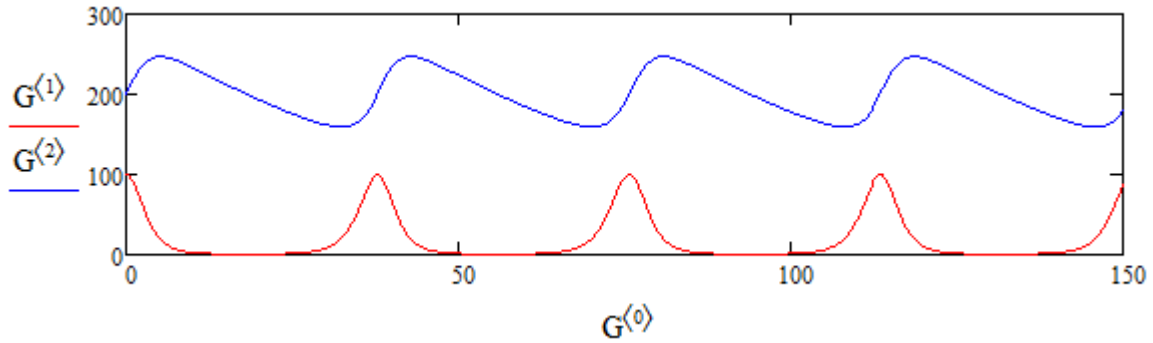


Рисунок 3.4 Система при значении параметров из Таблицы 3.2

Теперь колебания системы происходят более естественно. При данных предположениях система существует вполне гармонично, отсутствуют резкие возрастания и убывания количества численности в обеих популяциях. Делаем вывод, что при данных параметрах обе популяции развиваются достаточно равномерно для совместного обитания на одной территории.

Зададим начальную численность хищника равную 100 особей, численность жертв 200 при сохранении остальных параметров (рисунок 3.5).

Таблица 3.3 Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	100	200
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

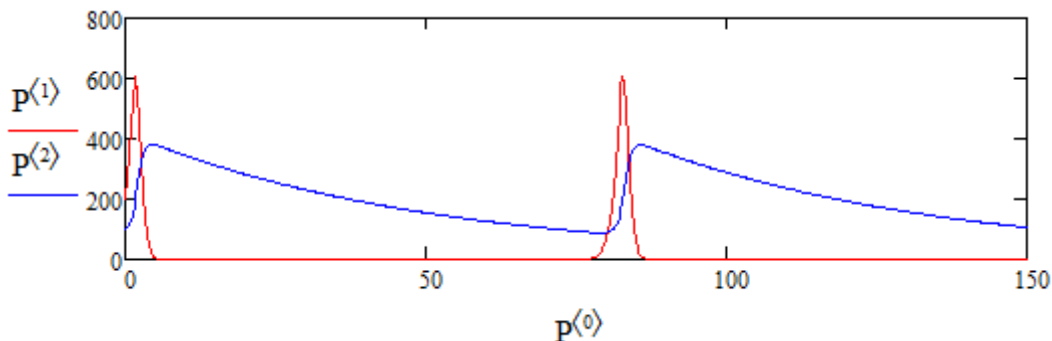


Рисунок 3.5 Система при значении параметров из Таблицы 3.3

В данном случае ситуация близка к первой рассмотренной ситуации. Заметим, что при взаимном увеличении популяций переходы от

возрастания к убыванию популяции жертвы стали более плавными, а популяция хищника сохраняется в отсутствии жертв при более высоком численном значении. Делаем вывод, что при близком отношении одной популяции к другой их взаимодействие происходит более гармонично, если конкретные начальные численности популяций достаточно большие.

Рассмотрим изменение других параметров системы. Пусть начальные численности соответствуют второму случаю. Увеличим коэффициент размножения жертв (рисунок 3.6).

Таблица 3.4. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	4
Межвидовое взаимодействие	0,001	0,01

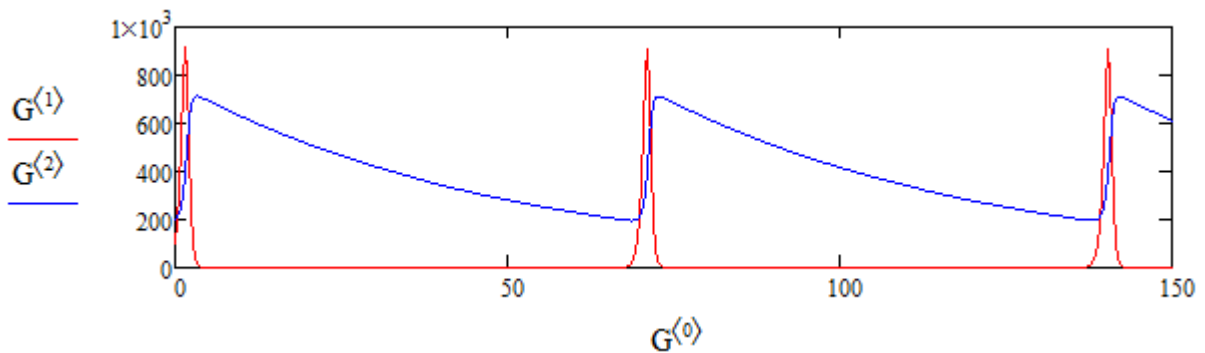


Рисунок 3.6 Система при значении параметров из Таблицы 3.4

Сравним данный результат с результатом, полученным во втором случае. В этом случае наблюдается более быстрый прирост жертвы. При этом и хищник, и жертва ведут себя так, как в первом случае, что объяснялось невысокой численностью популяций. При таком взаимодействии обе популяции достигают пика со значениями, намного большими, чем во втором случае.

Теперь увеличим коэффициент прироста хищников (рисунок 3.7).

Таблица 3.5. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,04	2
Межвидовое взаимодействие	0,001	0,01

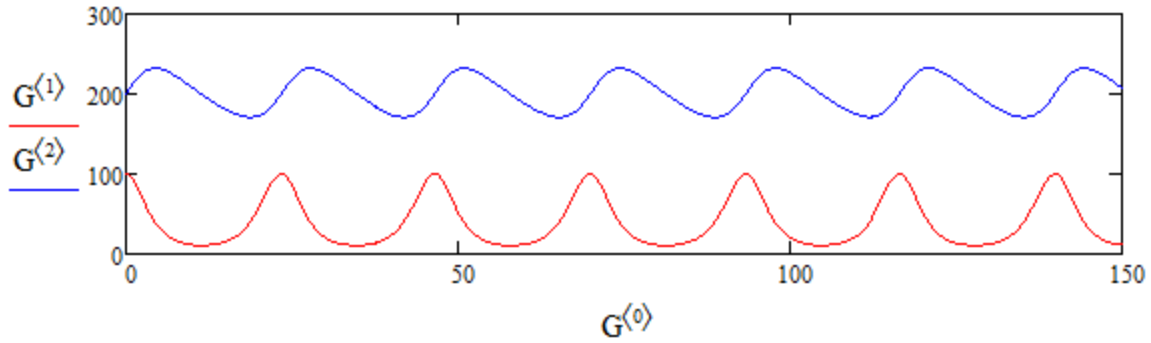


Рисунок 3.7 Система при значении параметров из Таблицы 3.5

Сравним результаты аналогично. В этом случае общая характеристика системы остаётся прежней, за исключением изменения периода. Как и следовало ожидать, период стал меньше, что объясняется быстрым уменьшением популяции хищника в отсутствии жертв.

И, наконец, изменим коэффициент межвидового взаимодействия. Для начала увеличим частоту поедания хищниками жертв:

Таблица 3.6. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,004	0,01

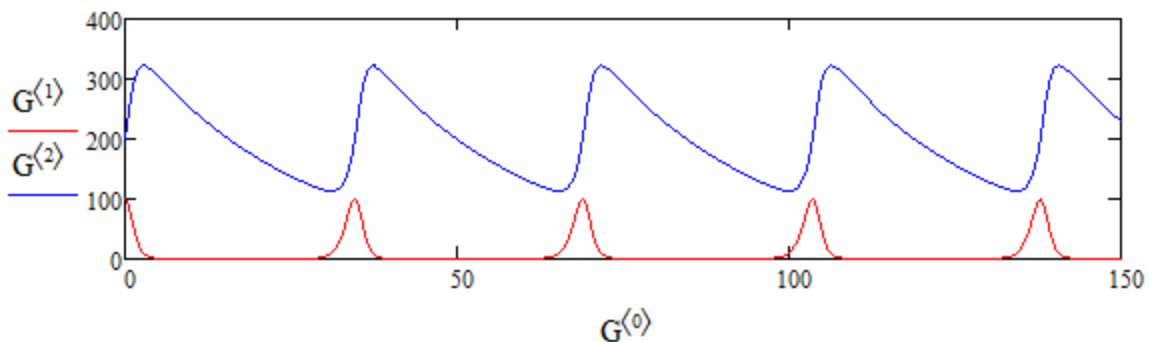


Рисунок 3.8 Система при значении параметров из Таблицы 3.6

Так как хищник поедает жертву чаще, то максимум численности его популяции увеличился по сравнению со вторым случаем, а также уменьшилась разность между максимальным и минимальным значениями численности популяций. Период колебаний системы остался прежним.

И теперь уменьшим частоту поедания хищниками жертв:

Таблица 3.7. Коэффициенты колебательного режима системы

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,0001	0,01

1.

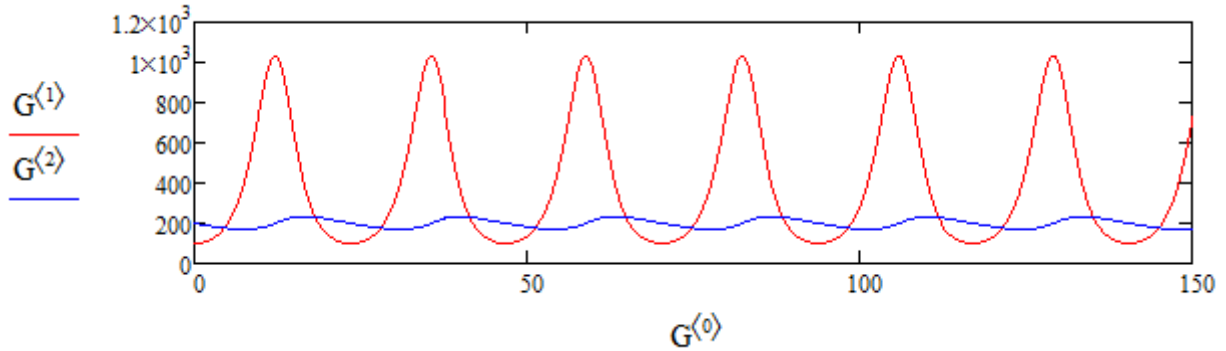


Рисунок 3.9 Система при значении параметров из Таблицы 3.7

Теперь хищник поедает жертву реже, максимум численности его популяции уменьшился по сравнению со вторым случаем, а максимум численности популяции жертвы увеличился, причём в 10 раз. Отсюда следует, что при данных условиях популяция жертвы имеет большую свободу в смысле размножения, ведь хищнику хватает меньшей массы, чтобы насытиться. Также уменьшилась разность между максимальным и минимальным значениями численности популяций.

При попытке моделирования сложных процессов в природе или обществе, так или иначе, возникает вопрос о корректности модели. Естественно, что при моделировании происходит упрощение процесса, пренебрежение некоторыми второстепенными деталями. С другой стороны, существует опасность упростить модель слишком сильно, выкинув при этом важные черты явления вместе с несущественными. Для того чтобы избежать данной ситуации, необходимо перед моделированием изучить предметную область, в которой используется данная модель, исследовать все её характеристики и параметры, а главное, выделить те черты, которые являются наиболее значимыми. Процесс должен иметь естественное описание, интуитивно понятное, совпадающее в основных моментах с теоретической моделью.

Рассмотренная в данной работе модель обладает рядом существенных недостатков. Например, предположение о неограниченных ресурсах для жертвы, отсутствие сторонних факторов, влияющих на смертность обоих видов и т.д. Все эти предположения не отражают реальную ситуацию. Однако, несмотря на все недостатки, модель получила широкое распространение во многих областях, даже далёких от экологии. Это

можно объяснить тем, что система «хищник-жертва» даёт общее представление именно о взаимодействии видов. Взаимодействие с окружающей средой и прочими факторами можно описать другими моделями и анализировать их в совокупности.

Взаимоотношения типа «хищник-жертва» - существенная черта различных видов жизнедеятельности, в которых происходит столкновение двух взаимодействующих между собой сторон. Данная модель имеет место не только в экологии, но и в экономике, политике и других сферах деятельности. Например, одно из направлений, касающихся экономики, это анализ рынка труда, с учётом имеющихся потенциальных работников и вакантных рабочих мест.

Тема 6. Моделирование и методы типа Монте-Карло

Во время Второй Мировой войны при разработке атомной бомбы в лаборатории Лос-Аламоса возникла задача о расчете глубины проникновения нейтронов в вещество. Решить её традиционными методами не удавалось. Станислав Улам и Джон фон Нейман предложили новый метод, основанный на использовании тактики, которой руководствовался игрок в кости.

Этот метод получил кодовое название метода Монет-Карло.

Идея метода такова. Исходный объект или явление заменяется другим объектом или явлением, поведение которого связано со случайными величинами (СВ). Общим у исходного объекта и у вновь сконструированного является решение обеих моделей.

Подбор объект может быть никак не связан с исходным ни по своей природе, ни по математическому описанию.

Общая идея методов типа Монте-Карло заключается в следующем:

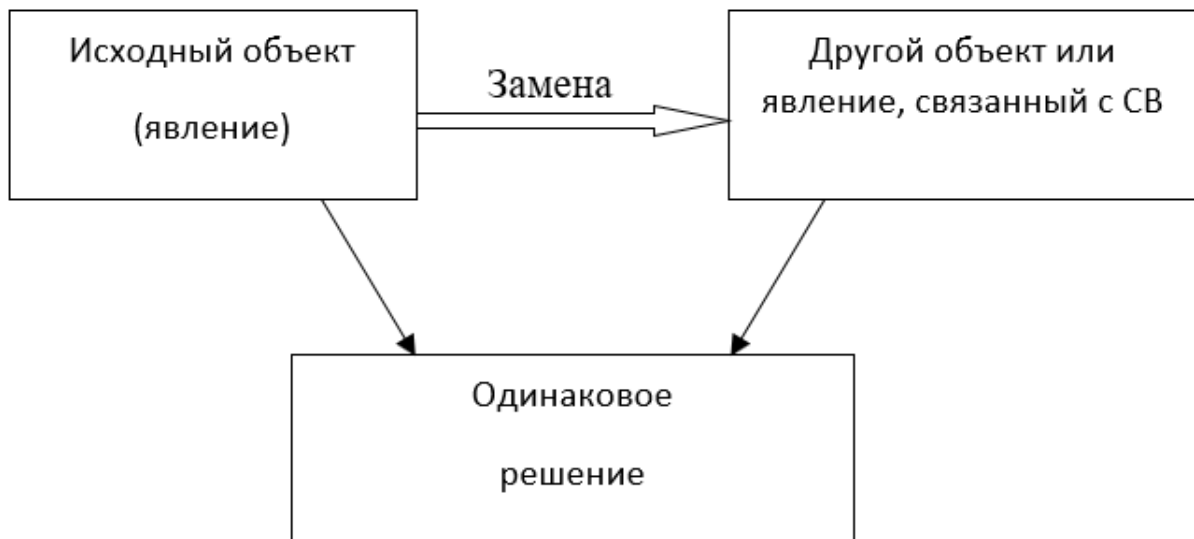


Рисунок 6.1 Схема методов типа Монте-Карло

Детерминированные модели – это такие модели, поведение которых однозначно описывается (определено).

Стохастические (вероятностные) модели - модели, решение которых описывается при той или иной достоверности. Основаны на использовании случайных величин.

Отметим, что методы Монте-Карло могут использоваться независимо от того, был ли исходный объект детерминированным или стохастическим.

Продемонстрируем это на примере задачи Бюффона.

В XVIII веке французский натуралист граф де Бюффон предложил необычное решение задачи о вычислении числа π с любым количеством знаков.

При этом он предложил бросать тонкую иглу на поверхность, размеченную горизонтальными линиями с заданным шагом и подсчитывать количество пересечений иглы с линиями.

В соответствии с рисунком 6.1 исходная задача - вычисление числа π , а сконструированная – эксперимент по бросанию иглы на поверхность. Покажем связь между этими задачами: длина иглы - $2 \cdot l$, поверхность разлинована с шагом $2 \cdot d$.

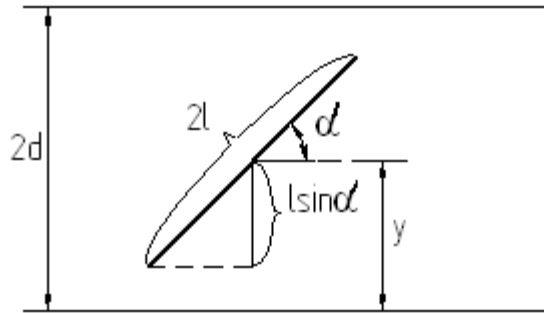


Рисунок 6.2 Возможное положение иглы после бросания.

Достаточно рассмотреть в силу симметрии ситуацию, когда

$$\alpha \in \left[0; \frac{\pi}{2}\right], y \in [0; d].$$

Понятно, что пересечение возможно, если $y \leq l \cdot \sin(\alpha)$.

На рисунке 6.3 изобразим на плоскости (α, y) соответствующую область. Заштрихуем область, где $y \leq l \cdot \sin(\alpha)$.

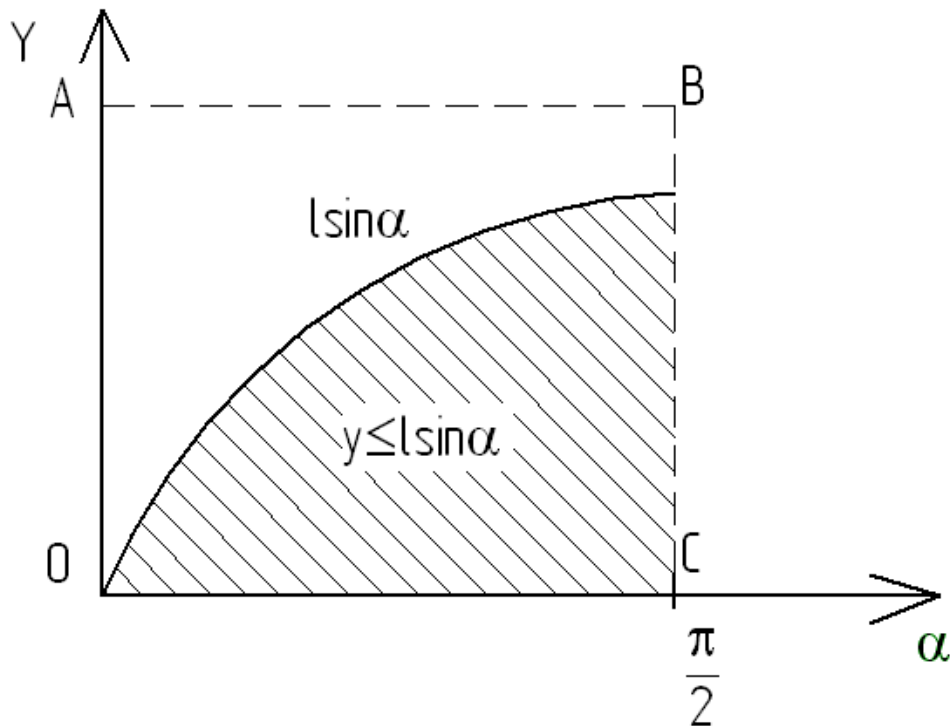


Рисунок 6.3. Заштрихованная область соответствует наличию пересечения.

Так как угол α равномерно распределён на интервале $\left[0; \frac{\pi}{2}\right]$, то вероятность того, что точка попадет в заштрихованную область будет равна отношению площади этой области к площади прямоугольника OABC.

$$P_{\text{попадания}} = \frac{S_{\text{заштрихованная}}}{S_{OABC}}$$

$$S_{OABC} = \frac{\pi}{2} \cdot d$$

$$S_{\text{заштрихованная}} = \int_0^{\frac{\pi}{2}} l \cdot \sin(\alpha) \cdot d\alpha = -l \cdot \cos \alpha \Big|_0^{\frac{\pi}{2}} = l$$

Так как при увеличении числа испытаний частота появления события стремится к вероятности, получим:

$$\lim_{N \rightarrow \infty} \frac{N_{\text{попаданий}}}{N} = p$$

$$\frac{N_{\text{попаданий}}}{N} \approx \frac{l}{\frac{\pi}{2} \cdot d}$$

тогда

$$\pi \approx \frac{2 \cdot N \cdot l}{N_{\text{попаданий}} \cdot d}$$

Реализация этого эксперимента в среде MATHCAD может выглядеть так

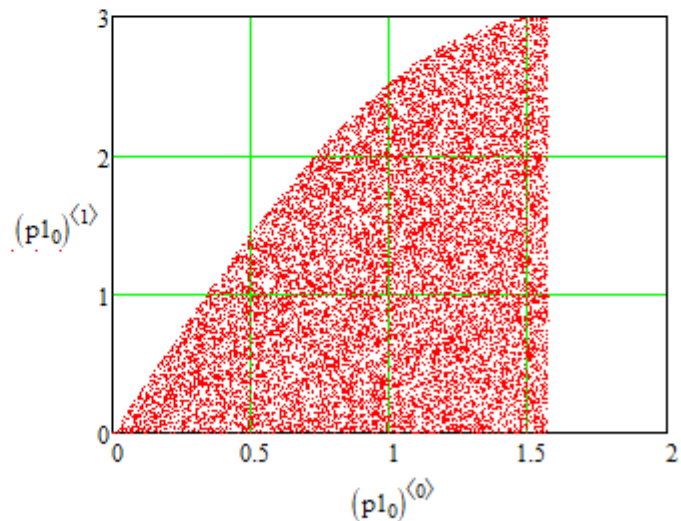
$L := 3$ $d := 5$ $f(x,y) := L \cdot \sin(x) - y$ $g(x,y) := -y$ $n := 40000$

```

p(n) :=
  n1 ← 0
  for i ∈ 1..n
    x ← md( $\frac{\pi}{2}$ )
    y ← md(d)
    if 1 = (g(x,y) ≤ 0) · (f(x,y) ≥ 0)
      n1 ← n1 + 1
      ti,0 ← x
      ti,1 ← y
  rez ← ( t
         n1 )
rez

```

p1 := p(n)



$$pi := \frac{2 \cdot L}{d \cdot \frac{p1_1}{n}}$$

pi = 3.115 полученное приближенное значение пи

Вычисление площадей и объёмов тел с помощью случайных событий.

Идея заключается в том, что область Ω полностью помещается в N -мерный параллелепипед, измерения которого известны и объём $V = a_1 \cdot a_2 \cdot \dots$

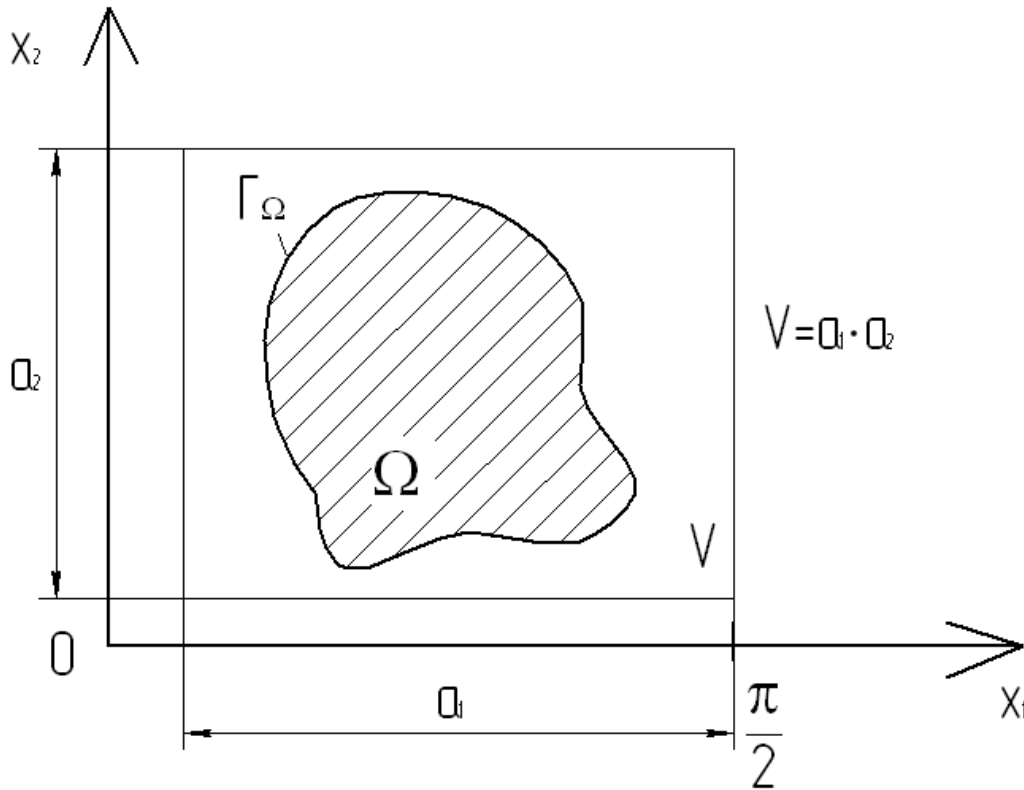


Рисунок 6.4. Область Ω в двумерном пространстве.

Пусть нам удастся набрасывать случайным образом точки в этот параллелепипед V , причем эти точки должны рассеиваться равномерно, тогда очевидно, что

$$\frac{\Omega}{V} \approx \frac{N_{\text{попаданий}}}{N}$$

$$\Omega \approx \frac{V \cdot N_{\text{попаданий}}}{N}$$

$N_{\text{попаданий}}$ - количество попавших точек в область Ω .

N – общее число рассеянных точек.

В изложенном подходе существуют две сложности:

1. Нужно иметь датчик случайных чисел, при обращении к которому удастся получить точку x_i , принадлежащую интервалу (a, b) и выбранную на этом интервале без предпочтения. Распределение такой случайной величины называется **равномерным**.

2. Нужно иметь возможность ответить на вопрос, попала ли точка внутрь области Ω или нет. Для этого нужно аналитически описать уравнение границы Ω : $\Gamma(\Omega)$. Нужно также уметь правильно строить область V , в которой целиком содержится Ω .

Во многих средах программирования существуют процедуры, позволяющие получать значения такой случайной величины.

Например, в среде MATHCAD:

$\text{Rnd}(\varpi)$ – функция, генерирующая одно значение случайной величины, с равномерным распределением на интервале $[0, \varpi]$.

Тогда $\text{Rnd}(b-a)+a$ –выражение, генерирующая одно значение случайной величины, с равномерным распределением на интервале $[a,b]$.

$\text{Runif}(N,a,b)$ – функция, генерирующая N значений случайной величины, с равномерным распределением на $[a,b]$.

Приведём пример документа MATHCAD для вычисления площади фигуры, ограниченной двумя кривыми с уравнениями:

$$\begin{aligned} -(x+5)^2 - y + 6 &= 0 \\ \sin(x) - y - 7 &= 0 \end{aligned}$$

Изобразим эти кривые:

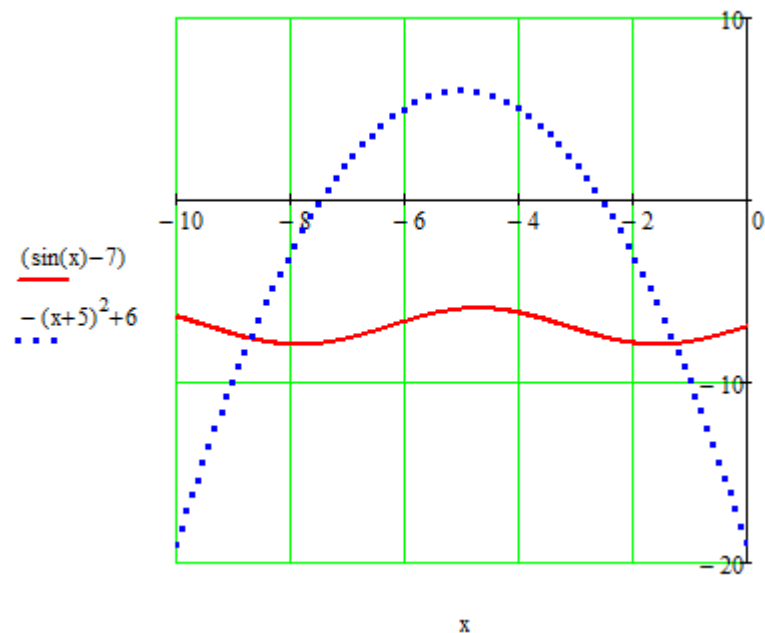


Рисунок 6.5. Кривые, ограничивающие фигуру.

Исходные функции: $f(x, y) := -(x + 5)^2 + 6 - y$ $g(x, y) := \sin(x) - 7 - y$

Вспомогательные функции $ab(x, y) := x$ $cd(x, y) := y$

Начальные "затравочные" значения $x := 0$ $y := 0$

Поиск границ области:

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Minimize}(ab, x, y)$ $a := ab(z_0, z_1)$ $a = -8.697$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Maximize}(ab, x, y)$ $b := ab(z_0, z_1)$ $b = -1.264$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Minimize}(cd, x, y)$ $c := cd(z_0, z_1)$ $c = -8$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Maximize}(cd, x, y)$ $d := cd(z_0, z_1)$ $d = 6$

Функция, вычисляющая количество точек, попавших в область и их координаты:

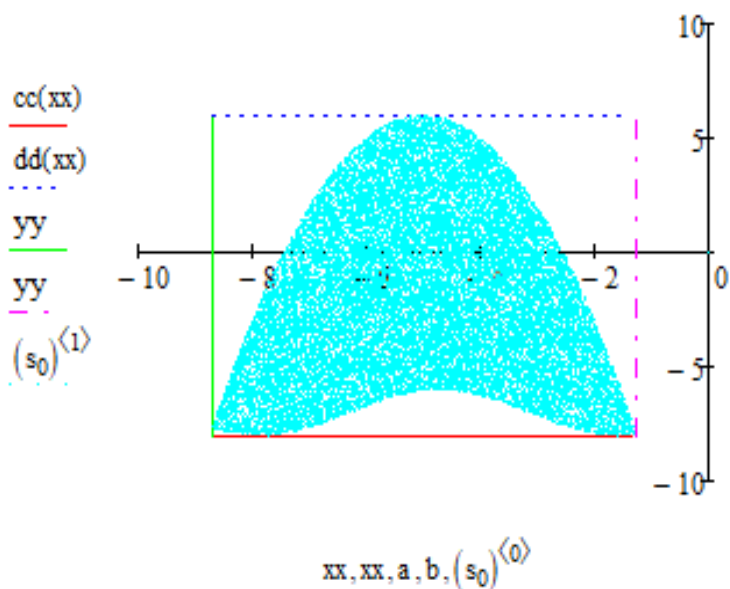
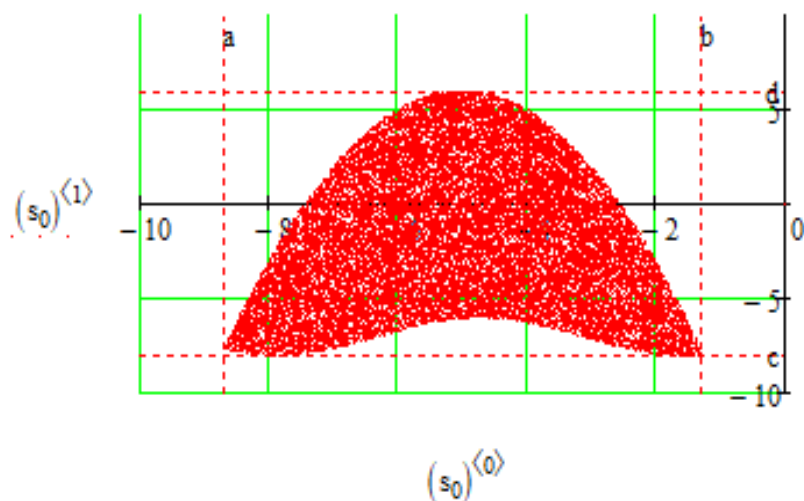
```

p(n) :=
  n1 ← 0
  for i ∈ 1..n
    x ← md(b - a) + a
    y ← md(d - c) + c
    if 1 = (g(x, y) ≤ 0) · (f(x, y) ≥ 0)
      n1 ← n1 + 1
      ti,0 ← x
      ti,1 ← y
  rez ← ( t
          n1 )
  rez
n := 40000
s := p(n)

```

$cc(x) := c$ $dd(x) := d$ $aa(x) := x$ $bb(x) := x$
 $xx := a, a + 0.1.. b$
 $yy := c, c + 0.1.. d$

Графическое отображение:



Сравнение полученного значения с значением вычисленным интегралом

$$s := (b - a) \cdot (d - c) \cdot \frac{s_1}{n} \quad s = 63.999$$

$$s1 := \int_a^b 13 - \sin(x) - (x + 5)^2 dx \quad s1 = 63.452$$

На рисунке красным цветом отображены точки, попавшие в область.

Напишем процедуру, возвращающую векторы абсцисс и ординат точек, попавших внутрь области ограниченной кривыми, при N набросанных точек внутрь двумерной области $[a,b] \times [c,d]$.

```

P(a,b,c,d,N) :=
  k ← 0
  for i ∈ 1..N
    x ← rnd(b - a) + a
    y ← rnd(d - c) + c
    if (g(x,y) ≤ 0)(f(x,y) > 0)
      k ← k + 1
      Pk,1 ← x
      Pk,2 ← y
  P
  
```

Главным недостатком статистических методов при моделировании является их медленная сходимость. Она пропорциональна величине $\frac{1}{\sqrt{n}}$, где n – количество выполненных экспериментов.

Например, для получения очередного знака в числе π потребуется выполнить в 100 раз больше экспериментов, чем для вычисления предыдущего знака. В общем случае справедливо следующее утверждение:

$$P \left\{ \left| \frac{N_{\text{попаданий}}}{N_{\text{общее}}} - \frac{\text{объём}\Omega}{\text{объём}V} \right| < \sqrt{\frac{\frac{\text{объём}\Omega}{\text{объём}V} \cdot \left(1 - \frac{\text{объём}\Omega}{\text{объём}V}\right)}{\xi \cdot N_{\text{общее}}}} \right\} \geq 1 - \xi$$

ξ - задаваемая величина.

P – достоверность того, что отличие вычисленной величины от истинной не будет

превосходить $\sqrt{\frac{\frac{\text{объём}\Omega}{\text{объём}V} \cdot \left(1 - \frac{\text{объём}\Omega}{\text{объём}V}\right)}{\xi \cdot N_{\text{общее}}}}$ и эта достоверность окажется не меньше $1 - \xi$.

Примеры документов MATHCAD вычисления объемов на основе изложенного подхода.

1. Шар вписанный в единичный куб

[Оглавление](#)

$$0 \leq x, y, z \leq 1$$

ORIGIN := 0 Сфера в единичном кубе R := 0.5

Исходные функции: $f(x, y, z) := (x - R)^2 + (y - R)^2 + (z - R)^2 - R^2$

Вспомогательные функции $ab(x, y, z) := x$ $cd(x, y, z) := y$ $Ef(x, y, z) := z$

Начальные "затравочные" значения $x := 0.5$ $y := 0.5$ $z := 0.5$

Поиск границ области:

Given $f(x, y, z) \leq 0$ $p := \text{Minimize}(ab, x, y, z)$ $a := ab(p_0, p_1, p_2)$ $a = -0.0006$

Given $f(x, y, z) \leq 0$ $p := \text{Maximize}(ab, x, y, z)$ $b := ab(p_0, p_1, p_2)$ $b = 1.0006$

Given $f(x, y, z) \leq 0$ $p := \text{Minimize}(cd, x, y, z)$ $c := cd(p_0, p_1, p_2)$ $c = -9.452699 \times 10^{-7}$

Given $f(x, y, z) \leq 0$ $p := \text{Maximize}(cd, x, y, z)$ $d := cd(p_0, p_1, p_2)$ $d = 1.000186$

Given $f(x, y, z) \leq 0$ $p := \text{Minimize}(Ef, x, y, z)$ $e := Ef(p_0, p_1, p_2)$ $e = -0.0006$

Given $f(x, y, z) \leq 0$ $p := \text{Maximize}(Ef, x, y, z)$ $ff := Ef(p_0, p_1, p_2)$ $ff = 1.0006$

Функция, вычисляющая количество точек, попавших в область и их координаты:

```
p(n) :=
| n1 ← -1
| x ← runif(n, a, b)
| y ← runif(n, c, d)
| z ← runif(n, e, ff)
| for K ∈ 0..n - 1
|   if 1 = (f(xK, yK, zK) ≤ 0)
|     | n1 ← n1 + 1
|     | tn1,0 ← xK
|     | tn1,1 ← yK
|     | tn1,2 ← zK
|   rez ← ( t
|           n1 )
| rez
```

$n := 40000$ $s := p(n)$
 прора := s₁ прора = 20898 количество точек, попавших в область

V_{kub} := 1 Объем единичного куба

$V := V_{kub} \cdot \frac{\text{прора}}{n}$ $V = 0.52245$ Вычисленный Объем тела

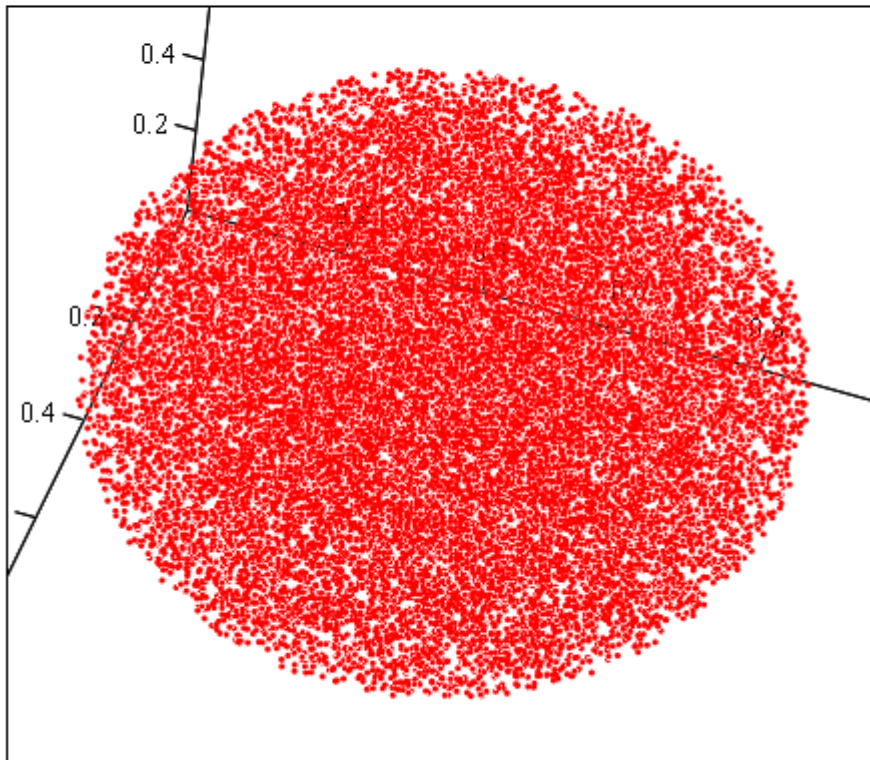
$VS := \frac{4}{3} \cdot \pi \cdot R^3$ $VS = 0.523599$ Объем тела Аналитическая формула

	0	1	2
0	0.073503	0.673144	0.654561
1	0.646865	0.289756	0.186019
2	0.218316	0.524511	0.425811
3	0.544084	0.086067	0.507489
4	0.625901	0.355692	0.944494
5	0.362645	0.525801	0.226666
6	0.535873	0.596999	0.815997
7	0.636509	0.563184	0.924174
8	0.165921	0.308764	0.668593
9	0.599152	0.347301	0.864188
10	0.242327	0.428723	0.842658
11	0.15672	0.242012	0.384796
12	0.34646	0.565491	0.119435
13	0.206989	0.419512	0.450113
14	0.10336	0.603748	0.783952
15	0.376564	0.4434	0.444198
$s_0 =$ 16	0.711496	0.676972	0.364008
17	0.871321	0.181933	0.576567
18	0.888468	0.429759	0.29948
19	0.526829	0.491257	0.681082
20	0.254779	0.71452	0.514615
21	0.288351	0.189362	0.662566
22	0.378428	0.309679	0.203476
23	0.55295	0.601344	0.630756
24	0.615452	0.673228	0.81624
25	0.426893	0.360165	0.802903
26	0.386181	0.396944	0.28607
27	0.850623	0.246105	0.364675
28	0.590738	0.181772	0.172162
29	0.569661	0.521375	0.080201
30	0.60199	0.237822	0.2203
31	0.524144	0.138462	0.370034
32	0.56918	0.230401	0.531074
33	0.933211	0.602964	...

координаты точек,
попавших в область

Облако точек попавших в область

$$\alpha := \left[(s_0)^{\langle 0 \rangle} \right] \quad \beta := (s_0)^{\langle 1 \rangle} \quad \gamma := (s_0)^{\langle 2 \rangle}$$



(α, β, γ)

2. Куб, заданный соотношением:

$$\frac{1}{4} \leq x, y, z \leq \frac{3}{4}$$

ORIGIN := 0

Исходные функции: $f(x, y, z) := \begin{cases} 1 & \text{if } \left[\left(\frac{1}{4} < x < \frac{3}{4} \right) \cdot \left(\frac{1}{4} < y < \frac{3}{4} \right) \cdot \left(\frac{1}{4} < z < \frac{3}{4} \right) \right] \\ 0 & \text{otherwise} \end{cases}$

Вспомогательные функции $ab(x, y, z) := x$ cd(x, y, z) := y $Ef(x, y, z) := z$

Начальные "затравочные" значения $x := 0.5$ $y := 0.5$ $z := 0.5$

Поиск границ области:

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Minimize}(ab, x, y, z)$ $a := ab(p_0, p_1, p_2)$ $a = 0.25$

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Maximize}(ab, x, y, z)$ $b := ab(p_0, p_1, p_2)$ $b = 0.75$

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Minimize}(cd, x, y, z)$ cd := $cd(p_0, p_1, p_2)$ $c = 0.25$

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Maximize}(cd, x, y, z)$ $d := cd(p_0, p_1, p_2)$ $d = 0.75$

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Minimize}(Ef, x, y, z)$ ef := $Ef(p_0, p_1, p_2)$ $e = 0.25$ +

Given $\left(\frac{1}{4} < x < \frac{3}{4} \right)$ $\left(\frac{1}{4} < y < \frac{3}{4} \right)$ $\left(\frac{1}{4} < z < \frac{3}{4} \right)$
 $p := \text{Maximize}(Ef, x, y, z)$ $ff := Ef(p_0, p_1, p_2)$ $ff = 0.75$

Функция, вычисляющая количество точек, попавших в область и их координаты:

```

p(n) :=
  n1 ← -1
  x ← runif(n, a, b)
  y ← runif(n, c, d)
  z ← runif(n, e, ff)
  for K ∈ 0..n - 1
    if 1 = (f(xK, yK, zK))
      n1 ← n1 + 1
      tn1,0 ← xK
      tn1,1 ← yK
      tn1,2 ← zK
  rez ← ( t
          n1 )
rez

```

n := 4000 $\overset{\text{рез}}{s_1} := p(n)$

$\text{прора} := s_1$	$\text{прора} = 3999$	Количество точек, попавших в область
$V_{\text{kub}} := (b - a) \cdot (d - c) \cdot (ff - e)$		Объем области, ограничивающей тело
$\overset{\text{рез}}{V} := V_{\text{kub}} \cdot \frac{\text{прора}}{n}$	$V = 0.124969$	Вычисленный Объем тела
$VS := 0.5^3$	$VS = 0.125$	Объем тела Аналитическая формула

$$s_0 =$$

	0	1	2
0	0.466912	0.604513	0.528072
1	0.396056	0.339194	0.582495
2	0.500088	0.393635	0.527546
3	0.442994	0.694075	0.45994
4	0.459228	0.66354	0.665135
5	0.61091	0.514767	0.570946
6	0.654719	0.4378	0.380795
7	0.299265	0.744556	0.730664
8	0.635673	0.284378	0.714426
9	0.53253	0.702638	0.631158
10	0.36503	0.443895	0.387402
11	0.661125	0.363301	0.698696
12	0.653427	0.466537	0.642621
13	0.391576	0.71409	0.279991
14	0.55503	0.561054	...

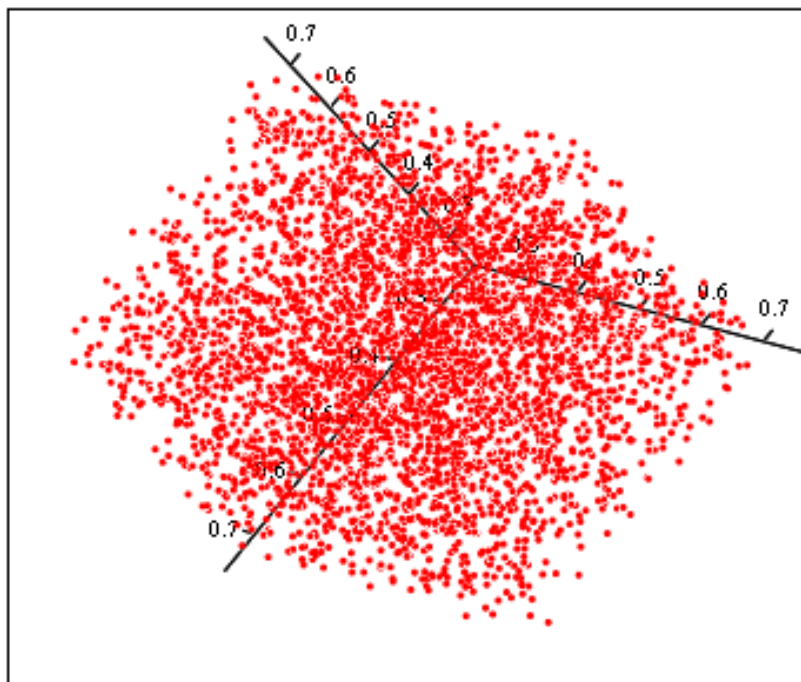
координаты точек,
попавших в область

Облако точек, попавших в область

$$\alpha := \left[(s_0)^{\langle 0 \rangle} \right]$$

$$\beta := (s_0)^{\langle 1 \rangle}$$

$$\gamma := (s_0)^{\langle 2 \rangle}$$



(α, β, γ)

3. Геометрическое множество точек, удовлетворяющих неравенствам:

$$\begin{cases} x^2 \cdot \sin(10 \cdot \pi \cdot y) \geq \sqrt{z} \\ x \cdot y \cdot z \geq 0,001 \\ 0 \leq x, y, z \leq 1 \end{cases}$$

ORIGIN := 0

Исходные функции:

$$f(x, y, z) := \begin{cases} 1 & \text{if } \left[(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y)) \cdot (0.001 < x \cdot y \cdot z) \cdot (0 < x < 1) \cdot (0 < y < 1) \cdot (0 < z < 1) \right] \\ 0 & \text{otherwise} \end{cases}$$

Вспомогательные функции $ab(x, y, z) := x$ $cd(x, y, z) := y$ $Ef(x, y, z) := z$

Начальные "затравочные" значения $x := 0.5$ $y := 0.5$ $z := 0.5$

Поиск границ области:

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Minimize}(ab, x, y, z)$ $a := ab(p_0, p_1, p_2)$ $a = 0.331027$

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Maximize}(ab, x, y, z)$ $b := ab(p_0, p_1, p_2)$ $b = 1.000001$

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Minimize}(cd, x, y, z)$ $c := cd(p_0, p_1, p_2)$ $c = 0.20224$

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Maximize}(cd, x, y, z)$ $d := cd(p_0, p_1, p_2)$ $d = 0.499035$

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Minimize}(Ef, x, y, z)$ $e := Ef(p_0, p_1, p_2)$ $e = 0.002005$

Given $(\sqrt{z} < x^2 \cdot \sin(10\pi \cdot y))$ $(0.001 < x \cdot y \cdot z)$ $(0 < x < 1)$ $(0 < y < 1)$ $(0 < z < 1)$
 $p := \text{Maximize}(Ef, x, y, z)$ $ff := Ef(p_0, p_1, p_2)$ $ff = 1.000691$

Функция, вычисляющая количество точек, попавших в область и их координаты:

```

p(n) :=
  n1 ← -1
  x ← runif(n, a, b)
  y ← runif(n, c, d)
  z ← runif(n, e, ff)
  for K ∈ 0..n - 1
    if 1 = (f(xK, yK, zK))
      n1 ← n1 + 1
      tn1,0 ← xK
      tn1,1 ← yK
      tn1,2 ← zK
  rez ← ( t
         n1 )
  rez

```

n := 40000 $\overset{\text{рез}}{s} := p(n)$

прора := s₁ прора = 3931 Количество точек, попавших в область

V_{kub} := (b - a) · (d - c) · (ff - e) Объем области, ограничивающей тело

$\overset{\text{рез}}{V} := V_{\text{kub}} \cdot \frac{\text{прора}}{n}$ V = 0.019487 Вычисленный Объем тела

	0	1	2
0	0.82312	0.221336	0.037957
1	0.900846	0.227805	0.221148
2	0.916684	0.258414	0.301825
3	0.790273	0.434849	0.173765
4	0.720251	0.248573	0.164093
5	0.651052	0.258922	0.105258
6	0.875669	0.21774	0.107915
7	0.755838	0.46454	0.222678
8	0.91721	0.425903	0.183298
9	0.753274	0.262163	0.255046
10	0.946985	0.243347	0.363633
11	0.948197	0.480502	0.175054
12	0.638449	0.268262	0.046192
13	0.908441	0.458829	0.311144
14	0.977915	0.222505	...

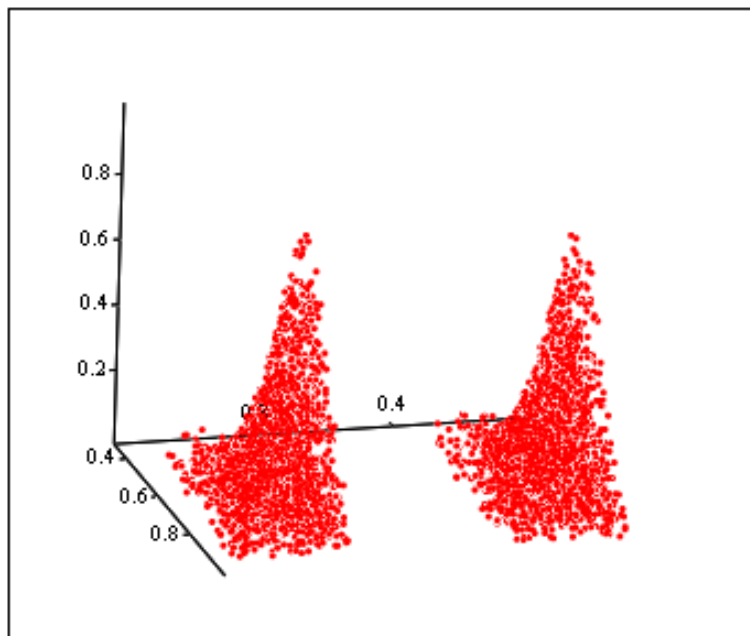
координаты точек,
попавших в область

Облако точек, попавших в область

$$\alpha := \left[(s_0)^{\langle 0 \rangle} \right]$$

$$\beta := (s_0)^{\langle 1 \rangle}$$

$$\gamma := (s_0)^{\langle 2 \rangle}$$



(α, β, γ)

Использование случайных блужданий для решения граничных задач теории потенциалов. Метод Судзуо Какутани

Рассмотрим следующий процесс. Из заданной точки в случайном направлении выполнен шаг заданной величины. Из вновь полученной точки ещё раз в случайном направлении выполняется шаг такой же величины. Такой процесс называется случайным блужданием. Он хорошо описывает человека после праздника. Физическое явление – броуновское движение пылинки.

Оказывается, можно доказать следующий очевидный факт: при неограниченном увеличении количества шагов расстояние от начальной точки до текущей пропорционально величине $A \cdot \sqrt{n}$, где A – константа. Это означает, что любая сколь угодно удаленная граница будет достигнута при увеличении числа блужданий.

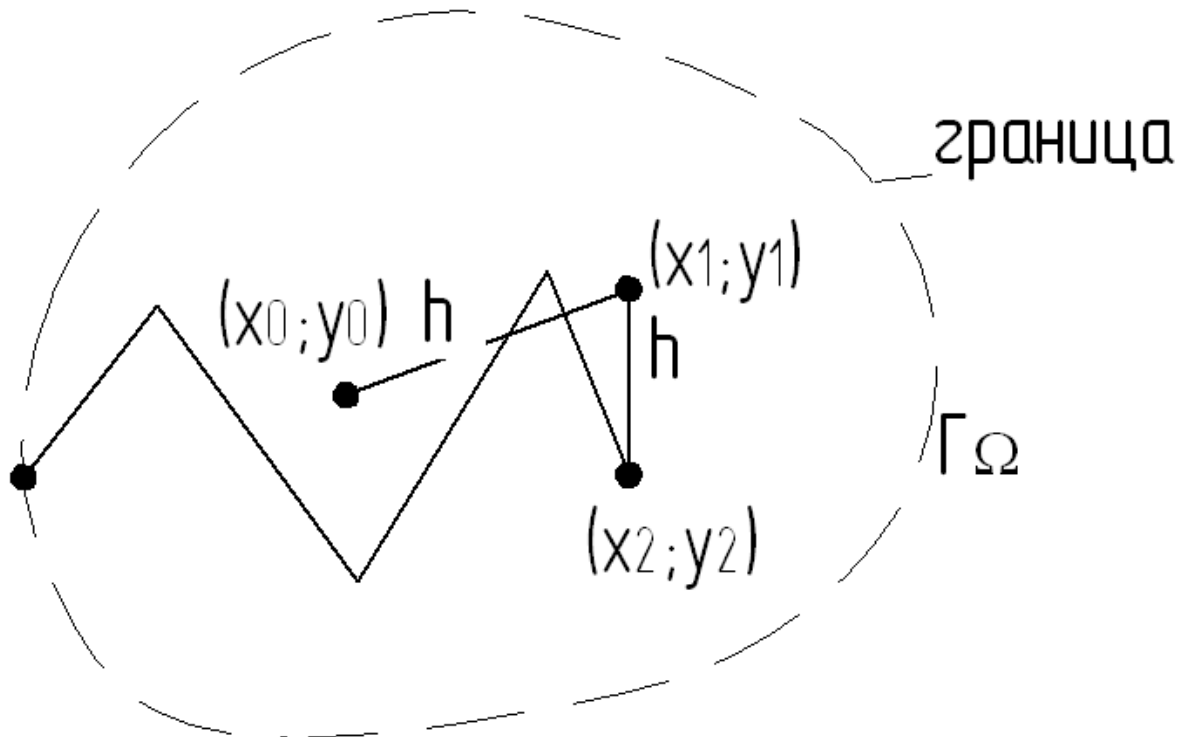


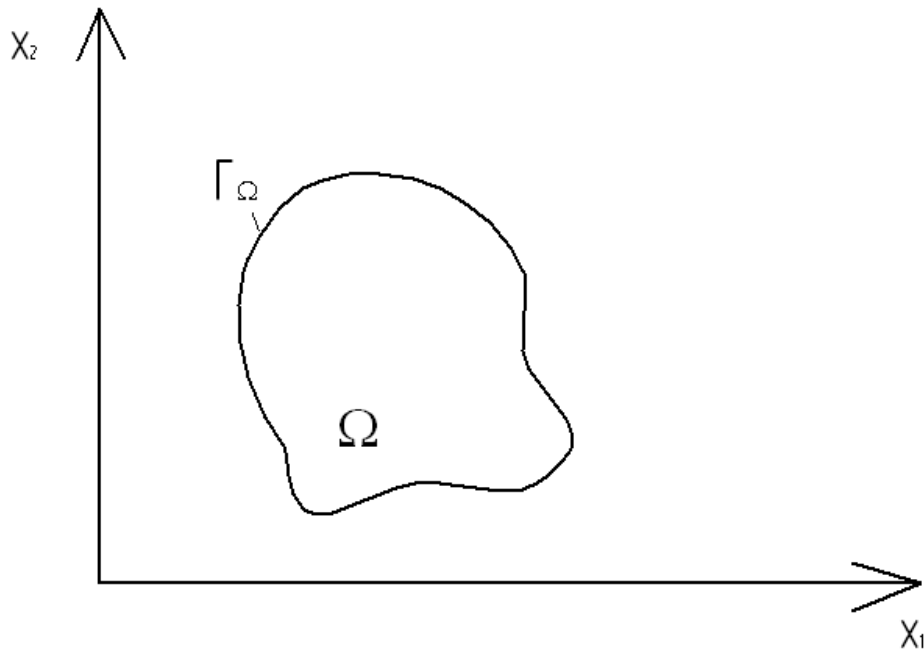
Рисунок 6.6. Иллюстрация случайных блужданий.

Японский ученый Судзуо Какутани предложил использовать этот факт для решения граничных задач теории потенциала.

Под граничной задачей понимается следующее: нужно найти значение некоторой функции, которая внутри области Ω удовлетворяет дифференциальному уравнению (ДУ), чаще всего в частных производных, вида

$$B \cdot u = r, \quad (6.1)$$

где B – дифференциальный оператор, $u = u(x_1, x_2, \dots, x_k)$ – неизвестная функция, $r = r(x_1, x_2, \dots, x_k)$ – заданная функция).

Рисунок 6.7. Область Ω в двумерном случае

На границе Γ_Ω области Ω задаются известные значения

$$C \cdot u = q \quad (6.2)$$

где C – другой дифференциальный оператор, $q = q(x_1, x_2, \dots, x_k)$ – заданная функция.

Пример.

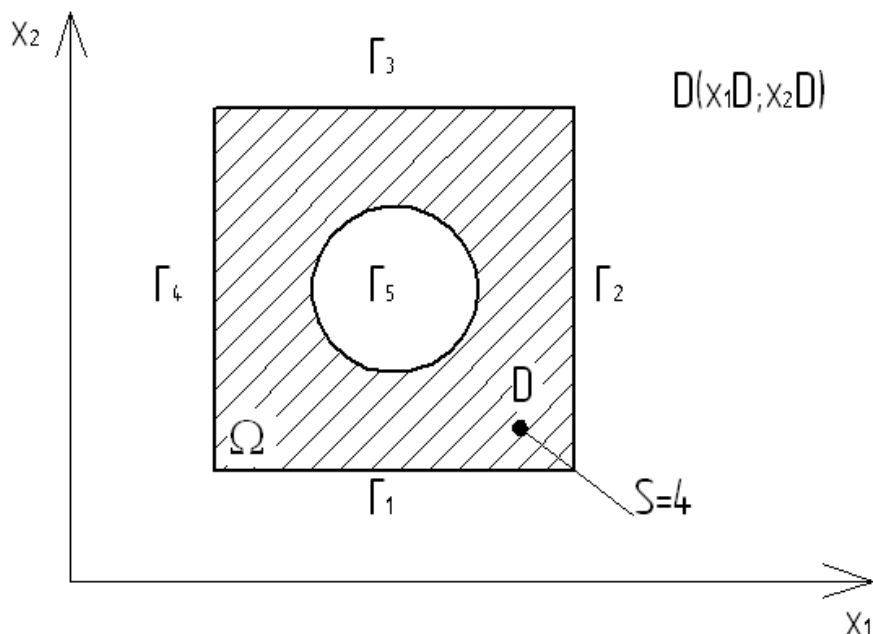


Рисунок 6.8. Распределение температуры в данной области.

Рассмотрим задачу о распределении температур внутри этой области. Пусть у нас на внешней границе происходит теплообмен с окружающей средой. На внутренней границе Γ_5 известно значение температуры в каждой точке, и в точке

D расположен источник тепла с постоянной интенсивностью равной 4, k – коэффициент теплопроводности материала.

Можно показать, что распределение температуры $u = u(x_1, x_2)$ внутри области Ω описывается следующим ДУ:

$$k \cdot \frac{\partial^2 u}{\partial x_1^2} + k \cdot \frac{\partial^2 u}{\partial x_2^2} - f(x_1, x_2) = 0 \quad (в \Omega) \quad (6.3)$$

$$где f(x_1, x_2) = \begin{cases} 0, & \text{если } x_1 \neq x_D, x_2 \neq x_D \\ 4, & \text{если } x_1 = x_D, x_2 = x_D \end{cases}$$

Тогда дифференциальный оператор B из уравнения (6.1) в этом случае имеет вид:

$$B = k \cdot \frac{\partial^2 u}{\partial x_1^2} + k \cdot \frac{\partial^2 u}{\partial x_2^2}$$

$$r = f(x_1, x_2)$$

Граничные условия, описывающие теплообмен на границе области, словами можно сформулировать так: поток тепла по направлению нормали к поверхности пропорционален разности между температурой точки поверхности и температурой окружающей среды.

Пусть температура окружающей среды $u_{\text{возд}}$.

На $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ можно записать:

$$\frac{\partial u}{\partial \vec{n}} = H(u - u_{\text{возд}}) \quad (6.4)$$

Тогда дифференциальный оператор C из соотношения (6.2) в данном случае применительно к (6.4):

$$C = \frac{\partial}{\partial x_1} n_{x_1} + \frac{\partial}{\partial x_2} n_{x_2} - H$$

$$\vec{n} = (n_{x_1}, n_{x_2})$$

$$q = -H \cdot u_{\text{возд}}$$

На границе Γ_5 :

$$u = u_{\text{жидк}}$$

Здесь дифференциальный оператор имеет вид $C=1$

$$q = u_{\text{жидк}}$$

Граничная задача теории потенциалов

Оказывается, что различные по своей физической сущности процессы описываются одинаковыми по виду ДУ в частных производных:

1. Задача о чистом кручении стержня произвольного сечения
2. Задача о распределении тепла
3. Задача об уровне жидкости в пористом материале
4. Задача о распределении электрических потенциалов и т.д.

Эти задачи описываются уравнениями вида

$$L \cdot u = f$$

$$L = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \dots + \frac{\partial^2}{\partial x_k^2}$$

Оказывается, что решения таких уравнений обладают следующими свойствами: Решение в точке $(x_{10}, x_{20}, \dots, x_{k0})$ будет средним из значений этой функции на произвольной границе Γ_ξ , окружающей эту точку

$$u(x_0, y_0) \approx \frac{1}{m} \sum_{i=1}^m u(x_i^*, y_i^*) \quad (6.5)$$

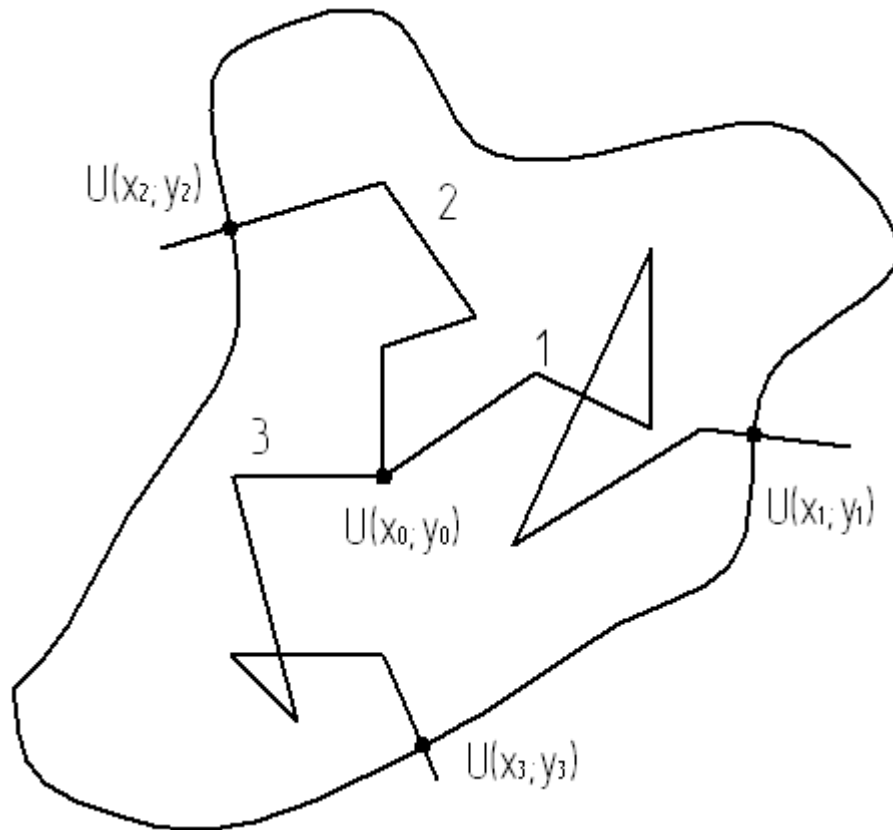


Рисунок 6.9. Иллюстрация метода Какутани.

Таким образом выполняется отдельное случайное блуждание. Оно прекращается, когда достигнута граница области. Значение функции на границе запоминается. Выполняется m блужданий и вычисляется усредненное значение.

Физический смысл такого процесса понятен: если точка начала блужданий расположена ближе к какой-то части границы, то большее количество блужданий будут заканчиваться на этой части граница, а значит, что и в сумме (6.5) большее количество слагаемых будет отражать температуру этой части.

Фрагмент документа MATHCAD для реализации случайных блужданий, соответствующих следующему рисунку, представлен ниже.

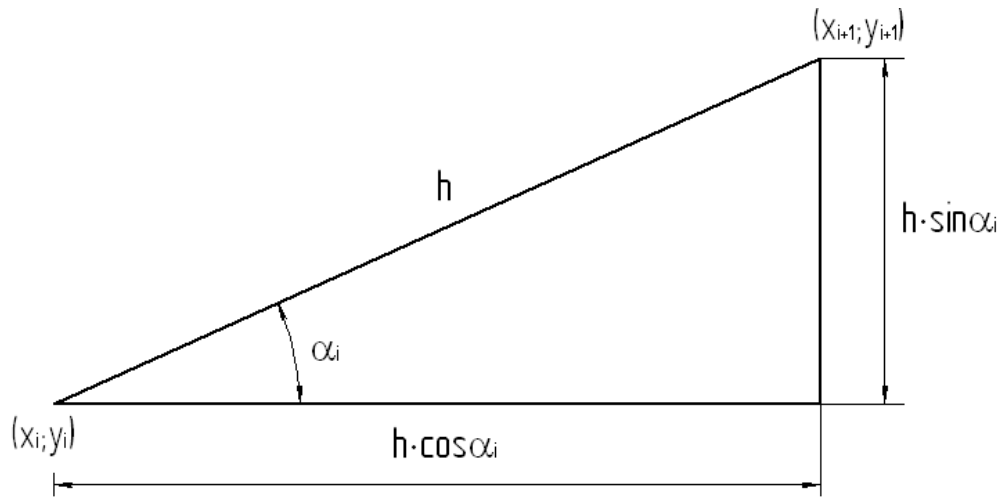


Рисунок 6.10. Шаг случайного блуждания с шагом h .
Для двумерного случая фрагмент документа MATHCAD:

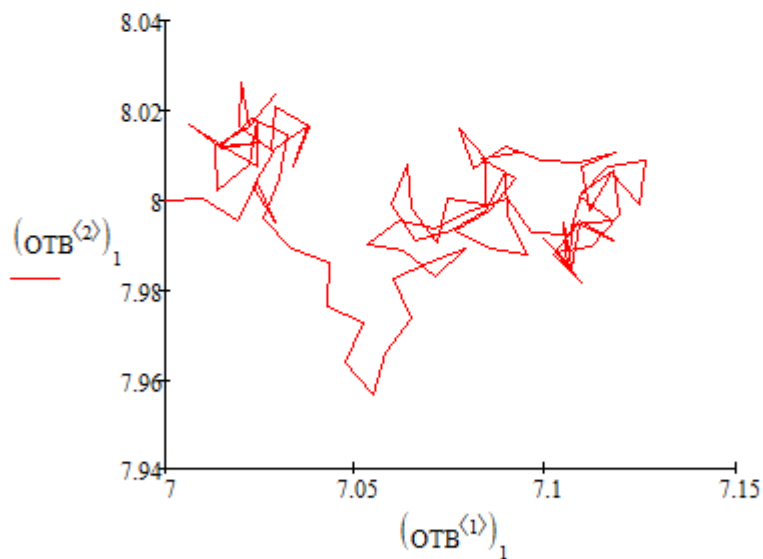
$$\begin{aligned} x_{i+1} &= x_i + h \cdot \cos(\alpha_i) & h &= \text{const} \\ y_{i+1} &= y_i + h \cdot \sin(\alpha_i) & \alpha_i &\in [0; 2\pi] \end{aligned}$$

ORIGIN := 1

```
SLB(x0,y0,h,N,xx) := | Seed(xx)
                    | alfa ← runif(N,0,2·π)
                    | x1 ← x0
                    | y1 ← y0
                    | for i ∈ 2..N
                    |   | xi ← xi-1 + h·cos(alfai-1)
                    |   | yi ← yi-1 + h·sin(alfai-1)
                    | rez ← (x y)
```

x0 := 7 y0 := 8 xx := 6 N := 100 h := 0.01

OTB := SLB(x0,y0,h,N,xx) OTB = ({100,1} {100,1})



$$\left(\text{OTB}^{(1)} \right)_1^T =$$

	1	2	3	4	5	6	7
1	7	7.01	7.019	7.024	7.029	7.023	...

$$\left(\text{OTB}^{(2)} \right)_1^T =$$

	1	2	3	4	5	6	7
1	8	8	7.995	8.004	7.995	8.003	...

Для трехмерного случая фрагмент документа MATHCAD:

```

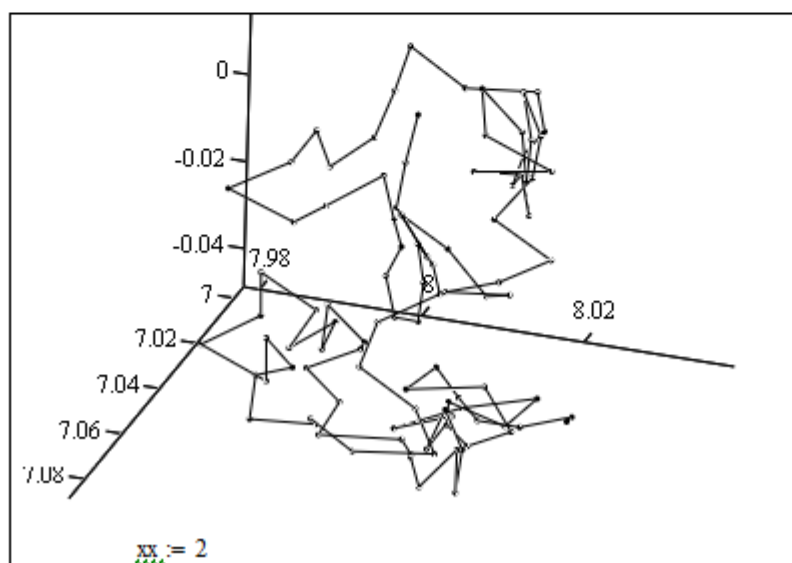
SLB(x0,y0,z0,h,N,xx) := | Seed(xx)
                        | alfa ← runif(N,0,2·π)
                        | tt ← runif(N, -π/2, π/2)
                        | x1 ← x0
                        | y1 ← y0
                        | z1 ← z0
                        | for i ∈ 2..N
                        |   | xi ← xi-1 + h·cos(alfai-1)·cos(tti-1)
                        |   | yi ← yi-1 + h·sin(alfai-1)·cos(tti-1)
                        |   | zi ← zi-1 + h·sin(tti-1)
                        | rez ← (x y z)

```

z0 := 0 x0 := 7 y0 := 8 xx := 6 N := 100 h := 0.01

OTB := SLB(x0,y0,z0,h,N,xx)

$\alpha := (\text{OTB}^{(1)})_1$ $\beta := (\text{OTB}^{(2)})_1$ $\gamma := (\text{OTB}^{(3)})_1$



(α, β, γ)

Укрупненный алгоритм метода Какутани. Решение граничной задачи теории потенциалов

1. Выбрать случайное направление.
2. Выполнить шаг фиксированной величины в этом направлении.

3. Есть ли пересечение с контуром границы на выполненном шаге?

Если да, то перейти к шагу 6.

Если нет, то перейти к шагу 1.

4. Запомнить значение искомой функции в точке пересечения с границей, исходя из граничных условий.

5. Найти новое среднее значение искомой функции в точке начала блужданий по формуле (6.5).

6. Отличается ли новое среднее от старого больше, чем на ξ ?

Если да, то перейти к шагу 7.

Если нет, то вычисления прекращаем.

7. Переходим в точку начала блужданий и возвращаемся к шагу 1.

В этом алгоритме предполагается, что у пользователя имеется процедура, возвращающая ему точку пересечения двух геометрических объектов: отрезка, соответствующего очередному шагу блужданий, и кривой, описывающей нужный участок границы.

Тема 7. Метод наихудшего случая и статистический анализ

Понятие о параметрах математической модели

Рассмотрим пример объекта проектирования (рисунок 7.1.).

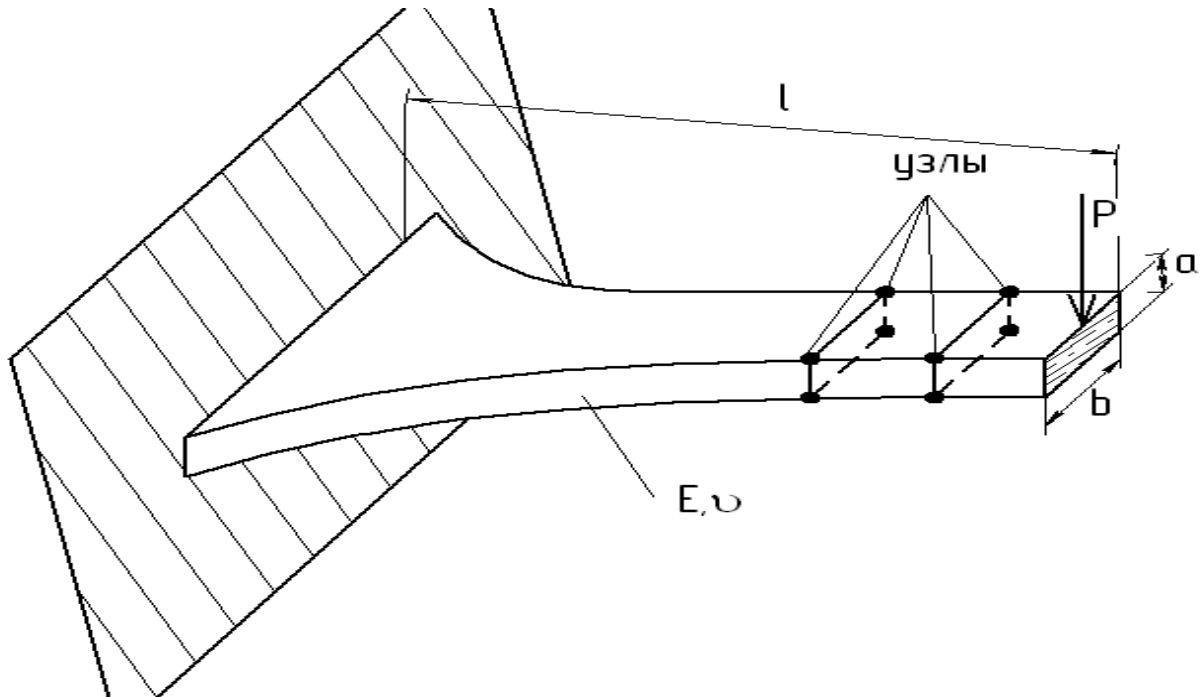


Рисунок 7.1. Консольная балка.

В результате расчета исследователя интересует u_{max} - перемещения правого конца и σ_{max} - максимальные напряжения в материале балки. Расчеты выполняются либо с использованием аналитических формул, либо на основании какого-либо численного метода.

Тогда формально можно считать, что имеет место связь:

$$\bar{Y} = \vec{F}(\bar{X}, \bar{Q}), \quad (7.1)$$

где

\bar{X} - вектор внутренних параметров модели;

\bar{Q} - вектор внешних параметров модели;

Y - вектор выходных параметров;

$\vec{F}(\bar{X}, \bar{Q})$ - связь между \bar{X}, \bar{Q} и Y .

Соотношение (1) мы и будем называть математической моделью объекта.

Например, в методе конечных элементов решение подобной задачи сводится к решению системы ОДУ второго порядка:

$$K \cdot \bar{U} + C \cdot \dot{\bar{U}} + M \cdot \ddot{\bar{U}} = \vec{P}(t)$$

$\vec{U} = \bar{U}(t)$ - вектор узловых перемещений.

На основании полученных узловых перемещений по формуле можно вычислить σ_{max} , а y_{max} будет совпадать с перемещением одного из узлов. При моделировании всегда система или объект мысленно отделяются от окружающей среды. Сама же внешняя среда заменяется воздействиями на этот объект. Эти воздействия задаются в виде сил, перемещений, температур и т.д., на которые исследователь повлиять не может.

Понятно, что такое разделение на объект и внешнюю среду носит условный характер.

Применительно к рассматриваемому примеру, вектор \bar{X} имеет следующие компоненты:

$$\vec{X} = \begin{pmatrix} E \\ \nu \\ a \\ b \\ L \\ \dots \end{pmatrix}$$

Здесь E- модуль Юнга материала, ν - коэффициент Пуассона материала, a,b,L- геометрические размеры.

Как видно, компоненты этого вектора могут иметь разную физическую природу.

$\vec{Q} = \begin{pmatrix} p(t) \\ t^0 \\ \dots \end{pmatrix}$ - перемещения заделки, температуры и т.д.

$\vec{Y} = \begin{pmatrix} y_{max} \\ \sigma_{max} 0 \end{pmatrix}$ - это вектор, состоящий из компонент, которые мы хотим наблюдать.

$\vec{F}(\bar{X}, \bar{Q})$ - модель.

Понятие о работоспособности технического объекта.

Под работоспособностью будем понимать выполнение ограничений на выходные параметры.

Эти ограничения обычно задаются в виде односторонних или двусторонних неравенств, где $y_{i_{доп}}$ - допустимые заданные величины.

Например

$$i = \overline{1, n} \quad y_i \leq y_{i_{доп}}$$

$u_{i\text{доп}}$ задают исходя из технических требований, сформулированных заказчиком.

Метод наихудшего случая при анализе работоспособности технического объекта.

При изготовлении технического объекта в силу особенностей технологического процесса могут возникнуть отличия в значениях предполагаемых и фактических внутренних параметров.

Предполагаемые значения называются **номинальными**.

$$x_{j\text{ном}}, \text{ где } j = \overline{1, m}$$

Значения внешних параметров тоже могут отличаться от номинальных.

$$q_{k\text{ном}}, \text{ где } k = \overline{1, l}$$

Из этого следует, что и значения выходных параметров, получаемых при реальной эксплуатации изделия могут отличаться от значений, соответствующих номинальным входным параметрам.

Возникает вопрос: если условие работоспособности выполняется для u_i номинального, то выполнится ли оно для реального u_i ? Из этого следует, что однократного анализа недостаточно для ответа на вопрос о работоспособности технического объекта.

Одним из способов более обоснованного ответа является метод наихудшего случая. Идея этого метода предполагает, что известны отклонения для каждого из внутренних параметров от их номинальных значений (рисунок 7.2).

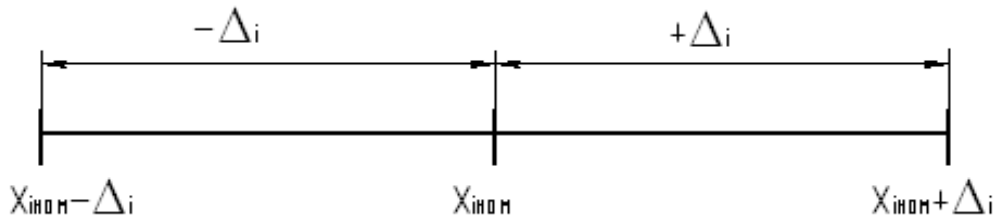


Рисунок 7.2. Отклонения для каждого из внутренних параметров

Аналогично и для внешних параметров.

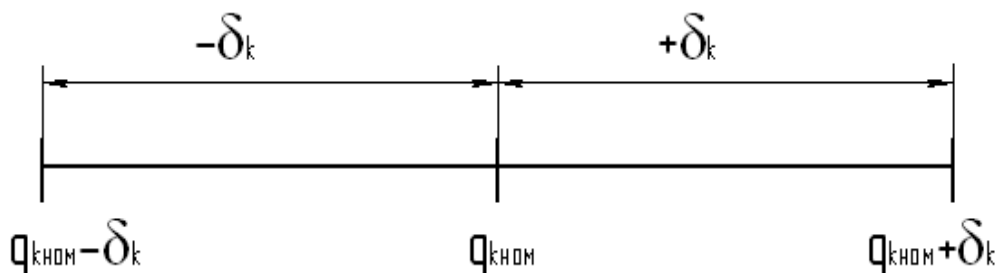


Рисунок 7.3. Отклонения для каждого из внешних параметров

Для параметра x_1 выберем теперь значение на том из концов интервала (значений $x_{1\text{номинальное}} - \Delta_i$ и $x_{1\text{номинальное}} + \Delta_i$), которое обеспечивает **приближение** выходного параметра y к ограничениям (по сравнению с номинальным значением) и назовём его $x_{1\text{наихудшее}}$; зафиксируем его и повторим то же для второй, третьей и так далее компоненты - вектора \bar{X} внутренних параметров модели.

Затем выполним все то же самое для всех компонент вектора \bar{Q} внешних параметров модели.

Таким образом мы получим вектор $\bar{X}_{\text{наихудшее}}$ и $\bar{Q}_{\text{наихудшее}}$.

Если окажется, что компоненты вектора

$$\bar{Y}_{\text{наихудшее}} = \vec{F}(\bar{X}_{\text{наихудшее}}, \bar{Q}_{\text{наихудшее}}),$$

еще удовлетворяют условиям работоспособности, то технический объект признается работоспособным **в смысле метода наихудшего случая**.

Понятно, что такой подход приведет к **излишнему завышению требований для технического объекта**, т.к. возникновение ситуации, когда одновременно «все стало плохо» (одновременно стали плохими все компоненты) едва ли возможно.

Это приведет к завышению массы и стоимости изделия, однако этот подход все же используют при проектировании объектов, функционирование которых может привести к гибели людей или экологической катастрофе, связывая это с безопасностью.

Статистический анализ работоспособности

Более обоснованным подходом для оценки работоспособности технического объекта является вероятностная оценка выходных параметров. При этом можно поступать примерно так:

- изучить поведение x_i (например, с помощью наблюдений) и подобрать соответствующий закон распределения, описывающий x_i как случайную величину $i = \overline{1, m}$.
- выполнить отдельный анализ на основе математической модели, при этом:
 - а. сформировать x_i с помощью выбранного распределения
 - б. выбрать компоненты q_i исходя из требований наихудшего случая

Таких экспериментов выполнить p штук. При этом для компоненты y_k получится p значений.

$$y_k^{(1)}, y_k^{(2)}, \dots, y_k^{(p)}$$

Остается выполнить осреднение и принять

$$\tilde{y}_k \approx \frac{1}{p} \sum_{s=1}^p y_k^{(s)} \quad (7.2)$$

Эту величину теперь можно использовать для проверки условия работоспособности. Возникает вопрос о количестве экспериментов, которое гарантирует достоверность результатов. Для ответа на этот вопрос поступают так. Кроме величины \tilde{y}_k вычисляют такую величину:

$$\sigma_k^2 = \frac{1}{p-1} \sum_{s=1}^p \left(y_k^{(s)} - \tilde{y}_k \right)^2 \quad (7.3)$$

Это несмещенная оценка дисперсии.

Если теперь вычислить величину

$$\xi_j = \frac{1}{\sqrt{p}} \cdot t_\gamma \cdot \sigma_k, \quad (7.4)$$

p – количество экспериментов,

t_γ - величина, определяемая из соотношения:

$$\begin{aligned} \gamma &= \Phi(t_\gamma), \\ \Phi(x) &= \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt, \end{aligned}$$

γ - задаваемая достоверность.

то можно гарантировать, что

$$|\tilde{y}_k - M_k| \leq \xi$$

и \tilde{y}_k отличается от ожидаемого не больше, чем на ξ .

Связь между γ и t_γ можно найти либо по таблице, либо по встроенной процедуре MATHCAD. Чаще всего используются следующие значения:

$$\gamma = 0,95 \rightarrow t_\gamma = 1,96$$

$$\gamma = 0,945 \rightarrow t_\gamma = 2$$

Тогда формула (3) примет вид для $\gamma = 0,945$

$$p=50 \quad \xi_M = 0,28\sigma_K$$

$$p=100 \quad \xi_M = 0,196\sigma_K$$

$$p=1000 \quad \xi_M = 0,062\sigma_K$$

Понятие о чувствительности математической модели

Пусть задана математическая модель:

$$\bar{Y} = F(\bar{X}, \bar{Q}).$$

Анализ чувствительности заключается в определении матрицы чувствительности A , элементами которой являются абсолютные коэффициенты влияния (коэффициенты чувствительности), вычисляемые так:

$$A_{ji} = \frac{dy_j}{dx_i}, \quad (7.5)$$

где y_j - j -ый выходной, x_i - i -ый внутренний (или внешний параметры).

$$A = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \dots & \frac{\partial y_m}{\partial x_n} \end{pmatrix} \quad (7.6)$$

$$\vec{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{pmatrix} \quad \vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

Через коэффициенты матрицы чувствительности иногда определяют относительный коэффициент влияния:

$$B_{ji} = A_{ji} \cdot \frac{x_{iном}}{y_{jном}}, \quad (7.7)$$

Из (7.5) видно, что A_{ji} представляет собой скорость изменения j -того выходного параметра по отношению к изменению i -того внутреннего параметра.

Заметим, что выбор значения внутреннего параметра метода наихудшего случая, рассмотренного ранее, при наличии одного выходного параметра может быть представлен следующими аналитическими выражениями:

$$x_{ин.с.} = x_{i_ном} - sign(A_i) \cdot \Delta x_{ипред}$$

в том случае, когда условие работоспособности сформулировано в виде

$$y \geq y_{доп.}$$

Или в виде

$$x_{ин.с.} = x_{i_ном} + sign(A_i) \cdot \Delta x_{ипред}$$

в том случае, когда условие работоспособности сформулировано в виде

$$y \leq y_{доп.}$$

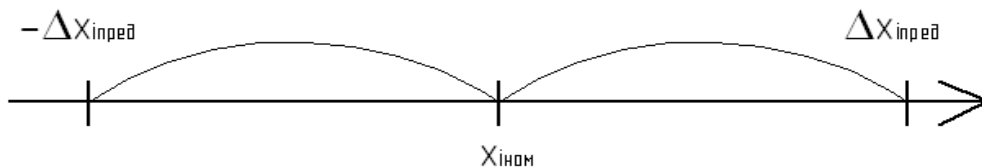


Рисунок 7.4. Графическая иллюстрация условия работоспособности где

$$sign(A_i) = \begin{cases} 1, \text{ если } A_i > 0 \\ 0, \text{ если } A_i = 0 \\ -1, \text{ если } A_i < 0 \end{cases}$$

То же касается и выбора внешних параметров.

Анализ чувствительности имеет смысл выполнять, например, при использовании оптимизации в том случае, когда однократное обращение к математической модели требует большего времени расчета.

Пошаговый метод анализа чувствительности (метод приращений)

Запишем разложение функции $y_j(\vec{X})$ в ряд Тейлора в окрестности аргумента $\vec{X} = \vec{X}_{\text{НОМ}}$

$$y_j(\vec{X}) = y_j(\vec{X}_{\text{НОМ}}) + \left(\frac{\partial y_j}{\partial \vec{X}}, \Delta \vec{X} \right) + \frac{1}{2} (\Delta \vec{X})^T \frac{\partial^2 y_j}{\partial \vec{X}^2} \cdot \Delta \vec{X} + \dots$$

Если удерживать в этой формуле только линейные слагаемые, то получим

$$y_j \approx y_{j\text{НОМ}} + \left(\frac{\partial y_j}{\partial \vec{X}}, \Delta \vec{X} \right)$$

Выражая отсюда коэффициент:

$$A_{ji} \approx \frac{y_j - y_{j\text{НОМ}}}{\Delta x_i} \quad (7.8)$$

Пусть

$$\begin{aligned} \Delta x_1 = \Delta x_2 = \dots = \Delta x_{i-1} &= 0 \\ \Delta x_i &\neq 0 \\ \Delta x_{i+1} = \Delta x_{i+2} = \dots = \Delta x_n &= 0 \end{aligned}$$

Тогда

$$y_j \approx y_{j\text{НОМ}} + A_{ji} \cdot \Delta x_i$$

Таким образом для получения коэффициента A_{ji} методом приращений можно поступить так:

- выполнить вычисление j -того выходного параметра при номинальных значениях внутренних параметров;
- i -тому внутреннему параметру дать приращение Δx_i , а по другим параметрам приращение запретить $\Delta x_k = 0$ ($k \neq i$);
- Вычислить y_j для такого вектора внутренних параметров.

По формуле (7.5) вычислить коэффициент чувствительности.

- При необходимости повторить эти шаги для остальных внутренних параметров.

Таким образом, если внутренних параметров n , то для вычисления коэффициента A_{ji} при фиксированном j потребуется $n+1$ обращений к математической модели.

Иногда для повышения точности используют следующую формулу:

$$A_{ji} \approx \frac{y_j^+ - y_j^-}{2 \cdot \Delta x_i} \quad (7.9)$$

y_j^+ - вычисляется на положительных приращениях Δx_i .

y_j^- - вычисляется на отрицательных приращениях Δx_i .

Формула (7.9) соответствует аппроксимации производной центральной разностью и имеет более высокий порядок точности, чем формула (7.8).

За повышение точности нужно заплатить $2n$ обращениями к математической модели в формуле (7.9).

Понятно, что в силу выполненной линеаризации, метод приращений может быть «хорошо» использован для медленно меняющихся функций. Это его главный недостаток.

К достоинствам относятся простота и универсальность.

Понятие о параметрах математической модели

Рассмотрим пример объекта проектирования (рисунок 5.1.).

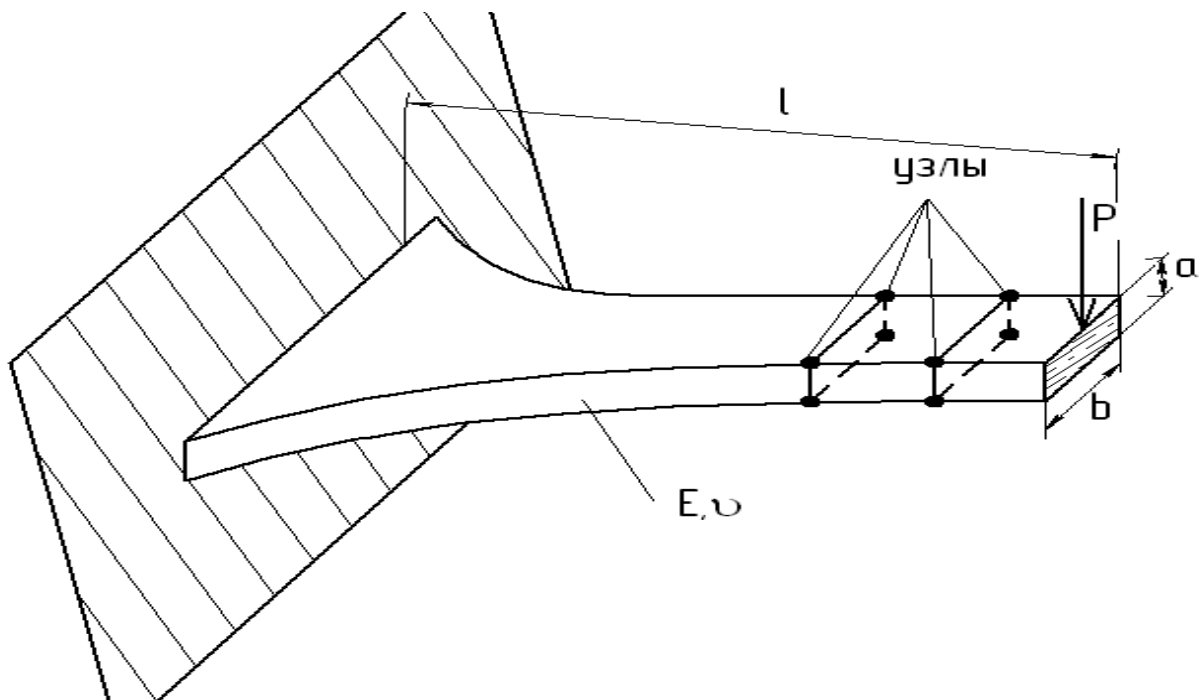


Рисунок 5.1. Консольная балка.

В результате расчета исследователя интересует y_{max} - перемещения правого конца и σ_{max} - максимальные напряжения в материале балки.

Расчеты выполняются либо с использованием аналитических формул, либо на основании какого-либо численного метода.

Тогда формально можно считать, что имеет место связь:

$$\bar{Y} = \vec{F}(\bar{X}, \bar{Q}), \quad (5.1)$$

где

[Оглавление](#)

\bar{X} - вектор внутренних параметров модели;

\bar{Q} - вектор внешних параметров модели;

Y – вектор выходных параметров;

$\vec{F}(\bar{X}, \bar{Q})$ - связь между \bar{X}, \bar{Q} и Y .

Соотношение (1) мы и будем называть математической моделью объекта.

Например, в методе конечных элементов решение подобной задачи сводится к решению системы ОДУ второго порядка:

$$K \cdot \bar{U} + C \cdot \dot{\bar{U}} + M \cdot \ddot{\bar{U}} = \vec{P}(t)$$

$\vec{U} = \bar{U}(t)$ - вектор узловых перемещений.

На основании полученных узловых перемещений по формуле можно вычислить σ_{max} , а y_{max} будет совпадать с перемещением одного из узлов.

При моделировании всегда система или объект мысленно отделяются от окружающей среды. Сама же внешняя среда заменяется воздействиями на этот объект. Эти воздействия задаются в виде сил, перемещений, температур и т.д., на которые исследователь повлиять не может.

Понятно, что такое разделение на объект и внешнюю среду носит условный характер.

Применительно к рассматриваемому примеру, вектор \bar{X} имеет следующие компоненты:

$$\vec{X} = \begin{pmatrix} E \\ \nu \\ a \\ b \\ L \\ \dots \end{pmatrix}$$

Здесь E - модуль Юнга материала, ν - коэффициент Пуассона материала, a, b, L - геометрические размеры.

Как видно, компоненты этого вектора могут иметь разную физическую природу.

$\vec{Q} = \begin{pmatrix} p(t) \\ t^0 \\ \dots \end{pmatrix}$ - перемещения заделки, температуры и т.д.

$\vec{Y} = \begin{pmatrix} y_{max} \\ \sigma_{max}() \end{pmatrix}$ - это вектор, состоящий из компонент, которые мы хотим наблюдать.

$\vec{F}(\bar{X}, \bar{Q})$ - модель.

Понятие о работоспособности технического объекта.

Под работоспособностью будем понимать выполнение ограничений на выходные параметры.

Эти ограничения обычно задаются в виде односторонних или двусторонних неравенств, где $y_{i_{\text{доп}}}$ - допустимые заданные величины.

Например

$$i = \overline{1, n} \quad y_i \leq y_{i_{\text{доп}}}$$

$y_{i_{\text{доп}}}$ задают исходя из технических требований, сформулированных заказчиком.

Тема 8. Моделирование конкурирующих стратегий

Пример конкурирующих стратегий

Поясним идею моделирования конкурирующих стратегий на примере задачи о выборе стратегии эксплуатации автобуса.

Пусть автобус с регулярным маршрутом должен выполнить в день N -рейсов. При этом каждый день он начинает работу в хорошем состоянии. Если рейс выполняется в хорошем состоянии, то существует вероятность a того, что он закончит рейс в ухудшенном состоянии A (небольшая поломка, не влияющая на возможность выполнить очередной рейс). На устранение этой поломки для восстановления автобуса до исходного состояния требуется время, за которое автобус пропустит один рейс. Стратегию эксплуатации автобуса, когда возникшая небольшая поломка устраняется сразу же после рейса, в котором она возникла, назовем « α ».

Если же не проводить этот ремонт, и продолжать рейсы в ухудшенном состоянии, то существует вероятность b критической поломки автобуса (назовем такое состояние B), когда рейс будет прерван, и потребуются незамедлительный ремонт. При этом больше ни одного рейса до конца дня нельзя будет выполнить. Назовем эту стратегию эксплуатации автобуса « β ». Эти две стратегии эксплуатации автобуса представлены в схемах на рисунке 6.1.

Вопрос заключается в том, какая из этих стратегий окажется лучшей при заданных вероятностях a , b и заданном числе предполагаемых рейсов N в день. В этом смысле стратегии конкурируют между собой.

В изложенной постановке такая задача была решена Морзом, а после него Крайзенем. В этой постановке были получены оценки среднего числа рейсов в день для стратегий « α » и « β » соответственно:

$$M_{\alpha} = \frac{N}{1+a} + a \frac{1-(1-a)^N}{(1+a)^2},$$

$$M_{\beta} = \frac{a^2(1-(1-b)^N) - b^2(1-(1-b)^N)}{ab(a-b)}.$$

Здесь

a – вероятность перехода в фазу А в данном рейсе;

b – вероятность перехода в фазу В, если автобус выполняет рейс в фазе А;

N – число запланированных рейсов в день.

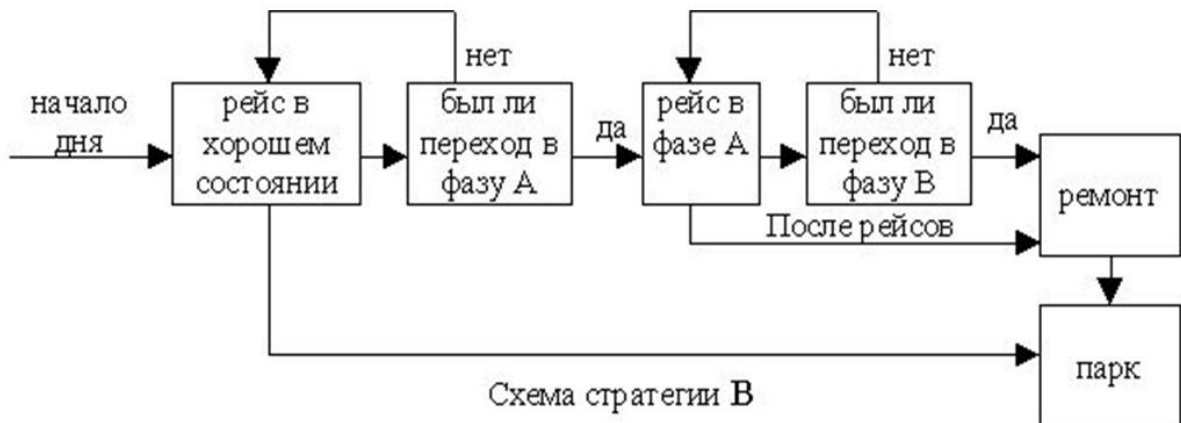
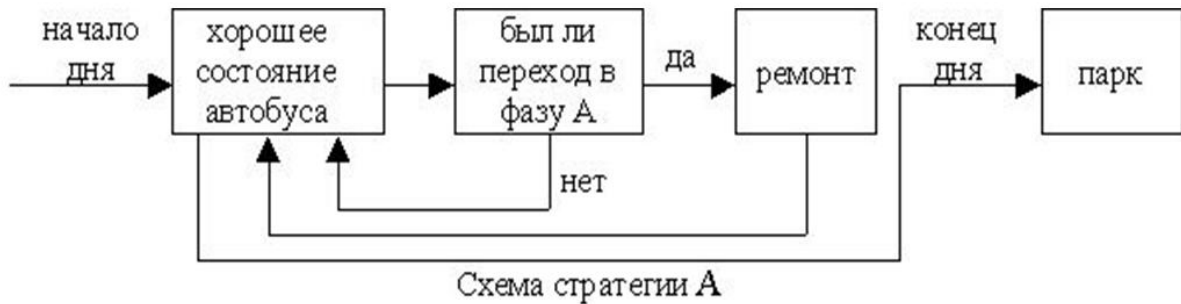


Рисунок 6.1 Схемы стратегий « α » и « β ».

Построение алгоритмической модели конкурирующих стратегий

Получение представленных выше аналитических выражений даже при незначительном усложнении стратегий крайне затруднительно. В то же время, построение алгоритмической модели затруднений не вызывает, и позволяет легко модифицировать модель при изменении формулировки задачи.

На рисунке 6.2 представлена схема алгоритма для вычисления среднего числа рейсов в день при стратегиях « α » и « β ».

Пояснения к схеме:

SG1 – флаг, который обозначает хорошее или ухудшенное состояние при стратегии « α »;

SG1=1 – автобус находится в хорошем состоянии,

SG1=0 – автобус находится в ухудшенном состоянии;

SG2 – флаг, который обозначает хорошее или ухудшенное состояние при стратегии « β »;

SG2=1 – автобус находится в хорошем состоянии,

SG2=0 – автобус находится в ухудшенном состоянии;

SB2 – флаг, который обозначает фазу В при стратегии « β »;

SB2=1 – автобус находится в фазе В при стратегии « β »,

SB2=0 – в иных случаях;

D_{max} – число дней, которые мы хотим моделировать;

D – номер моделируемого дня;

RA , RB – количество действительно выполненных рейсов в течение рассматриваемого дня при стратегиях « α » и « β » соответственно;

SRA , SRB – общее число выполненных рейсов за D рассмотренных дней при стратегиях « α » и « β » соответственно;

$AVERA$, $AVERB$ – среднее число выполненных рейсов за день при стратегиях « α » и « β » соответственно;

$Z = RA - RB$ - разность между количеством рейсов при стратегиях « α » и « β » соответственно;

$SZ = \sum_D Z$ - общая сумма разностей между количеством рейсов при стратегиях « α » и « β » соответственно;

$AVE = AVERA - AVERB$ - разность средних значений между количеством рейсов за день при стратегиях « α » и « β » соответственно;

ЗАМЕЧАНИЯ:

Автобус начинает рейс в хорошем состоянии.

Для получения в среде MathCAD значения случайной величины r с равномерным законом распределения на интервале (c,d) используем встроенную функцию $\text{runif}(k,c,d)$ (где в нашем случае $c=0,d=1$). Описание функции можно найти в системе помощи. Здесь k – количество значений.

Если r попадает на интервал $[0, a]$, то можно считать, что с вероятностью a рассматриваемое событие происходит.

При тестировании программы вычислений на основе данного алгоритма можно использовать следующие крайние ситуации:

а) Если $a=0$, то при стратегии « α » общее число выполненных рейсов должно быть равным $SRA=N \cdot D_{max}$, и SRB тоже будет таким же.

б) Если $a=1$ и $b=1$, то при стратегии « β » за один день будет выполнен 1 рейс, а общее число выполненных рейсов за все дни будет D_{max} .

с) Если $a=1$ и $b=0$, то при стратегии « β »

$$SRB = D_{max} \cdot N, \text{ при стратегии } \alpha \text{ } SRA = D_{max} \cdot \left(\frac{N}{2}\right).$$

Можно, например, рассмотреть и стратегию « γ », отличие которой от стратегии « β » заключается в том, что автобус закончивший день в фазе А не ремонтируется.

В данном методическом пособии стратегия « γ » подробно не рассматривается.

Попробуйте самостоятельно произвести изменения в схеме алгоритма так, чтобы она соответствовала стратегии « γ ».

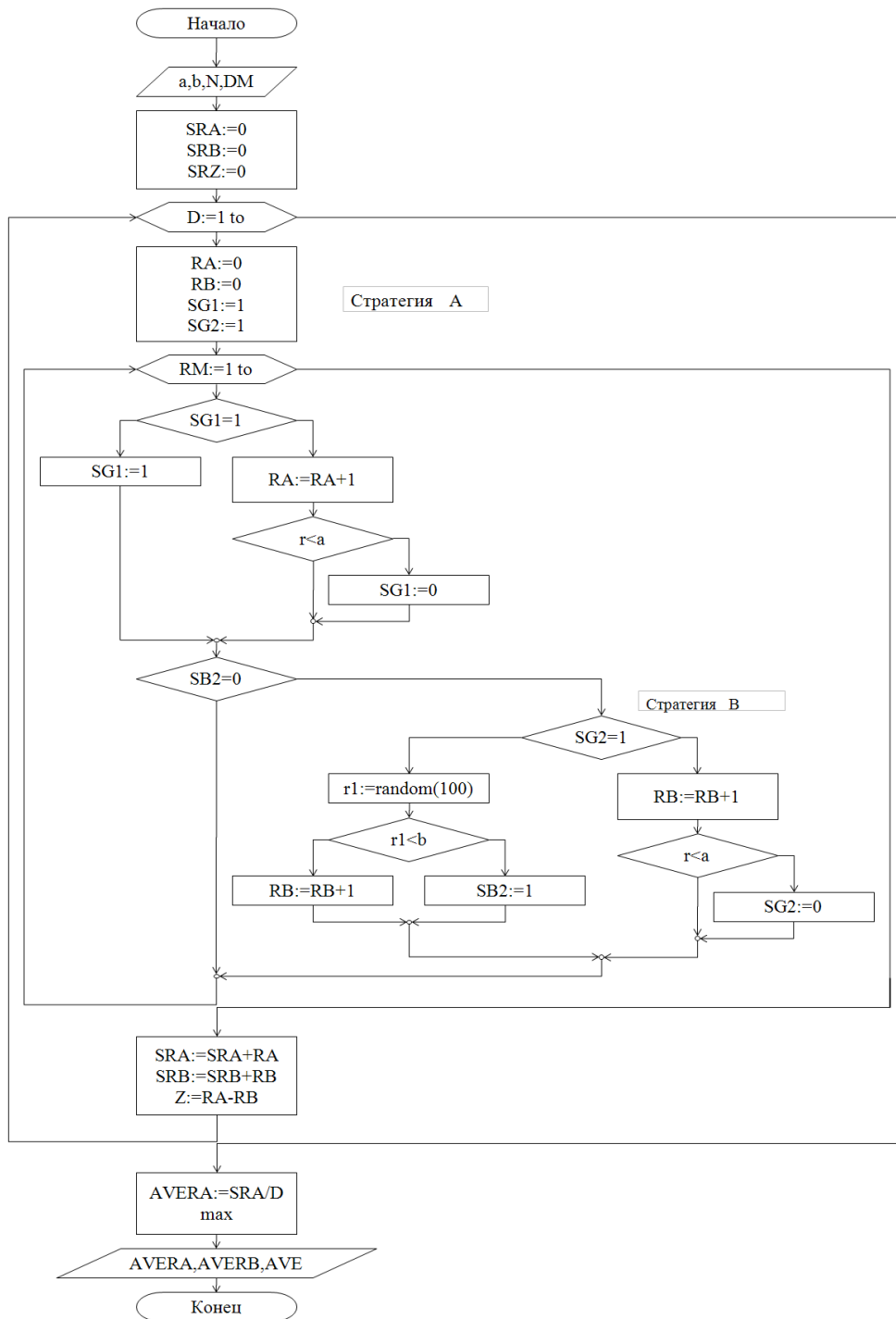


Рисунок 6.2 Схема алгоритма моделирования конкурирующих стратегий

Программная реализация в среде MATHCAD алгоритмической модели конкурирующих стратегий

ORIGIN:- 1

Процедура вычисления числа рейсов при стратегиях α и β , а также разницы в рейсах при данных стратегиях

```

Func(a,b,N,Dmax) :-
  D ← 0      Текущий номер моделируемого дня
  SRA ← 0    Общее число рейсов за D
  SRB ← 0    дней при стратегиях  $\alpha$  и  $\beta$ 
  SZ ← 0     Разница в числе рейсов в стратегиях  $\alpha$  и  $\beta$ 
  SG2 ← 1    Флаг обозначает хорошее состояние стратегии  $\beta$ .
  SB2 ← 0    НЕ критическое состояние при стратегии  $\beta$ 
  while D < Dmax
    RA ← 0    Количество действительно выполненных рейсов
    RB ← 0    в течении дня при стратегиях  $\alpha$  и  $\beta$ 
    RM ← 0    соответственно.
    SG1 ← 1
    if SB2 = 1 ПОЛНЫЙ ремонт ПРОПУСК всех рейсов дня стратегии  $\beta$ .
      SB2 ← 0  НЕ критическое состояние при стратегии  $\beta$ 
      SG2 ← 1  Хорошее состояние стратегии  $\beta$ .
    while RM < N
      RM ← RM + 1
      if SG1 = 1 Хорошее состояние при стратегии  $\alpha$ 
        RA ← RA + 1 Счётчик рейсов в день при стратегии  $\alpha$ 
        R $\alpha$  ← runif(1,0,1)
        SG1 ← 0 if R $\alpha$ 1 < a      Есть переход в ухудшенное состояние
                               при стратегии  $\alpha$ .
        SG1 ← 1 otherwise
      if SB2 = 0 НЕ критическое состояние при стратегии  $\beta$ 
        R $\beta$  ← runif(1,0,1)
        if SG2 = 1 Хорошее состояние стратегии  $\beta$ .
          SG2 ← 0 if R $\alpha$ 1 < a      Есть переход в ухудшенное состояние
                               при стратегии  $\beta$ .
          RB ← RB + 1 Счётчик рейсов в день при стратегии  $\beta$ 
        otherwise
          SB2 ← 1 if R $\beta$ 1 < b      Переход в критическое
                               состояние при стратегии  $\beta$ 
          RB ← RB + 1 otherwise     Рейс выполняется, если не было
                               перехода в критическое состояние при
                               стратегии  $\beta$ 
      SRA ← SRA + RA
      SRB ← SRB + RB
    -----
    Z ← RA - RB
    SZ ← SZ + Z
    D ← D + 1      Увеличиваем номер дня
  AVERA ←  $\frac{SRA}{Dmax}$ 
  AVERB ←  $\frac{SRB}{Dmax}$  Среднее число рейсов в день при стратегиях  $\alpha$ 
                    и  $\beta$  соответственно.
  AVERZ ←  $\frac{SZ}{Dmax}$  Средняя разница в рейсов передач
                    между стратегиями  $\alpha$  и  $\beta$  за один день.
  return (AVERA AVERB AVERZ)

```

Полученные результаты тестирования

Пример при вероятности перехода в ухудшенное и совсем плохое состояние =1

$$\text{Func}(1,1,100,4) = (50 \ 1 \ 49)$$

Пример при вероятности перехода в ухудшенное =1 и совсем плохое состояние =0

$$\text{Func}(1,0,100,4) = (50 \ 100 \ -50)$$

Пример при вероятности перехода в ухудшенное =0 и совсем плохое

$$\text{Func}(0,1,100,4) = (100 \ 100 \ 0)$$

Пример при вероятности перехода в ухудшенное =0 и совсем плохое

$$\text{Func}(0,0,100,4) = (100 \ 100 \ 0)$$

Эта задача может быть переформулирована и применительно к другим ситуациям (к станку, к компьютеру и другим объектам).

Рассмотрим реализацию различных законов распределения в среде MathCAD.

Различные законы распределения случайных величин Геометрическое распределение

Случайные числа получаются из случайных чисел r_i , распределенных равномерно на $[0,1]$, на основе соотношения

$$x_i = \left[\frac{\ln r_i}{\ln(1 - \rho)} \right],$$

где $[..]$ – целая часть числа.

Экспоненциальное распределение

Моделирующая формула имеет вид:

$$x_i = -\frac{1}{\lambda} \ln r_i, \quad r_i \in R[0,1].$$

Распределение Вейбулла

Моделирующая формула имеет вид:

$$x_i = \left(-\frac{1}{\lambda} \ln r_i \right)^{1/\alpha}, \quad r_i \in R[0,1].$$

Распределение Парето

Моделирующая формула имеет вид:

$$x_i = x_0 \left(\frac{1}{r_i} \right)^{1/\alpha}, r_i \in R[0,1].$$

Здесь x_0 – параметр положения (левая граница области возможных значений).

Распределение Эрланга

Моделирующая формула имеет вид:

$$\ln\left(\prod_{i=0}^{\alpha} r_{j+i*\alpha}\right), r_i \in R[0,1].$$

Распределение Коши

Моделирующая формула имеет вид:

$$x_i = \lambda \cdot \tan\left[\pi \cdot \left(r_i - \frac{1}{2}\right)\right] + \mu, r_i \in R[0,1].$$

Логистическое распределение

Моделирующая формула имеет вид:

$$x_i = \mu - \lambda \ln \frac{1 - r_i}{r_i}, r_i \in R[0,1].$$

Нормальное распределение

Моделирующая формула имеет вид:

$$x_i = m_x + \sigma_x \left(\sum_{i=1}^{12} r_i - 6 \right), r_i \in R[0,1].$$

Здесь r_i – значение случайной величины, равномерно распределенной на интервале $[0,1]$. Эти значения в MathCAD можно получить с помощью встроенной функции `runif`.

Все греческие буквы – параметры соответствующих распределений.

Пример программной реализации закона распределения Коши в среде MATHCAD

Процедура формирования массива случайных величин x , распределённых по закону Коши в MathCAD.

$$\text{KORASP}(\lambda, \mu, n) := \left| \begin{array}{l} r \leftarrow \text{runif}(n, 0, 1) \\ \text{for } i \in 1..n-1 \\ \quad x_i \leftarrow \lambda \cdot \tan \left[\pi \cdot \left(r_i - \frac{1}{2} \right) \right] + \mu \\ x \end{array} \right.$$

Процедура вычисления количества случайных величин из заданного массива x , попадающих на каждый из подинтервалов длиной

$$\Delta = \frac{x_{\text{end}} - x_{\text{beg}}}{N}$$

из интервала

$$[x_{\text{beg}}, x_{\text{end}}]$$

может выглядеть так:

```

Raspr(x,x_beg,x_end,N) :=
| Δ ←  $\frac{x\_end - x\_beg}{N}$ 
| curX ← x_beg
| j ← 0
| while curX < x_end
|   | a ← 0
|   | for i ∈ 0..length(x) - 1
|   |   a ← 1 + a if curX ≤ x1 < curX + Δ
|   | freqj,1 ← curX +  $\frac{\Delta}{2}$ 
|   | freqj,0 ← a
|   | j ← j + 1
|   | curX ← curX + Δ
| freq

```

Обращение к этим процедурам (n – количество моделируемых случайных величин):

$\lambda := 1$ $\mu := 2$ $n := 5000$

$X := \text{KORASP}(\lambda, \mu, n)$ $S := \text{Raspr}(X, 0, 6, 50)$

Обращение к встроенной в MathCAD процедуре gcauchy формирования массива случайных величин, распределённых по закону Коши:

$X1 := \text{gcauchy}(n, \mu, \lambda)$ $S1 := \text{Raspr}(X1, 0, 6, 50)$

Графическое представление результатов для написанной и встроенной процедур в виде гистограммы:

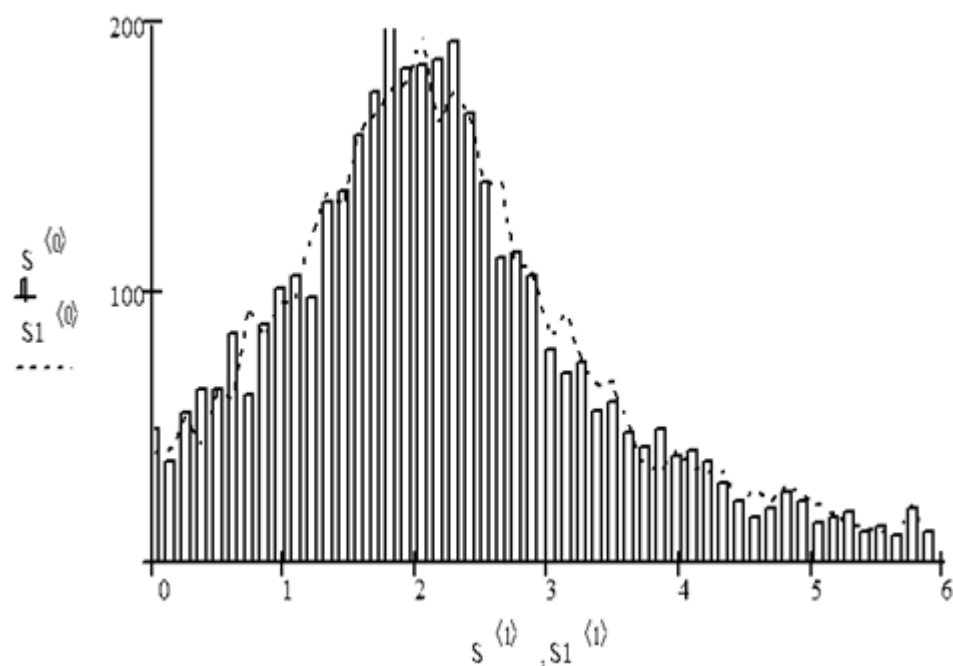


Рисунок 6.3 Гистограмма

Процедура вычисления массива накопленных частот:

$$\text{NAK_castot}(f) := \left| \begin{array}{l}
 n \leftarrow \sum_{j=0}^{\text{length}(f)-1} f_j \\
 \text{for } k \in 0.. \text{length}(f) - 1 \\
 \quad \sum_{j=0}^k f_j \\
 \text{REZ}_k \leftarrow \frac{\sum_{j=0}^k f_j}{n} \\
 \text{REZ}
 \end{array} \right.$$

Вычисление значений функции распределения:

$$\text{FS} := \text{NAK_castot}(S^{(0)})$$

График функции распределения:

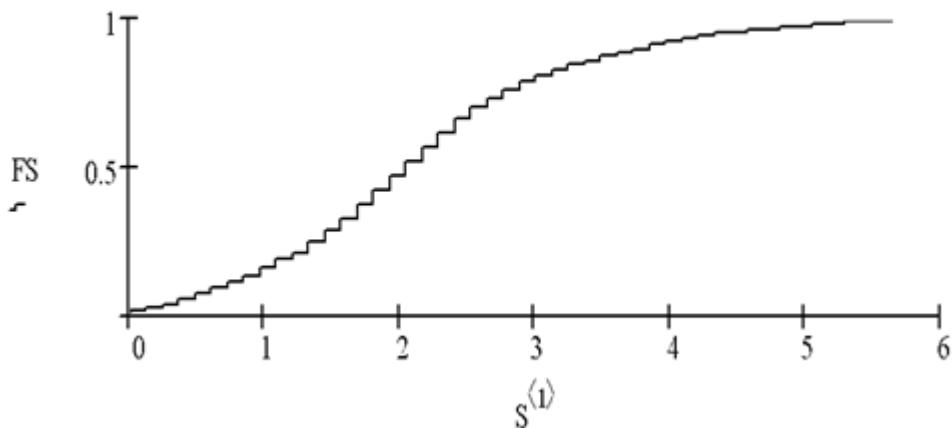


Рисунок 6.4 График функции распределения

Генерация двух векторов XX и YY , компоненты которых есть значения случайной величины с распределением по закону Коши.

$\lambda:=1$ $\mu:=2$ $n:=100$ $XX:=\text{KORASP}(\lambda,\mu,n)$ $YY:=\text{KORASP}(\lambda,\mu,n)$

Отображение точек с координатами (XX_i, YY_i) на координатной плоскости.

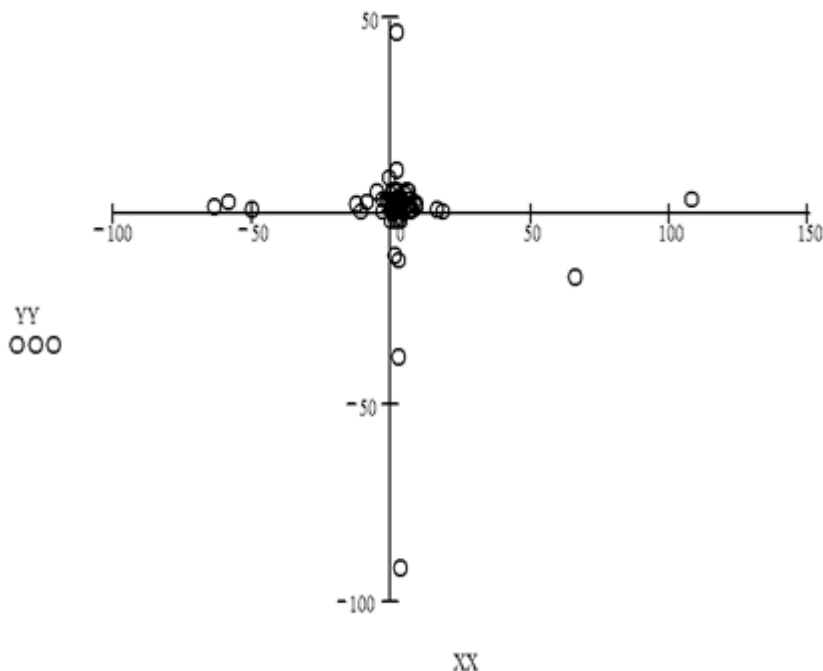


Рисунок 6.5 Значения случайной величины с распределением по закону Коши в двумерном случае

Отображение этих же точек с координатами (XX_i, YY_i) , попадающих в прямоугольник с координатами левого нижнего угла $(-2;-2)$ и правого верхнего угла $(6;6)$, на координатной плоскости.

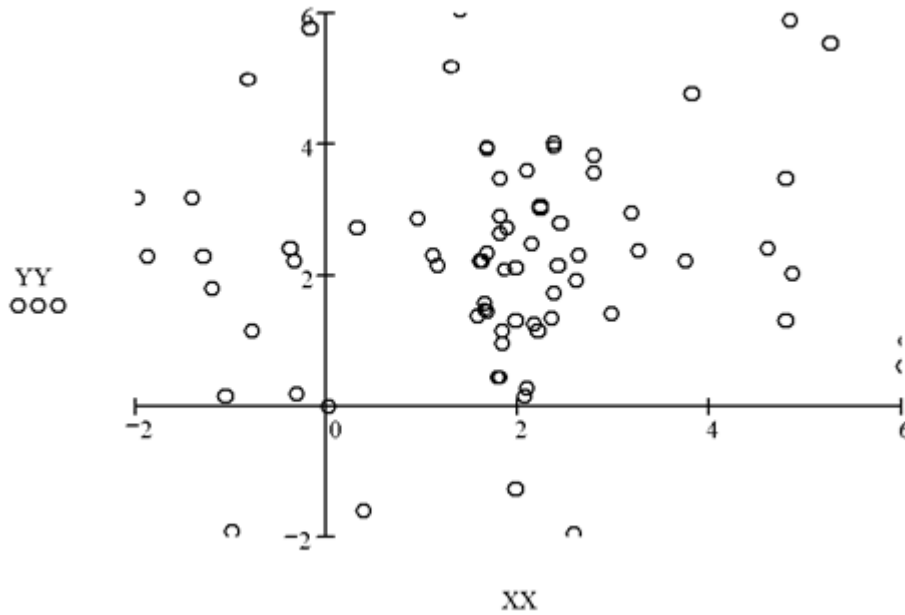


Рисунок 6.6 Значения случайной величины с распределением по закону. Коши.
Увеличение

Задание по работе

Наземное устройство наблюдения в дневное время суток передаёт порциями информацию пролетающему над ним спутнику. Спутник пролетает над объектом N раз в дневное время суток. За каждый пролёт передаётся одна порция информации.

В передающем устройстве в течение одного сеанса связи может возникнуть несущественная неисправность с вероятностью a . При этом передатчик будет продолжать функционировать. Однако, в этой ситуации при следующем сеансе с вероятностью b может возникнуть критическая неисправность, после которой передатчик не сможет функционировать. При этом ремонт передатчика займёт столько времени, что все сеансы связи до конца дня, включая текущий, будут отменены. Если же выполнять ремонт передатчика после завершения сеанса, в ходе которого возникла несущественная неисправность, то будет пропущен только следующий сеанс.

Возникает вопрос: какая из двух стратегий эксплуатации предпочтительней в этом смысле, что за M дней эксплуатации при N планируемых сеансах связи в день, количество фактически завершённых сеансов будет наибольшим?

Рассмотрим две стратегии эксплуатации передающего устройства. Первая стратегия (назовем её « α ») заключается в том, что ремонт производится после несущественной поломки, а вторая стратегия (назовём её « β ») заключается в том, что ремонт производится только после

критической поломки. Поскольку из двух стратегий нужно выбрать наилучшую, то они называются конкурирующими.

В работе требуется с использованием среды MathCAD составить программу, позволяющую при заданных законах распределения случайной величины, описывающих ситуацию возникновения несущественной и критической неисправностей в передатчике, получить среднее число сеансов связи за M дней (при N планируемых сеансах в день) при стратегиях « α » и « β », и выбрать наилучшую стратегию.

Схема алгоритма, реализующая эти стратегии, практически такая же, как в рассмотренной задаче об эксплуатации автобуса.

Очевидно, что эту задачу можно сформулировать и в более общем виде, заменив условие вида «с вероятностью s » на условие вида «значение случайной величины, с заданным законом распределения, попало в заданный интервал».

При создании программной реализации границы интервалов, соответствующих событиям «возникла несущественная неисправность» и «возникла критическая неисправность», задайте в программе самостоятельно, в соответствии с параметрами закона распределения из задания по варианту.

Возможный порядок выполнения работы в соответствии с заданием варианта.

- Создать и протестировать часть программы, реализующую получение случайной величины с заданным законом распределения в среде MathCAD (согласно варианту) так же, как это сделано в рассмотренном выше примере о распределении Коши.
- Модифицировать схему алгоритма, представленную на рис.2 применительно к своей задаче.
- Реализовать модифицированный алгоритм в среде MathCAD.
- Выполнить вычисления и на их основе рекомендовать рациональную стратегию.
- Написать отчёт по работе.

Варианты заданий для законов распределения и их параметров

Таблица 6.1

№	Входные параметры	Закон распределения
1	$\rho = 0.1$	Геометрическое
2	$m_x := 3; \sigma_x := 0.5$	Нормальное
3	$\lambda := 1$	Экспоненциальное
4	$\lambda := 1; \alpha := 1$	Вейбулла.
5	$\alpha := 0.5$	Парето.

6	$\beta := 4; \alpha := 2$	Эрланга.
7	$\lambda := 0.5; \mu := 2$	Коши.
8	$\lambda := 1; \mu := 1$	Логистическое.
9	$\lambda := 0.5$	Экспоненциальное
10	$\rho = 0.2$	Геометрическое
11	$\lambda := 1; \mu := 1$	Логистическое.
12	$\lambda := 0.5; \alpha := 1$	Вейбулла.
13	$\lambda := 0.7; \mu := 1.5$	Коши.
14	$m_x := 4; \sigma_x := 1$	Нормальное
15	$\alpha := 1$	Парето.
16	$\rho = 0.15$	Геометрическое
17	$\beta := 3; \alpha := 1$	Эрланга.
18	$\lambda := 2; \alpha := 1$	Вейбулла.
19	$\lambda := 0.7; \mu := 1.5$	Коши.
20	$\beta := 2; \alpha := 4$	Эрланга.
21	$\lambda := 0.5; \mu := 1$	Логистическое.
22	$\lambda := 2$	Экспоненциальное .
23	$m_x := 2; \sigma_x := 1.5$	Нормальное
24	$\alpha := 1.5$	Парето.
25	$\beta := 1; \alpha := 1$	Эрланга.
26	$\lambda := 0.8; \mu := 2$	Коши.
27	$\beta := 2; \alpha := 2$	Эрланга.
28	$\rho = 0.12$	Геометрическое
29	$\lambda := 2; \alpha := 0.5$	Вейбулла.
30	$m_x := 1; \sigma_x := 1$	Нормальное

Тема 9. Моделирование систем массового обслуживания

Рассмотрим несколько примеров:

1. Портовый кран перегружает контейнеры с автотранспорта на корабль. Автомобили подвозят контейнеры через случайные промежутки времени. На перемещение контейнера требуется случайный промежуток времени.
2. Имеется цех. С предыдущей технологической операции к нему поступают детали через случайные промежутки времени. На обработку партии также требуется случайный промежуток времени.
3. Имеется стойка телефонной станции, на вход которой поступают вызовы абонентов через случайные промежутки времени. На обработку вызова требуется случайный промежуток времени.

Во всех примерах имеется нечто общее. Можно выделить так называемые *заявки*, которые образуют поток. Первый пример – автомобили, подвозящие контейнеры, во втором – партии деталей, в третьем – телефонные звонки.

Во всех примерах имеются *обслуживающие аппараты* (ОА). Их еще называют *каналами*. В первом примере – это портовый кран, во втором – цех, в третьем – стойка на телефонной станции.

Это примеры систем массового обслуживания (СМО).

В зависимости от соотношения производительности обслуживающего аппарата и интенсивности потока заявок может образовываться *очередь*. В простейшем случае эту ситуацию можно изобразить следующим образом:

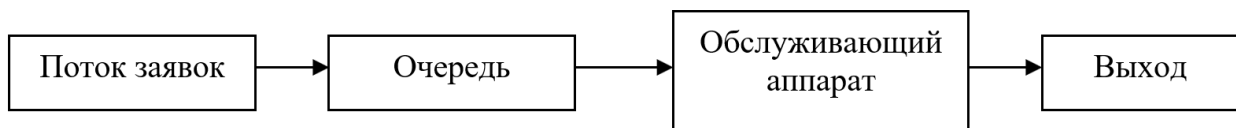


Рисунок 7.1. Схема простейшей системы массового обслуживания

Заметим, что СМО могут быть достаточно сложными: в них могут присутствовать несколько обслуживающих аппаратов (каналов) и несколько источников заявок. Обслуживание может вестись с учетом приоритетов заявок.

Основными показателями СМО являются:

1. Загрузка обслуживающих аппаратов.
2. Коэффициент простоя ОА: $k = 1 - \rho$ (где ρ - загрузка).
3. Количество заявок, обслуженных за рассмотренный промежуток времени t (производительность).
4. Средняя и максимальная длина очереди.
5. Время пребывания заявки в очереди.

Понятно, что можно определенным образом построить модель, позволяющую вычислить эти характеристики. **Основная задача при моделировании СМО** – определить типы и количество обслуживающих аппаратов, а также их связь между собой (структуру СМО) так, чтобы обеспечить максимальную требуемую производительность системы массового обслуживания при выполнении заданных ограничений (например, стоимость ОА).

Построение алгоритмической модели простейшей СМО

Введем следующие обозначения:

t_i - момент поступления i -той заявки на вход очереди;

w_i - время пребывания i -той заявки в очереди;

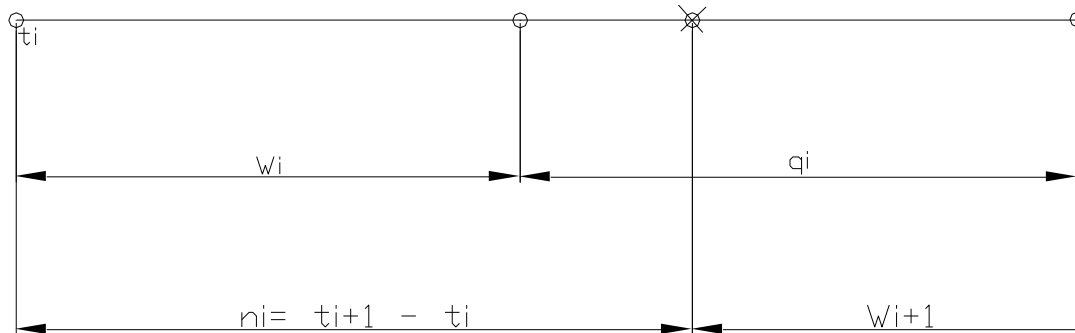
q_i - время обслуживания i -той заявки ОА;

θ_i - момент выхода i -той заявки из ОА;

$\eta_i = t_{i+1} - t_i$ - интервал времени между поступлением $i+1$ и i -той заявок на вход в очередь.

На следующем рисунке представлены две возможные ситуации для момента поступления $i+1$ заявки:

а) Ситуация 1



б) Ситуация 2

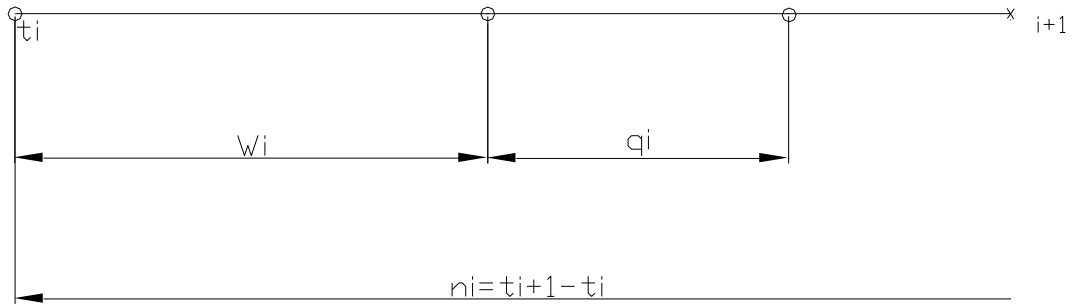


Рисунок 7.2. Две ситуации при поступлении очередной заявки

Разница в этих ситуациях заключается в том, что в ситуации а) ОА занят при поступлении $i+1$ заявки, а в ситуации б) ОА свободен и значит, $i+1$ заявка сразу начнет обрабатываться. Значит время пребывания $i+1$ -той заявки в очереди определится в соответствии с рекуррентной формулой

$$w_{i+1} = \begin{cases} 0, & w_i + q_i < \eta_i \\ w_i + q_i - \eta_i, & w_i + q_i > \eta_i \end{cases}$$

При реализации этой модели на ЭВМ следует организовать цикл для перебора моделируемого количества заявок, ввести начальные заявки.

$$1) \quad t_i = 0,$$

$$w_i = 0.$$

$$2) \quad i \geq 2$$

$$t_i = t_{i+1} + \eta_{i+1}$$

Найти $\theta_i = t_i + w_i + q_i$

3) На основании полученных данных необходимо вычислить характеристики:

- загрузку СМО;
- время загрузки;
- общее время функционирования СМО;
- производительность (среднее количество заявок, обслуживаемых за рассматриваемый период времени).

Возможная реализация этого алгоритма в MATHCAD представлена ниже.

Здесь

$M1$ -математическое ожидание интервала между моментами поступления двух последовательных заявок (нормальный закон распределения), σ^2 - среднее квадратическое отклонение этого закона.,

$M2$ -математическое ожидание времени обслуживания заявки (нормальный закон распределения), σ^2 - среднее квадратическое отклонение этого закона., n - количество заявок при моделировании. На выходе три значения – коэффициент загрузки, среднее и максимальное время пребывания заявки в очереди.

ORIGIN := 1

$F(M1, \sigma1, M2, \sigma2, n) :=$

$t_1 \leftarrow 0$
$\eta \leftarrow \text{morm}(n-1, M1, \sigma1)$
$q \leftarrow \text{morm}(n, M2, \sigma2)$
for $i \in 2..n$
$t_i \leftarrow t_{i-1} + \eta_{i-1}$
$w_1 \leftarrow 0$
for $i \in 2..n$
$w_i \leftarrow 0$ if $w_{i-1} + q_{i-1} < \eta_{i-1}$
$w_i \leftarrow w_{i-1} + q_{i-1} - \eta_{i-1}$ if $w_{i-1} + q_{i-1} \geq \eta_{i-1}$
$\Theta_1 \leftarrow q_1$
for $i \in 2..n$
$\Theta_i \leftarrow t_i + w_i + q_i$
$\sum_{i=1}^n q_i$
$\rho \leftarrow \frac{\Theta_n}{\Theta_n}$
$k \leftarrow 1 - \rho$
$pR \leftarrow \frac{n}{\Theta_n}$
$\sum_{i=1}^n w_i$
$wsr \leftarrow \frac{\sum_{i=1}^n w_i}{n}$
$wmax \leftarrow \max(w)$
return (ρ wsr $wmax$)

$F(25, 0.1, 20, 0.1, 7) = (0.829 \ 0 \ 0)$

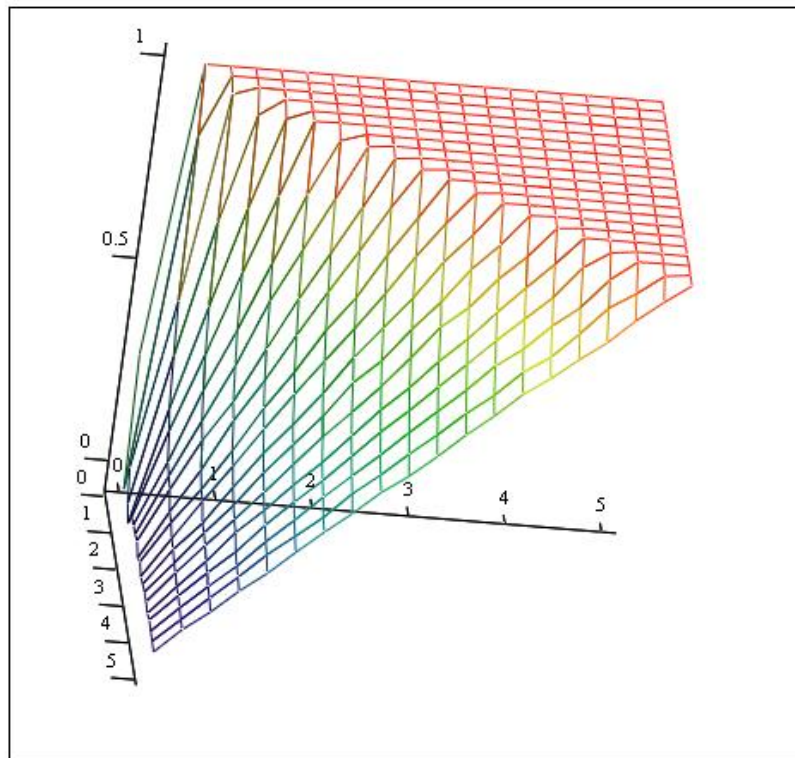
$$T(M1, M2) := (F(M1, 0.1, M2, 0.1, 50)^T)_1$$

$$Y(M1, M2) := (F(M1, 0.1, M2, 0.1, 50)^T)_2$$

$$U(M1, M2) := (F(M1, 0.1, M2, 0.1, 50)^T)_3$$

$x := 5$ $y := 5$ $mesh := 20$

$\rho := \text{CreateMesh}(T, 0, x, 0, y, mesh)$



ρ

Простейшие потоки событий

Потоком событий называется последовательность однородных событий, следующих одно за другим в случайные моменты времени. Такой поток можно изобразить как последовательность точек $t_1 \dots t_n$ на числовой оси, соответствующих случайным моментам появления событий (Рисунок 7.3):



Рисунок 7.3. Последовательность однородных событий

Поток событий называется *стационарным*, если вероятность

попадания того или иного числа событий на участок времени зависит только от длины этого участка и не зависит от того, где именно на оси t расположен этот участок.

Поток событий называется *потоком без последствия*, если для любых непересекающихся участков времени число событий, попадающих на один из них, не зависит от того, сколько событий попало на другой. Поток событий называется *ординарным*, если вероятности попадания на участок времени малой длины двух или более событий пренебрежимо малы по сравнению с вероятностью попадания на этот участок одного события.

Стационарность потока означает, что вероятностные характеристики этого потока не должны меняться в зависимости от времени. Например, такая характеристика, как *интенсивность* потока событий, *равная математическому ожиданию числа событий в единицу времени*, должна оставаться постоянной для стационарного потока.

На практике часто встречаются потоки событий, которые стационарны только на ограниченном участке времени. Например, поток вызовов, поступающих на телефонную станцию в дневное время, может считаться таковым. Тот же поток в течение целых суток уже не будет стационарным, поскольку ночью интенсивность вызовов гораздо меньше, чем днем.

Отсутствие последствия в потоке означает, что события появляются в последовательные моменты времени *независимо друг от друга*. Например, поток пассажиров, входящих на станцию метро, можно считать потоком без последствия потому, что причины, обусловившие приход отдельного пассажира именно в тот, а не другой момент, не связаны с аналогичными причинами для других пассажиров. Однако условие отсутствия последствия может быть нарушено за счет появления такой зависимости. Например, поток пассажиров, покидающих станцию метро, уже не может считаться потоком без последствия, так как моменты выхода пассажиров, прибывших одним и тем же поездом, зависимы между собой.

Ординарность потока означает, что события в потоке приходят *поодиночке*, а не парами, тройками и т. д. Например, поток клиентов, направляющихся в парикмахерскую, практически можно считать ординарным, чего нельзя сказать о потоке клиентов, идущих в ЗАГС для регистрации брака.

Простейший поток играет среди потоков событий особую роль, до

некоторой степени аналогичную роли нормального закона среди других законов распределения. Согласно центральной предельной теореме, при суммировании большого числа независимых случайных величин, подчиненных практически любым законам распределения, получается случайная величина, приближенно распределенная по нормальному закону. Аналогично можно сказать, что *при суммировании (взаимном наложении) большого числа ординарных, стационарных потоков с практически любым последствием получается поток, сколь угодно близкий к простейшему*. Условия, которые должны для этого соблюдаться, аналогичны условиям центральной предельной теоремы, а именно: складываемые потоки должны оказывать на сумму приблизительно равномерно малое влияние.

Замкнутые системы массового обслуживания

Рассмотрим системы массового обслуживания, в которых *интенсивность потока поступающих заявок зависит от состояния самих систем*. Такие системы массового обслуживания называются *замкнутыми*.

Пусть система состоит из n каналов обслуживания и m источников заявок, $m > n$.

Предположим, что каждый источник порождает простейший поток заявок с интенсивностью λ , *причем источник не может посылать следующую заявку до завершения обслуживания своей предыдущей заявки* (в этом и выражается замкнутость данной системы). Предположим также, что каждый канал порождает простейший поток обслуженных заявок с интенсивностью μ . Все состояния данной системы можно разбить условно на три группы:

S_0 - "все каналы свободны",

S_i - "ровно i каналов занято и поступило ровно i заявок", $i = 1, \dots, n$,

S_j - "все каналы заняты и ровно $j-n$ заявок находятся в очереди для обслуживания", $j = n + 1, \dots, m$.

Графически все возможные переходы из состояния в состояние, а также интенсивности потоков событий, под воздействием которых эти переходы возможны, можно изобразить в виде размеченного графа так, как это показано на Рисунке 2. Действительно, если система находится в состоянии S_i $i = 0, 1, \dots, n - 1$, то в состояние " $i+1$ каналов занято" она может перейти под воздействием суммарного потока заявок от $m - i$ источников с интенсивностью $(m - i)\lambda$; из состояния S_i в состояние S_{i-1} " $i - 1$ каналов занято" она может перейти под воздействием суммарного потока обслуженных заявок, поступающего от i каналов обслуживания с интенсивностью $i\mu$. Напомним, что i источников

прекращают поставку заявок до завершения обслуживания своих последних заявок. Если же система находится в состоянии $S_j, j = n, \dots, m-1$, то в состояние S_{j+1} она может перейти под воздействием суммарного потока заявок с интенсивностью $(m-j)\lambda$, а в состояние S_{j-1} — под воздействием суммарного потока обслуженных заявок с интенсивностью $n\mu$, поступающего от n каналов обслуживания.

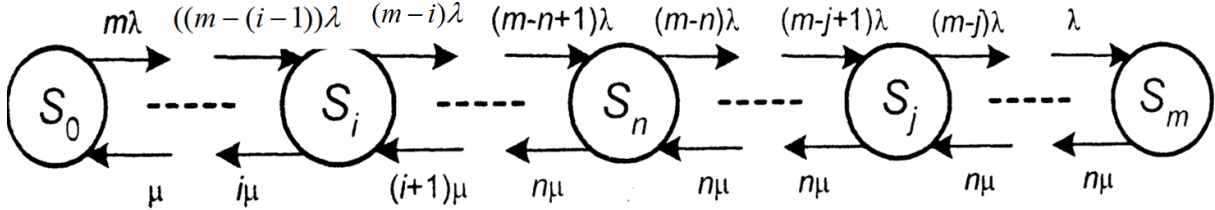


Рисунок 7.4. Размеченный граф многоканальной замкнутой СМО

Составим на основе этого размеченного графа уравнения Колмогорова. Эти уравнения представляют собой систему обыкновенных дифференциальных уравнений, описывающую вероятности $P_r(t)$ нахождения данной системы в состоянии S_r в момент времени $t, r = 0, 1, \dots, m$

$$\begin{cases} P_0'(t) = \mu P_1(t) - m\lambda P_0(t) \\ P_i'(t) = (m-i+1)\lambda P_{i-1}(t) + (i+1)\mu P_{i+1}(t) - ((m-i)\lambda + i\mu)P_i(t), \quad i=1, \dots, n-1 \\ P_j'(t) = (m-j+1)\lambda P_{j-1}(t) + n\mu P_{j+1}(t) - ((m-j)\lambda + n\mu)P_j(t), \quad j=n, \dots, m-1 \\ P_m'(t) = \lambda P_{m-1}(t) - n\mu P_m(t) \end{cases}$$

Особый интерес представляют вероятности $P_r(t)$ в предельном **стационарном** режиме, т. е. при $t \rightarrow \infty$, которые называются предельными вероятностями состояний системы.

Открытые системы массового обслуживания

Рассмотрим системы массового обслуживания, в которых **интенсивность потока поступающих заявок не зависит от состояния самих систем**. Такие системы массового обслуживания называются **открытыми**.

Пусть интенсивность простейшего потока поступающих заявок равна λ и не зависит от состояния системы. Предполагается, что система состоит из n каналов обслуживания и каждый канал порождает простейший поток обслуженных заявок с интенсивностью μ . Заявки, поступающие в момент, когда заняты все каналы, становятся в **очередь**

ожидаю обслуживания. Количество мест в очереди ограничено числом k : при наличии в очереди k заявок вновь поступающие заявки покидают систему необслуженными.

Все состояния данной системы можно разбить условно на три группы:

S_0 - "все каналы свободны",

S_i - "ровно i каналов занято и поступило ровно i заявок", $i = 1, \dots, n$,

S_j - "все каналы заняты и ровно $j-n$ заявок находятся в очереди для обслуживания", $j = n + 1, \dots, n+k$.

Графически все возможные переходы из одного состояния в другое, а также интенсивности потоков событий, под воздействием которых эти переходы возможны, можно изобразить в виде размеченного графа так, как это показано на рисунке ниже. Здесь $m=n+k$.

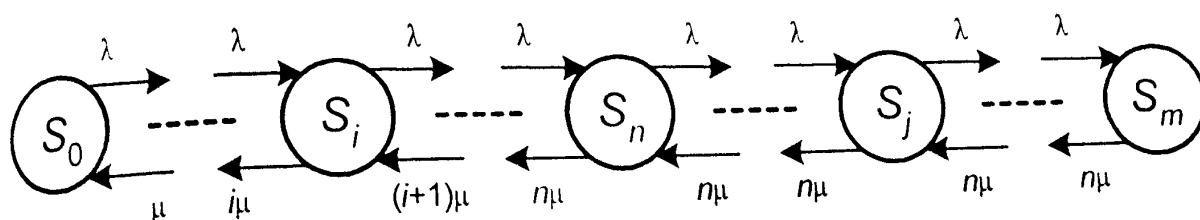


Рисунок 7.5. Размеченный граф многоканальной открытой СМО

Действительно, если система находится в состоянии S_i $i = 0, 1, \dots, m$, то в состоянии S_{i+1} " $i+1$ каналов занято" она может перейти под воздействием потока заявок с интенсивностью λ ;

Из состояния S_i в состояние S_{i-1} " $i-1$ каналов занято" $i = 1, \dots, n$ она может перейти под воздействием суммарного потока обслуженных заявок, поступающего от i каналов, с интенсивностью $i\mu$.

Из состояния S_j в состояние S_{j-1} $j = n + 1, \dots, m$, система может перейти под воздействием суммарного потока обслуженных заявок, поступающего от n каналов с интенсивностью $n\mu$.

Составим на основе этого размеченного графа уравнения Колмогорова. Приравнявая производные нулю для стационарного случая, получим систему линейных алгебраических уравнений, описывающую предельные вероятности состояний системы:

$$\begin{cases} 0 = \mu P_1 - \lambda P_0 \\ 0 = \lambda P_{i-1} + (i+1)\mu P_{i+1} - (\lambda + i\mu)P_i, & i = 1, \dots, n-1, \\ 0 = \lambda P_{j-1} + n\mu P_{j+1} - (\lambda + n\mu)P_j, & j = n, \dots, m-1, \\ 0 = \lambda P_{m-1} - n\mu P_m \\ P_0 + P_1 + \dots + P_m = 1 \end{cases}$$

Если мест в очереди не предусмотрено ($k=0$), то имеем частный случай открытой системы массового обслуживания. Графически этот случай описывается на рисунке 7.6.

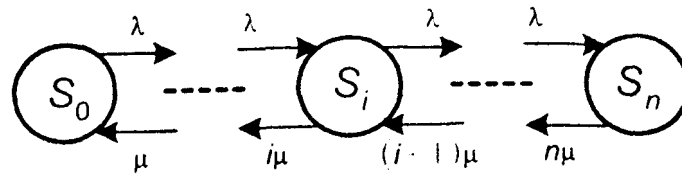


Рисунок 7.6. Размеченный граф многоканальной открытой СМО без очереди.

Для получения системы алгебраических уравнений, описывающей стационарный режим в этом случае, достаточно из последней системы удалить третий блок уравнений (при $j = n, \dots, m-1$) и положить $m = n$.

Если рассматриваемая система массового обслуживания одноканальная, то из системы линейных алгебраических уравнений исключается второй блок уравнений; если система одноканальная и без очереди, то исключаются второй и третий блоки уравнений.

Пусть система находится в предельном стационарном режиме. Тогда можно показать, что:

- вероятность P_{om} отказа заявке на обслуживание равна P_m ;
- вероятность Q принятия заявки на обслуживание равна $1 - P_m$;
- среднее число A заявок, принимаемых системой на обслуживание в единицу времени, равно λQ ;
- среднее число $N_{зан}$ занятых каналов равно A/μ ;
- среднее число $N_{оч}$ заявок в очереди равно $\sum_{i=1}^{m-n} iP_{n+i}$;
- среднее время t_w ожидания заявки в очереди равно $\frac{1}{n\mu} \sum_{i=1}^{m-n} iP_{n-1+i}$;
- среднее время t_{sys} нахождения заявки в системе равно $t_w + Q/\mu$;

Пример решения в среде MathCAD. Открытая СМО с очередью

Автозаправочная станция имеет **2** заправочных колонки. Площадка при станции допускает пребывание в очереди на заправку не более **3** машин одновременно. Если площадка занята, то очередная машина проезжает мимо. Поток машин, прибывающих для заправки, имеет интенсивность **2** машины в минуту, а процесс заправки одной машины в среднем длится **2** мин. Определить основные характеристики данной системы массового обслуживания.

Ответы:

Вероятность того, что машина будет вынуждена проехать мимо станции необслуженной, равна **0.512**; среднее число машин, обслуживаемых станцией за минуту, равно **0.976**; среднее число занятых колонок равно **1.952**; среднее число машин в очереди равно **2.176**; среднее время ожидания в очереди равно **1.088** мин; среднее время пребывания машины на станции равно **2.064** мин.

Открытая система массового обслуживания МАТЕМАТИКА ДЛЯ ЭКОНОМИСТОВ НА БАЗЕ MATHCAD

Глава 41, с. 319-323

Пример K41.12

ПОСЛЕДНЕЕ уравнение системы

должно быть заменено нормирующим

$$P_0 + P_1 + P_2 + P_3 + P_4 + P_5 = 1$$

$n := 2$				Число каналов (Две заправочные колонки)
$\mu := \frac{1}{2}$				Интенсивность обслуживания канала (К-во машин, заправляемых за ед. времени)
$r := 3$				Максимальная длина очереди
$m := n + r$				Число источников заявок
$\lambda := 2$				Интенсивность поступления заявок (К-во машин, приходящих на заправку за ед. времени)
$j := 1..m$	$P_j := 0$	$P_0 := 1$		Начальные условия для вероятностей состояний СМО
$\lambda := \lambda$	$\mu := \mu$	$P := P$		

НЕИЗМЕНЯЕМАЯ процедура формирования левой части системы линейных алгебраических уравнений относительно вероятностей состояний СМО

$$D(P) := \left\{ \begin{array}{l} D_0 \leftarrow P_1 \cdot \mu - P_0 \cdot \lambda \\ \text{for } i \in 1..n-1 \qquad \qquad \qquad \text{if } n > 1 \\ \quad D_i \leftarrow P_{i-1} \cdot \lambda + P_{i+1} \cdot (i+1) \cdot \mu - P_i \cdot (i \cdot \mu + \lambda) \\ \text{for } j \in n..m-1 \qquad \qquad \qquad \text{if } m > n \\ \quad D_j \leftarrow P_{j-1} \cdot \lambda + P_{j+1} \cdot n \cdot \mu - P_j \cdot (n \cdot \mu + \lambda) \\ D_m \leftarrow 1 - \sum_{i=0}^m P_i \\ D \end{array} \right.$$

Сформированные левые части системы линейных алгебраических уравнений относительно вероятностей состояний СМО для данной задачи

$$D(P) \rightarrow \begin{bmatrix} P_1 \cdot \mu - P_0 \cdot \lambda \\ P_0 \cdot \lambda + 2 \cdot P_2 \cdot \mu - P_1 \cdot (\mu + \lambda) \\ P_1 \cdot \lambda + 2 \cdot P_3 \cdot \mu - P_2 \cdot (2 \cdot \mu + \lambda) \\ P_2 \cdot \lambda + 2 \cdot P_4 \cdot \mu - P_3 \cdot (2 \cdot \mu + \lambda) \\ P_3 \cdot \lambda + 2 \cdot P_5 \cdot \mu - P_4 \cdot (2 \cdot \mu + \lambda) \\ 1 - P_0 - P_1 - P_2 - P_3 - P_4 - P_5 \end{bmatrix}$$

Решаем систему линейных алгебраических уравнений относительно вероятностей состояний

Given

$$D(P) = 0$$

$$\text{Prez} := \text{Find}(P)$$

Полученное решение

$$\text{Prez}^T = (8 \times 10^{-3} \quad 0.032 \quad 0.064 \quad 0.128 \quad 0.256 \quad 0.512)$$

На его основе вычисляем:

$$\text{Potv} := \text{Prez}_m \quad \text{Potv} = 0.512 \quad \text{Машина проедет необслуженной}$$

$$Q := 1 - \text{Potv} \quad Q = 0.488$$

$$A := \lambda \cdot Q \quad A = 0.976 \quad \text{Среднее число машин, обслуживаемых станцией за ед. времени}$$

$$\frac{1}{A} = 1.025 \quad \text{Среднее время обслуживания заявки} +$$

$$N_{\text{zan}} := \frac{A}{\mu} \quad N_{\text{zan}} = 1.952 \quad \text{Среднее число занятых каналов (колонок)}$$

$$N_{\text{ojid}} := \sum_{i=1}^{m-n} i \cdot \text{Prez}_{n+i} \quad N_{\text{ojid}} = 2.176 \quad \text{Среднее число машин в очереди}$$

$$tw := \frac{1}{n \cdot \mu} \cdot \sum_{i=1}^{m-n} i \cdot \text{Prez}_{n-1+i} \quad tw = 1.088 \quad \text{Среднее время ожидания в очереди}$$

$$tsys := tw + \frac{Q}{\mu} \quad tsys = 2.064 \quad \text{Среднее время пребывания машины на станции}$$

Пример решения в среде MathCAD. Нестационарный режим, замкнутая СМО с очередью

Задана система "экскаватор - самосвалы". Экскаватор погружает за один рабочий цикл 1 т грунта. Грузоподъемность самосвала равна 7 т. Число

самосвалов, обслуживающих экскаватор, равно **5**. Рабочий цикл экскаватора длится **0.295** мин, а время обращения самосвала равно **10** мин. Проанализировать поведение данной системы массового обслуживания за первые полчаса ее функционирования. Определить промежуток времени, в течение которого система переходит в стационарный режим. Определить продуктивность экскаватора, а также среднее число простаивающих машин.

Ответы:

- система переходит в стационарный режим приблизительно через **18** мин.
- среднее число простаивающих машин равно **0.746**;
- экскаватор обслуживает один самосвал в среднем в течение **2.84** минут.
- предельные вероятности состояний равны соответственно: **0.272, 0.281, 0.232, 0.144, 0.059, 0.012**.

**Замкнутая система массового обслуживания
МАТЕМАТИКА ДЛЯ ЭКОНОМИСТОВ НА БАЗЕ MATHCAD
Глава 40, с. 316
Пример K40.11**

$n := 1$	Число каналов (эскаваторов)
$\mu := \frac{1}{7 \cdot 0.295}$	Интенсивность обслуживания канала (К-во самосвалов, загружаемых за ед. времени (минуту))
$r := 4$	Максимальная длина очереди (кол-во самосвалов, которые могут ожидать погрузки)
$m := n + r$	Число источников заявок $m = 5$
$\lambda := \frac{1}{10}$	Интенсивность поступления заявок (К-во самосвалов, подъезжающих к эскаватору за ед. времени (минуту))

Начальные условия для вероятностей состояний СМО

$$j := 1..m \quad P_j := 0 \quad P_0 := 1$$

$$\lambda := \lambda \quad \mu := \mu \quad P := P$$

НЕИЗМЕНЯЕМАЯ процедура формирования левой части системы линейных алгебраических уравнений относительно вероятностей состояний СМО

$$D(P) := \left\{ \begin{array}{l} D_0 \leftarrow P_1 \cdot \mu - P_0 \cdot \lambda \cdot m \\ \text{for } i \in 1..n-1 \quad \text{if } n > 1 \\ \quad D_i \leftarrow P_{i-1} \cdot (m-i+1) \lambda + P_{i+1} \cdot (i+1) \cdot \mu - P_i \cdot [i \cdot \mu + \lambda \cdot (m-i)] \\ \text{for } j \in n..m-1 \quad \text{if } m > n \\ \quad D_j \leftarrow P_{j-1} \cdot \lambda \cdot (m-j+1) + P_{j+1} \cdot n \cdot \mu - P_j \cdot [n \cdot \mu + \lambda \cdot (m-j)] \\ D_m \leftarrow 1 - \sum_{i=0}^m P_i \\ D \end{array} \right.$$

Сформированные левые части системы линейных алгебраических уравнений относительно вероятностей состояний СМО для данной задачи

$$D(P) \rightarrow \begin{bmatrix} P_1 \cdot \mu - 5 \cdot P_0 \cdot \lambda \\ 5 \cdot P_0 \cdot \lambda + P_2 \cdot \mu - P_1 \cdot (\mu + 4 \cdot \lambda) \\ 4 \cdot P_1 \cdot \lambda + P_3 \cdot \mu - P_2 \cdot (\mu + 3 \cdot \lambda) \\ 3 \cdot P_2 \cdot \lambda + P_4 \cdot \mu - P_3 \cdot (\mu + 2 \cdot \lambda) \\ 2 \cdot P_3 \cdot \lambda + P_5 \cdot \mu - P_4 \cdot (\mu + \lambda) \\ 1 - P_0 - P_1 - P_2 - P_3 - P_4 - P_5 \end{bmatrix}$$

Решаем систему линейных алгебраических уравнений относительно предельных вероятностей состояний

Given

$$D(P) = 0$$

$$\text{Prez} := \text{Find}(P)$$

Полученное решение (**предельные вероятности состояний**)

$$\text{Prez}^T = (0.272 \quad 0.281 \quad 0.232 \quad 0.144 \quad 0.059 \quad 0.012)$$

На его основе вычисляем:

Среднее число занятых каналов

$$N_{\text{zan}} := \sum_{i=1}^n i \cdot \text{Prez}_i + n \cdot \sum_{j=n+1}^m \text{Prez}_j \quad N_{\text{zan}} = 0.728$$

Среднее число заявок, обслуживаемых СМО в ед. времени

$$A := N_{\text{zan}} \cdot \mu \quad A = 0.353$$

Среднее время обслуживания заявки (самосвала) **минут на самосвал**

$$\frac{1}{A} = 2.836$$

Среднее число заявок, поступивших

$$N_{\text{post}} := m - \frac{A}{\lambda} \quad N_{\text{post}} = 1.474$$

Среднее число заявок в очереди (**Среднее число простаивающих машин**)

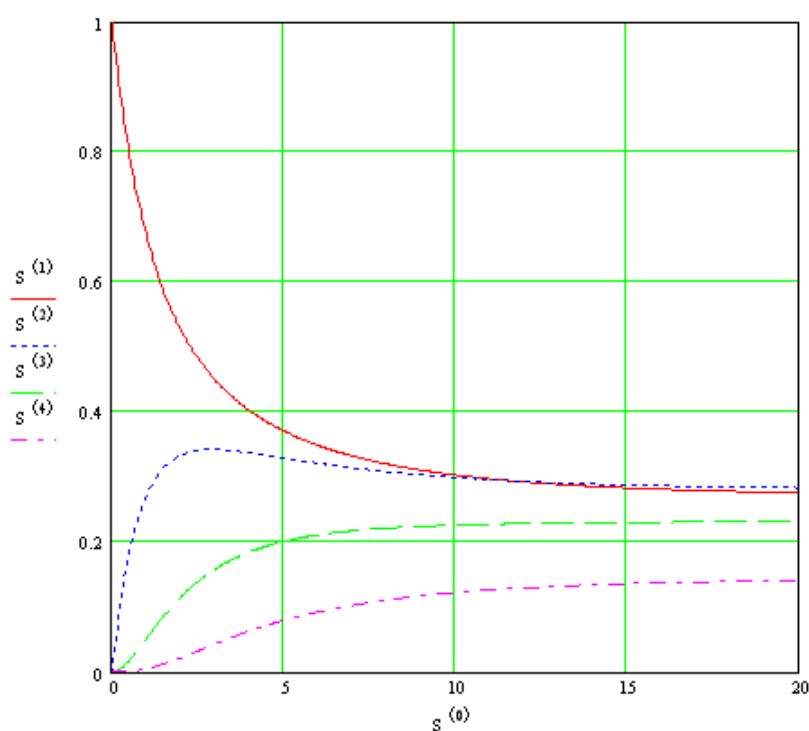
$$N_{\text{ocer}} := N_{\text{post}} - N_{\text{zan}} \quad N_{\text{ocer}} = 0.746$$

Получение функций с зависимостью от времени, описывающих изменение вероятностей нахождения системы в разных состояниях

$t_{begin} := 0$	Начальный момент наблюдения
$t_{end} := 20$	Конечный момент наблюдения
$N := 500$	Количество шагов для решения ОДУ
$DR(t,P) := D(P)$	Описание правых частей ОДУ в предписанном стандартном виде

$S := rkfixed(P, t_{begin}, t_{end}, N, DR)$ Вызов стандартной процедуры для решения ОДУ

Изменение вероятностей нахождения системы в разных состояниях в виде графиков



Литература.

1. Сигорский В.П. Математический аппарат инженера., «Техніка», 1975, 768с.
2. Черняк А.А., Новиков В.А., Мельников О.И., Кузнецов А.В. Математика для экономистов на базе MathCAD. –СПб.: БХВ-Петербург, 2003, 496 с.

Тема 10 Методы приближения инженерных и экономических данных

Тема 10. Методы приближения инженерных и экономических данных

В различных областях при проведении эксперимента инженер получает таблицу данных, эти дискретные данные необходимо обработать таким образом, чтобы можно было восстановить незамеренные значения в промежуточных точках или может быть подобрать зависимость, которая приближенно воспроизводит закон, позволяя вместо дискретных данных использовать непрерывные зависимости.

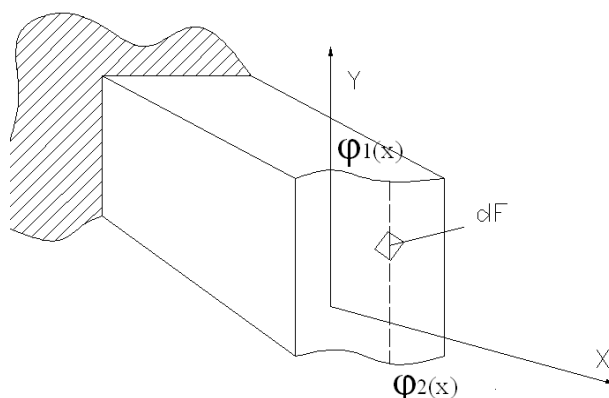


Рис.10.1.

Для того, чтобы рассчитать прочность и жесткость, необходимо знать геометрические характеристики: площадь, моменты инерции I_{xx}, I_{yy} .

$$S_x = \int_{\Omega} y dF$$

$$I_{xx} = \int_{\Omega} y^2 dF = \int_a^b (\phi_1(x) - \phi_2(x)) dx \cdot y^2$$

В этом случае необходимо знать зависимости $\phi_1(x)$ и $\phi_2(x)$.

В реальности можно выполнить замеры на кромках. Таким образом, получим таблицу данных:

X	X ₁	X ₂	X ₃	...	X _n
Y	Y ₁	Y ₂	Y ₃	...	Y _n

Возникает вопрос: как по этой таблице восстановить значения $\phi(x)$?

Существуют различные способы приближения. Одним из таких методов является интерполирование. Здесь требуют, чтобы приближающая функция совпадала с приближаемыми значениями, то есть если обозначить $F(x)$ как приближающую функцию, а приближающие значения x_i, y_i , то получим

$$F(x_i) = y_i, i = 1, \overline{N}.$$

Эти точки называются ее узлами. Чаще всего полученная таблица содержит не точные значения, а замеры, полученные с некоторой погрешностью. Поэтому требовать совпадения с табличными значениями приближенной функции неуместно. Имеет смысл использовать другие критерии близости.

1. Поточечное среднеквадратическое приближение.

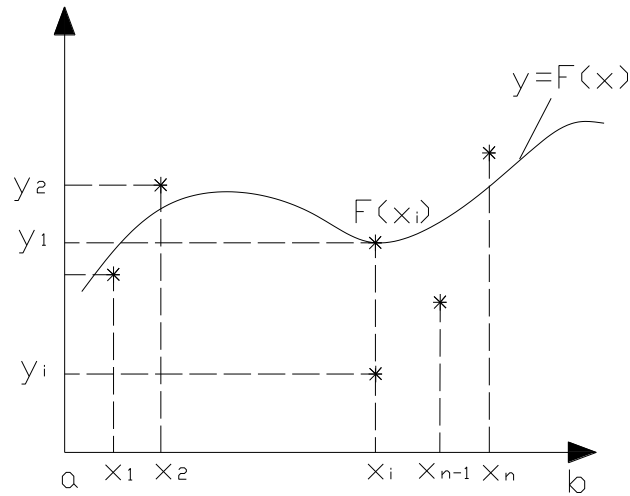


Рис.10. 2.

$$\sum_{i=1}^N (F(x_i) - y_i)^2 \rightarrow \min$$

Иногда возникает задача о приближении не табличных данных, а функции других функций. Пусть приближаемая функция $y = f(x)$ (является известной функцией), а приближающая $y = F(x)$ (подбираемая функция).

Аналогом приближения, рассматриваемого на рисунке 10.2, в этом случае будет непрерывное приближение в среднеквадратическом смысле.

2. Непрерывное приближение в среднеквадратичном смысле.

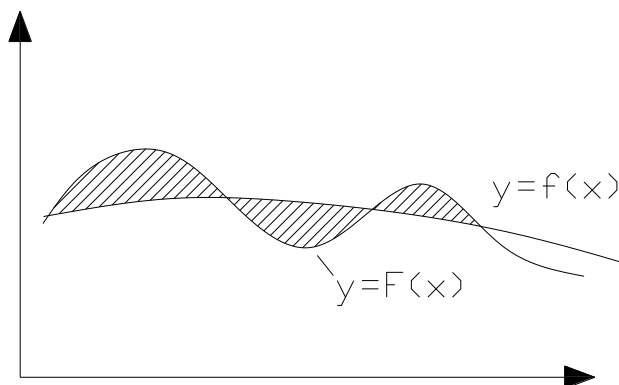


Рис.10. 3.

$$\int_a^b (f(x) - F(x))^2 dx \rightarrow \min$$

Такое приближение может использоваться в том случае, если исходная функция очень сложно или долго вычисляется, при этом ее заменяют другой функцией, которая известна.

3. *Равномерное приближение.*

$$\max_{x \in (a,b)} |f(x) - F(x)| \longrightarrow \min$$

Понятно, что равномерное распределение выдвигает самое жесткое требование.

4. *Аппроксимация и интерполяция.*

Аппроксимация является синонимом к слову «приближение», поэтому все рассмотренные ранее виды, это виды аппроксимации. А интерполяция - это один из видов аппроксимации.

Выбор функции-формы в методе конечных элементов.

При интерполяции приближаемая функция называется *интерполируемой*, а приближающая – *интерполирующей*.

Приближаемая функция при аппроксимации - *аппроксиматор*.

5. *Разложение аппроксиматора по системе базисных функций.*

Возникает естественный вопрос, за счет чего же можно изменить значения критерия при аппроксимации. Очевидно, $F(x)$ должна зависеть от параметров, варьируя которыми мы и будем менять ее вид.

Если представить график функции $F(x)$ в виде проволоки, то изменяя эти параметры, мы будем по-разному «изгибать эту проволоку».

Одним из естественных предположений для выбора функции $F(x)$ является следующее:

$$F(x) = \sum_{j=1}^m a_j \phi_j(x)$$

a_j - подбираемые варьируемые константы;

$\phi_j(x)$ - набор неизменяющихся функций, называемых базисными.

6. *Среднеквадратическая поточечная аппроксимация.*

7. *(Получение коэффициентов при поточечной аппроксимации методом наименьших квадратов).*

Пусть мы имеем (x_i, y_i) , $i = 1, \bar{n}$, тогда требование близости в среднеквадратическом смысле примет вид:

$$F(x) = \sum_{j=1}^m a_j \phi_j(x)$$

$$\sum_{i=1}^n (Y_i - F(x_i))^2 = \sum_{i=1}^n (Y_i - \sum_{j=1}^m a_j \phi_j(x))^2 = \gamma(a_1, a_2, \dots, a_m) - \min$$

где

$$a_1, a_2, \dots, a_m \in R$$

$$\frac{\delta \gamma}{\delta a_k} = 0$$

Найдем это выражение:

$$\frac{\delta \gamma}{\delta a_k} = 2 \cdot \sum_{i=1}^n (Y_i - \sum_{j=1}^m a_j \phi_j(x_i)) \cdot \frac{\delta}{\delta a_k}$$

$$(y_i - \underbrace{\sum_{j=1}^m a_j \phi_j(x_i)}_{\phi_k(x_i)}) = 0$$

$$a_1 \cdot \sum_{i=1}^n \phi_1(x_i) \cdot \phi_k(x_i) + a_2$$

$$\cdot \sum_{i=1}^n \phi_2(x_i) \cdot \phi_k(x_i) + \dots + a_m \cdot \sum_{i=1}^n \phi_m(x_i) \cdot \phi_k(x_i) = \sum_{i=1}^n Y_i \cdot \phi_k(x_i)$$

$$C_{q,p} = \sum_{i=1}^n \phi_p(x_i) \cdot \phi_q(x_i)$$

$$c_{11}a_1 + c_{12}a_2 + c_{13}a_3 + \dots + c_{1m}a_m = b_1$$

$$c_{21}a_1 + c_{22}a_2 + c_{23}a_3 + \dots + c_{2m}a_m = b_2$$

.....

$$c_{m1}a_1 + c_{m2}a_2 + c_{m3}a_3 + \dots + c_{mm}a_m = b_m \quad (10.1)$$

$$C \cdot \bar{A} = \bar{B}$$

$$C = \{C_{q,p}\}_{\substack{p=i,m \\ q=i,m}}$$

Получена система линейных уравнений:

$$\bar{A} = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} \quad \bar{B} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}$$

где $C_{p,q} = C_{q,p}$ - симметричная матрица.

Порядок выполнения лабораторной работы.

1. ввести систему базисных функций;
2. ввести функцию-аппроксиматор (необходимо ввести два параметра x и a);
3. насчитать в n -точках табличные значения (можно эти данные считать из текстового файла, n предварительно задано);
4. вычислить коэффициент $C_{q,p}$ (получить матрицу C);
5. вычислить компоненты вектора \bar{B} ;
6. решить систему $C \cdot \bar{A} = \bar{B}$;
7. в результате необходимо получить вектор \bar{A} из искомых компонентов;
8. используя выше определенную функцию $f(x)$ и вектор \bar{A} на одном графическом поле отобразить систему табличных точек и аппроксимирующую функцию;
9. повторить выполнение документа MathCad при двух других системах $\phi(x)$.

8. О выборе системы базисной функции.

Для того, чтобы гарантировать сходимость процесса аппроксимации при увеличении числа базисной функции эти функции должны быть линейно независимы.

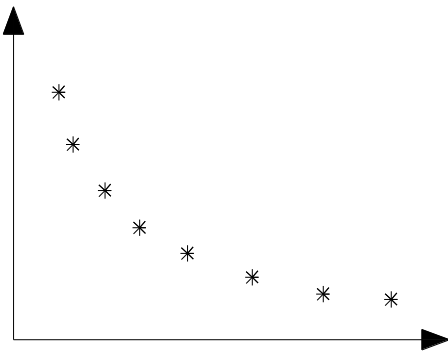


Рис.10. 4 – пример функции 1

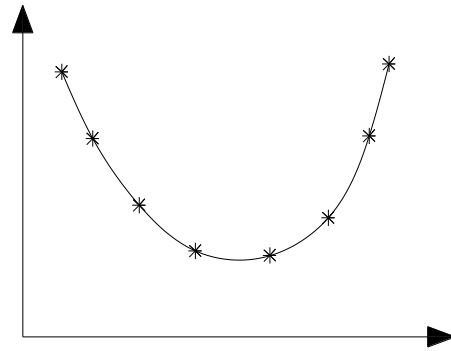


Рис.10. 5 – пример функции

Для случая, представленного на рис. 10.4 имеет смысл подбирать $\phi_k(x) = e^{-\lambda kx}$. Смысл в том, что при «хорошем» подборе базисных функций заданную точность приближения удастся достигнуть при малом числе слагаемых.

9. Интегральное среднеквадратическое приближение.

$$Q(x) = \sum_{i=0}^n G \phi_i(x) \quad (10.2)$$

Уравнение (1) представляет собой функцию-аппроксиматор. Необходимо определить $y = f(x)$ на интервале $[a, b]$, а в качестве критерия близости выбираем:

$$I_n = I_n(C_0, C_1 \dots C_n) = \int_a^b (Q(x) - f(x))^2 dx = \int_a^b (\sum_{i=0}^n C_i \phi_i(x) - f(x))^2 dx \quad (10.3)$$

$$\frac{\delta I}{\delta C_j} = 0;$$

$$j = 0, \bar{n}$$

$$\frac{1}{2} \cdot \frac{\delta I_n}{\delta C_0} = \int_a^b \left(\sum_{i=0}^n C_i \phi_i(x) - f(x) \right) \cdot \phi_0(x) dx = 0$$

$$\frac{1}{2} \cdot \frac{\delta I_n}{\delta C_1} = \int_a^b \left(\sum_{i=0}^n C_i \phi_i(x) - f(x) \right) \cdot \phi_1(x) dx = 0$$

$$\dots \dots \dots$$

$$\frac{1}{2} \cdot \frac{\delta I_n}{\delta C_n} = \int_a^b \left(\sum_{i=0}^n C_i \phi_i(x) - f(x) \right) \cdot \phi_n(x) dx = 0 \quad (10.4)$$

Введем обозначения:

$$(\phi \psi) \xrightarrow{Df} \int_a^b \phi(x) \cdot \psi(x) dx \quad (10.5)$$

$$C_0(\phi_0 \phi_0) + C_1(\phi_1 \phi_0) + \dots + C_n(\phi_n \phi_0) = (f, \phi_0)$$

$$C_0(\phi_0 \phi_1) + C_1(\phi_1 \phi_1) + \dots + C_n(\phi_n \phi_1) = (f, \phi_1)$$

$$\dots \dots \dots$$

$$C_0(\phi_0 \phi_n) + C_1(\phi_1 \phi_n) + \dots + C_n(\phi_n \phi_n) = (f, \phi_n) \quad (10.6)$$

$$a_{p,q} = (\phi_p, \phi_q)$$

$$b_i = (f, \phi_i)$$

$$A \cdot C = B \quad (10.7)$$

$$C = \begin{pmatrix} C_0 \\ C_1 \\ \dots \\ C_n \end{pmatrix}$$

Если сравнить (10.1) с формулой (10.5), то заметим, что суммирование по точкам заменено интегрированием по отрезку. Понятно, что в программной реализации в лабораторной работе по точечному среднеквадратичному приближению достаточно заменить (10.1) на формулу (10.5).

Рассмотрим пример:

Пусть мы имеем функцию $f(x) = \sqrt{x}$ на интервале $[0; 1]$. Необходимо приблизить функцией $Q(x) = C_0 + C_1 \cdot x$.

$$\phi_0(x) = x^0$$

$$\phi_1(x) = x^1$$

$$(\phi_0, \phi_0) = \int_0^1 1 \cdot 1 dx = 1$$

$$(f, \phi_0) = \int_0^1 \sqrt{x} \cdot 1 dx = \frac{2}{3}$$

$$(\phi_1, \phi_0) = \int_0^1 x \cdot 1 dx = \frac{1}{2}$$

$$(f, \phi_1) = \int_0^1 \sqrt{x} \cdot x dx = \frac{2}{5}$$

$$(\phi_1, \phi_1) = \int_0^1 x \cdot x dx = \frac{1}{3}$$

Тогда система уравнений (10.4) примет вид:

$$\left. \begin{aligned} C_0 + \frac{1}{2} \cdot C_1 &= \frac{2}{3} \\ \frac{1}{2} \cdot C_0 + \frac{1}{3} \cdot C_1 &= \frac{2}{5} \end{aligned} \right\}$$

$$C_0 = \frac{4}{15}$$

$$C_1 = \frac{4}{5}$$

$$Q(x) = \frac{4}{15} + \frac{4}{5}x$$

10. Алгебраическое интерполирование.

Алгебраическое интерполирование – это вид приближения, если приближенную функцию обозначить $y = f(x)$, то для таблицы из n точек (x_i, y_i) , $i = 1, \bar{n}$ необходимо потребовать выполнения условия:

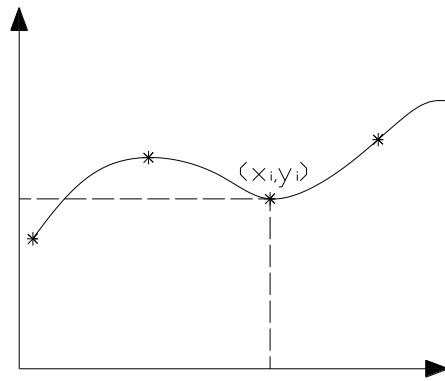


Рис.10.6 Значение приближающей функция совпадает с табличными данными в заданных точках

$$\begin{aligned} f(x_1) &= y_1 \\ f(x_2) &= y_2 \\ &\dots\dots \\ f(x_n) &= y_n \end{aligned}$$

Понятно, что в такой общей постановке решением задачи может оказаться бесконечное решение функций, поэтому, чтобы сделать задачу определенной, необходимо сузить класс подбираемых функций. Как было рассмотрено выше, при наличии определенной информации следует подбирать приближенную функцию так, чтобы она согласовывалась с приближаемыми данными. Если же такой информации нет, то одним из способов выбора является выбор полинома в качестве приближаемой функции. В этом случае интерполяцию называют *алгебраической*.

11. Алгебраическая интерполяция.

Будем искать интерполянт в виде полинома степени не выше $(n-1)$.

$$P_{n-1}(x) = a_1x^{n-1} + a_2x^{n-2} + \dots + a_n \quad (10.9)$$

Тогда условие (10.9) примет вид:

$$P_{n-1}(x_i) = y_i \quad i = \overline{1, n} \quad (10.10)$$

Понятно, что вид интерполирующего полинома полностью определяется набором коэффициента.

Рассмотрим следующее утверждение:

Пусть в n попарно-различных точках x_1, x_2, \dots, x_n заданы числа y_1, y_2, \dots, y_n , тогда существует единственный полином $P_{n-1}(x)$ степени не выше $(n-1)$, удовлетворяющий условию (10.10).

Доказательство:

Запишем это условие в каждой точке

$$\begin{aligned} a_1 x_1^{n-1} + a_2 x_1^{n-2} + \dots + a_n &= y_1 \\ a_1 x_2^{n-1} + a_2 x_2^{n-2} + \dots + a_n &= y_2 \\ &\dots \\ a_1 x_n^{n-1} + a_2 x_n^{n-2} + \dots + a_n &= y_n \end{aligned} \quad (10.11)$$

Здесь a_1, a_2, \dots, a_n - неизвестные, а правые части известны:

$$x = \begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \dots & x_1^1 x_1^0 \\ x_2^{n-1} & x_2^{n-2} & \dots & x_2^2 x_2^0 \\ \dots & \dots & \dots & \dots \\ x_n^{n-1} & x_n^{n-2} & \dots & x_n^n x_n^0 \end{bmatrix}$$

Эта матрица специального вида - матрица Вандермонда. Для нее легко показать, что определитель этой матрицы $\det(x) \neq 0$ отличен от нуля, но если это так, то система (10.10) имеет единственное решение.

Следствием этого утверждения является то, что вид интерполирующего полинома не зависит от способа его построения. А сам ход рассмотренных рассуждений дает один из способов построения такого полинома.

Другой способ построения интерполяционного полинома не требует решения системы линейных алгебраических уравнений и заключается в следующем.

Интерполяционный полином ищется в виде

$$L_{n-1}(X) = \sum_{i=1}^n Y_i \cdot p_i(X) \quad (10.12)$$

где $p_i(X)$ - полином n -й степени, принимающий значение 1 в i -м узле и значение 0 в остальных узлах интерполяции.

Очевидно, что полином (10.12) в этом случае в точках X_i будет принимать значения Y_i , т.е. требуемый критерий согласия будет выполняться.

Эта формула предполагает другой способ построения интерполяционного полинома.

Введем предварительно в рассмотрение так называемые **полиномы влияния** $P_i(x)$.

Этот полином должен удовлетворять следующим условиям:

12. степень полинома должна быть равна $(n-1)$;

$$13. p_i(x_j) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}$$

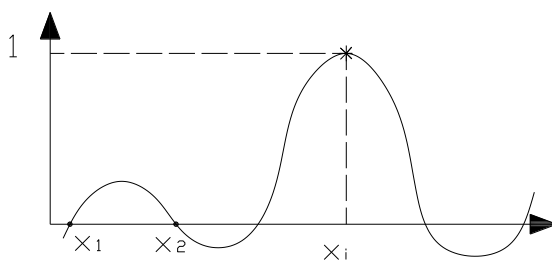


Рис.10.7 Полином влияния

Очевидно, что полином степени $(n-1)$, равный нулю во всех узлах кроме i -того.

Полиномы $p_i(X)$ были предложены Лагранжем и имеют следующий вид:

$$p_i(X) = \frac{(X-x_1) \cdot (X-x_2) \cdot \dots \cdot (X-x_{i-1})(X-x_{i+1}) \cdot \dots \cdot (X-x_{n+1})}{(x_i-x_1) \cdot (x_i-x_2) \cdot \dots \cdot (x_i-x_{i-1}) \cdot (x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_{n+1})} \quad (10.13)$$

$$p_i(x) = c_i(x-x_1) \cdot (x-x_2) \cdot \dots \cdot (x-x_{i-1})(x-x_{i+1}) \cdot \dots \cdot (x-x_n)$$

Остается определить константы C из условия $p_i = 1$, тогда

$$C_i = \frac{1}{(x_i-1)(x_i-2) \cdot \dots \cdot (x_i-x_{i-1})(x_i-x_{i+1})}$$

Подставив (10.13) в (10.12), получим следующую формулу интерполяции, которая называется формулой Лагранжа:

$$L_{n-1}(X) = \sum_{i=1}^n Y_i \cdot \frac{(X-x_1) \cdot \dots \cdot (X-x_{i-1}) \cdot (X-x_{i+1}) \cdot \dots \cdot (X-x_{n+1})}{(x_i-x_1) \cdot \dots \cdot (x_i-x_{i-1}) \cdot (x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_{n+1})} \quad (10.14)$$

Полином Лагранжа обычно обозначают $L_{n-1}(x)$.

Очевидно, что

$$\begin{aligned} L_{n-1}(x) &= \sum_{i=1}^n p_i(x) \cdot y_i \\ L_{n-1}(x) &= y_i \\ L_{n-1}(x) &= \sum_{i=1}^n y_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} \end{aligned}$$

Существуют также другие способы построения интерполяционного полинома (например, формула Ньютона). Следует отметить, что все способы в результате дают один и тот же полином, так как через $n+1$ узел можно провести только один полином n -й степени. Если аппроксимирующая функция $F(X)$ сама является полиномом n -й степени, то интерполяционный полином $P_n(X)$ при отсутствии ошибок округления совпадает с $F(X)$. В случае же произвольной функции $F(X)$ совпадение будет только в узлах интерполяции, а в промежуточных точках разность $R_n(X) = F(X) - P_n(X)$ не будет равна нулю. Величина $R_n(X)$ называется остаточным членом интерполяционной формулы (ошибкой аппроксимации).

Рассмотрим два частных случая полинома Лагранжа:

1. Пусть имеется таблица из двух точек

x_1	x_2
y_1	y_2

Тогда интерполирующий полином будет выглядеть

$$L_1(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

Это так называемый случай *линейной интерполяции*, поскольку данное уравнение - это уравнение прямой.

2. Пусть имеется таблица из координат $n=3$ точек

x_1	x_2	x_3
y_1	y_2	y_3

$$L_2(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} y_1 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_2)(x_3 - x_1)} y_3$$

Такое приближение называется параболическим или квадратическим (так как данное уравнение - уравнение параболы).

Рассмотрим пример.

Пусть задана таблица по значениям функции $y = \ln x$

x	$x_1 = 2$	$x_2 = 3$	$x_3 = 4$	$x_4 = 5$
y $= \ln x$	y_1 $= 0,6931$	y_2 $= 1,0986$	y_3 $= 1,3863$	y_4 $= 1,6094$

Построим интерполяционный полином по первым трем точкам и вычислим его значение при $x=2,5$.

$$\begin{aligned} L_2(x) &= \frac{(x - 3)(x - 4)}{(2 - 3)(2 - 4)} \cdot 0,6931 + \frac{(x - 2)(x - 4)}{(3 - 2)(3 - 4)} \cdot 1,0986 + \frac{(x - 2)(x - 3)}{(4 - 2)(4 - 3)} \cdot 1,3863 = \\ &= -0,0589x^2 + 0,7000x - 0,4713 \\ L_2(2,5) &= 0,9103 \end{aligned}$$

14. Остаточный член формулы Лагранжа.

Предположим, что все вычисления выполняются точно (без вычислительной погрешности). В этом случае в узлах интерполирования значения полинома Лагранжа будет совпадать со значениями приближаемой функции. Однако, в точках между узлами они будут отличаться на некоторую величину.

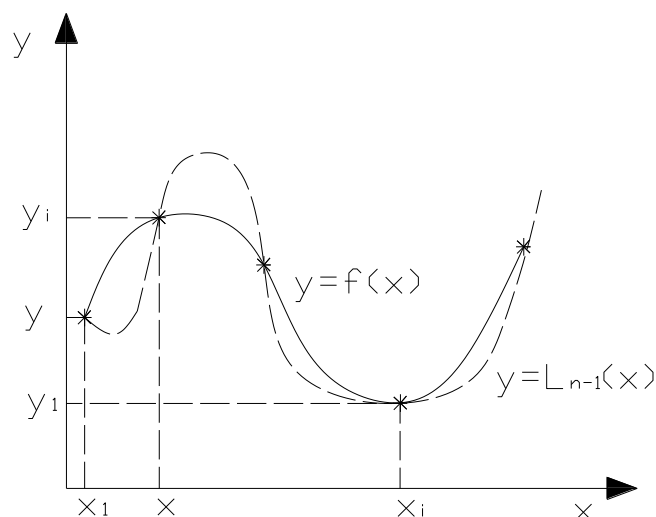


Рис.10.8 Совпадение только в узлах
 $y = L_{n-1}(x)$ - приближающая функция.

$$f(x) = L_{n-1}(x) + R_n(f, x)$$

$R_n(f, x)$ - остаточный член формулы Лагранжа.

$$R_n(f, x) = f(x) - L_{n-1}(x)$$

Понятно, что эта величина будет зависеть от класса приближаемой функции. Справедливо следующее утверждение: если $y = f(x)$ n раз непрерывно дифференцируема на интервале $[a, b]$, содержащем узлы интерполирования и точку x , то существует некоторая точка $\exists \xi \in [a, b]$, такая, что

$$R_n(f, x) = \frac{w_n(x)}{n!} \cdot f^{(n)}(\xi)$$

где $w_n(x) = (x - x_1)(x - x_2) \dots (x - x_n)$

Если обозначить $M_n = \sup_{x \in [a, b]} |f^{(n)}(x)|$, тогда получим

$$|R_n(f, x)| \leq \frac{M_n}{n!} |w_n(x)| \quad (10.15)$$

Это выражение представляет собой оценку сверху величины остаточного члена уравнения Лагранжа.

В соответствии со следствием из теоремы единственности интерполирующего полинома, остаточный член формулы Лагранжа не зависит от построения интерполирующего полинома. Значит приведенная выше оценка справедлива для любого интерполирующего полинома.

Пример.

Выше был рассмотрен пример интерполирования $y = \ln x$, где $x_i = 2, 3, 4$

$$L_2(x) = -0.0589x^2 + 0.7000x - 0.4713$$

$$\ln 2.5 = L_2(2.5) = 0.9103$$

Оценим погрешность, используя приведенную выше оценку (10.15).

$$|R_3(\ln(2.5))| \leq \frac{M_3}{3!} |w_3(2.5)|$$

$$\ln^{(3)}(x) = \frac{2}{x^3}$$

$$M_3 = \frac{1}{4}$$

$$w_3(2.5) = (2.5 - 2)(2.5 - 3)(2.5 - 4) = 0.375$$

$$|R_3(\ln(2.5))| \leq \frac{1}{4} \cdot \frac{1}{6} \cdot 0.375 = 0.0156$$

Замечание: интерполяционный полином Лагранжа может быть построен при любом расположении узлов интерполирования.

Если рассмотреть структуру полинома, то видно, что полиномы влияния $p_i(x)$ не зависят от значений приближаемой функции. Это удобно использовать при интерполировании сразу нескольких функций по одной и той же системе узлов.

Часто встречается ситуация, когда узлы оказываются равноотстоящими, то есть $x_i = x_1 + (i - 1) \cdot h$.

15. О наилучшем выборе узлов интерполирования.

Из формулы (10.15) видно, что погрешность интерполирования зависит от следующих фактов:

1. от свойств функции $f(x)$;
2. от количества узлов интерполирования ;
3. от значения узлов x_i ;

Какие из этих факторов можно изменить, чтобы уменьшить величину погрешности? Очевидно, в общем случае выбором функции $f(x)$ мы распоряжаться не можем, чаще всего количество узлов тоже фиксировано. В некоторых ситуациях можно изменять расположение узлов на интервале $[a, b]$.

$$w_n(x) = (x - x_1)(x - x_2) \dots (x - x_n)$$

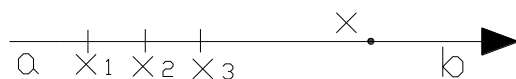


Рис.10. 9 - Иллюстрация ситуации при выборе узлов с постоянным шагом.

Чебышев показал, что наилучшим оказывается следующий выбор узлов на интервале $[a, b]$.

$$x_i = \frac{b+a}{2} + \frac{b-a}{2} \cdot \xi_i$$

$$i = 1, \bar{n}$$

$$\xi_i = \cos\left(\frac{\pi}{2} \cdot \frac{2i-1}{n}\right) \cdot |w_n(x)| \leq 2 \left(\frac{b-a}{4}\right)^n$$

В этом случае оценка (10.15) принимает вид

$$|R_n(f, x)| \leq \frac{M_n}{n!} \cdot \frac{(b-a)^n}{2^{2n-1}}$$

Заметим, что при этом узлы не являются равноотстоящими, а группируются на концах интервала. Даже при таком выборе узлов, при фиксированном n , нельзя гарантировать, что погрешность будет сколь угодно малой для произвольной функции.

16. Приближение известной функции интерполяционным полиномом.

В некоторых вычислениях часто встречается ситуация, когда вычисляемая функция имеет известное аналитическое выражение, однако, это выражение очень сложное, в том смысле, что требует большого количества вычислений. Если обращение к такой функции выполняется многократно, то объем вычислений будет велик. Возникает естественное предложение заменить (приблизить) эту функцию интерполяционным полиномом, посчитав предварительно ее значения в узлах. Если при этом вычислять полином окажется быстрее и на всем интервале мы сможем обеспечить приемлемую точность, то имеет смысл построить такой полином, удовлетворяющий условию.

$$\frac{M_n}{n!} \cdot \frac{(b-a)^n}{2^{2n-1}} < \xi \quad (10.16)$$

Эта формула позволит по заданному ξ подобрать количество узлов n для обеспечения заданной точности. Для иллюстрации такого подхода рассмотрим пример.

Пример.

Построить интерполирующий полином, приближающий функцию $y = e^x$ на промежутке $0 \leq x \leq 1$ с погрешность $\xi = 10^{-3}$.

С использованием полученного полинома вычислить значения e^x при $x = 0; 0.2; \dots; 1$.

Для вычисления функции e^x с использованием формулы Тейлора

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

для обеспечения точности $\xi = 10^{-3}$ требуется удерживать слагаемые вплоть до $\frac{x^6}{6!}$. При использовании экономного алгоритма потребуется 12 операций сложения.

Построим интерполяционный полином:

$$y^{(n)} = e^x$$

$$M_n = \sup_{[0,1]} |y^{(n)}| = e^1 = 2.7183$$

Тогда из формулы (10.16) получим:

$$\frac{2.7183}{n! 2^{2n-1}} < 10^{-3}$$

Подставляя значения n , можно убедиться, что условие выполняется при $n=4$, то есть достаточно выбрать 4 узла и интерполяционный полином будет иметь 3 степень.

Покажем, какими должны быть узлы:

$$\xi_1 = \cos \frac{\pi}{8} \text{ соответствует узел} \quad x_1 = \frac{1}{2} + \frac{0.9239}{2} = 0.9619$$

$$y_1 = 2.6168$$

$$\xi_2 = \cos \frac{3\pi}{8} \text{ соответствует узел} \quad x_2 = \frac{1}{2} + \frac{0.3827}{2} = 0.9619$$

$$y_2 = 1.9964$$

$$\xi_3 = \cos \frac{5\pi}{8} \text{ соответствует узел} \quad x_3 = \frac{1}{2} - \frac{0.3824}{2} = 0.3087$$

$$y_3 = 1.3616$$

$$\xi_4 = \cos \frac{7\pi}{8} \text{ соответствует узел} \quad x_4 = \frac{1}{2} - \frac{0.9232}{2} = 0.0381$$

$$y_4 = 1.0388$$

$$\begin{aligned} p_3(x) &= \frac{(x - 0.6913)(x - 0.3087)(x - 0.0381)}{(0.9619 - 0.6913)(0.9619 - 0.3087)(0.9619 - 0.0381)} \cdot 2.6168 + \\ &+ \frac{(x - 0.9619)(x - 0.3087)(x - 0.0381)}{(0.6913 - 0.9619)(0.6913 - 0.3087)(0.6913 - 0.0381)} \cdot 1.9964 + \\ &+ \frac{(x - 0.9619)(x - 0.3087)(x - 0.0381)}{(0.3087 - 0.6919)(0.3087 - 0.6913)(0.3087 - 0.0381)} \cdot 1.3616 + \\ &+ \frac{(x - 0.9619)(x - 0.6913)(x - 0.3087)}{(0.0381 - 0.9619)(0.0381 - 0.6913)(0.0381 - 0.3087)} \cdot 1.0388 = \\ &= 0.2782x^3 + 0.4243x^2 + 1.0156x + 0.9995 \end{aligned}$$

x	0.0	0.2	0.4	0.6	0.8	1.0
$p_3(x)$	0.9995	1.2218	1.4915	1.8217	2.2260	2.7177
e^x	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183

17. Интерполирование сплайнами.

Классическая полиномиальная интерполяция имеет следующие недостатки:

1. при увеличении количества узлов резко возрастает объем вычислительной работы, поскольку степень полинома растет;
2. даже при неограниченном увеличении числа узлов интерполяции в общем случае нельзя гарантировать достижения нужной точности приближения.

От этих недостатков свободна интерполяция с помощью сплайнов. Термин сплайн появился следующим образом. Чертежники для плавного соединения точек на чертеже использовали гибкие рейки, называемые *сплайнами*. К ним иногда подвешивали грузы, добиваясь того, чтобы сплайн прошел через заданные точки. Фактически выполнялась интерполяция таблицы с помощью полинома третьей степени.

Действительно, как известно из курса сопротивления материалов, дифференциальное уравнение изогнутой балки (в нашем случае гибкой рейки) имеет вид:

$$EI \frac{d^2y}{dx^2} = M(x)$$

Если к балке приложены силы, то $M(x) = p(x - x_i)$.

Интегрируя дважды уравнение изогнутой балки, получим кубический полином:

$$y(x) = a_{0i} + a_{1i}x + a_{2i}x^2 + a_{3i}x^3,$$

где индексы i у коэффициентов соответствуют номеру участка. Выходит, что на каждом участке уравнение линии – это кубический полином со своими коэффициентами. При этом в точках сочленения участков, в силу того, что сплайн непрерывен и не имеет «заломов», полиномы сшиваются по значениям и углам наклона. Этот факт был положен в основу идеи *полиномиальной интерполяции*. Дадим определение сплайну. Функция $S_{n,v}(x)$ называется *сплайном* степени n дефекта v (n и v - целые числа) если:

1. на каждом отрезке $(x_i, x_{i+1}) \in [a, b]$ функция $S_{n,v}(x)$ является полиномом степени n ;

2. функция $S_{n,v}(x)$ на всем интервале $[a, b]$ имеет непрерывные производные порядка до $(n - v)$ включительно.

Кусочные полиномы на каждом интервале называются *звеньями сплайна*, а условия непрерывности производных называются *условиями стыковки звеньев*.

Примеры сплайнов:

1. Сплайн первой степени.

- непрерывная кусочно-линейная функция.

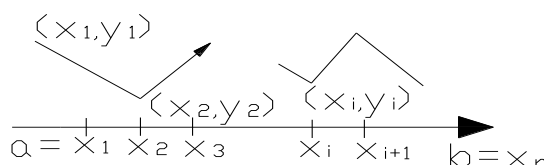


Рис.10.10 - непрерывная кусочно-линейная функция

$n = 1; v = 1; n - v = 0$ - количество независимых производных;

$$S_1(x) = \frac{(x-x_i)}{h_i} \cdot (y_{i+1} - y_i) + y_i$$

где $x \in (x_i, x_{i+1})$, а $h_i = x_{i+1} - x_i$ - длина интервала.

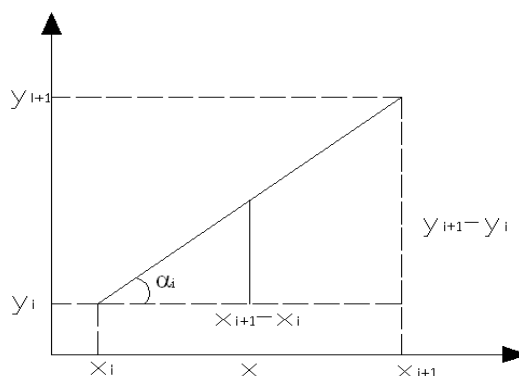


Рис.10.11 – Отдельный шаг

$$\operatorname{tg} \alpha_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = k_i$$

Сплайн первой степени относится к так называемым локальным сплайнам. Это означает, что для построения сплайна на участке (x_i, x_{i+1}) достаточно информации о значении функции.

2. Кубический сплайн дефекта 2 (или сплайн Эрмита).

Сплайн на интервале (x_i, x_{i+1}) является кубическим полиномом

$$S_{3,2}(x) = a_{i0} + a_{i1}x + a_{i2}x^2 + a_{i3}x^3$$

и на всем интервале $[a, b]$ имеет первую непрерывную производную.

$$n = 3$$

$$v = 2$$

$$n - v = 1$$

Для определения коэффициентов сплайна используем условие прохождения сплайна через узлы, то есть

$$S_{3,2}(x_i) = y_i$$

$$S_{3,2}(x_{i+1}) = y_{i+1}$$

Добавим к ним условия непрерывности производной в узлах:

$$S'_{3,2}(x_i) = m_i$$

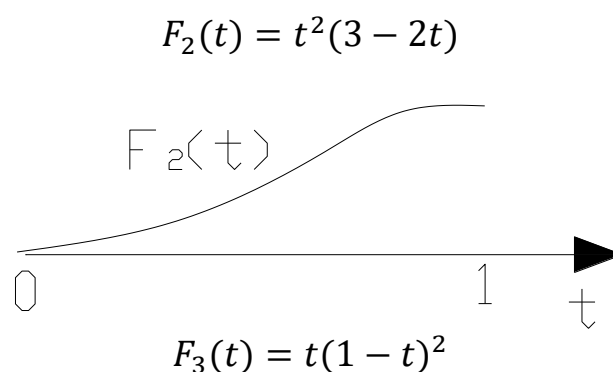
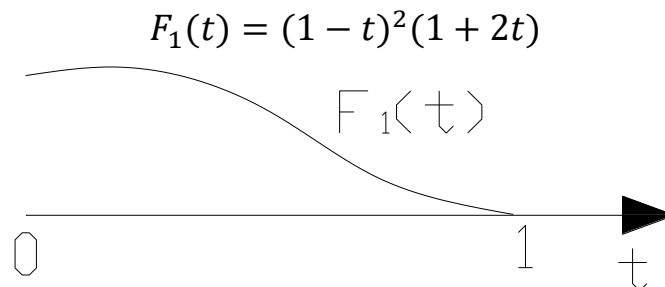
$$S'_{3,2}(x_{i+1}) = m_{i+1}$$

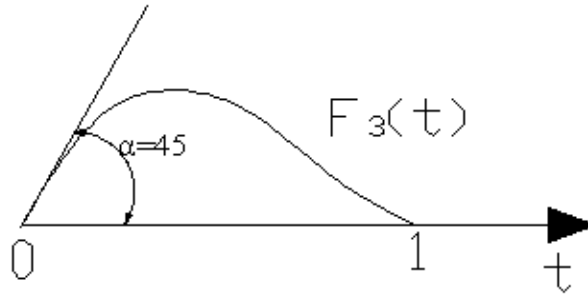
где m_i - тангенс угла наклона сплайна.

Эти четыре условия образуют систему линейных уравнений относительно коэффициентов $a_i \dots$. Если решить эту систему и подставить значения коэффициентов в формулу для сплайна, то получим:

$$S_{3,2}(x) = y_i \cdot F_1(t) + y_{i+1} \cdot F_2(t) + m_i \cdot h_i \cdot F_3(t) + m_{i+1} \cdot h_{i+1} \cdot F_4(t).$$

где $h_i = x_{i+1} - x_i$ - длина интервала, а $F_1(t), F_2(t), F_3(t), F_4(t)$ - полиномы Эрмита.





$$F_4(t) = -t^2(1-t)$$

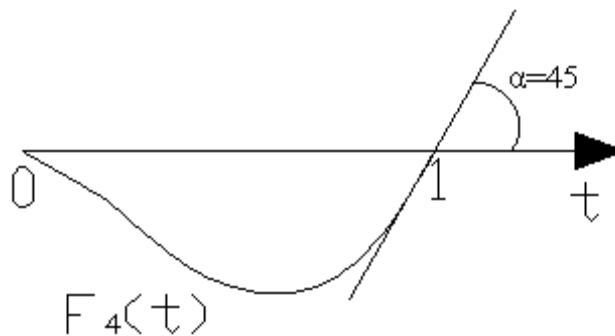


Рис.10.12 – полиномы Эрмита

$$t = \frac{x - x_i}{h_i}$$

$$x = x_i \quad t = 0$$

$$x = x_{i+1} \quad t = 1$$

Как видно, кроме значений y_i необходимо знать и наклон в узле. В практических задачах наклоны обычно неизвестны. В этом случае эти наклоны предварительно вычисляют по формулам приближенного дифференцирования.

Без вывода приведем формулы для вычисления этих наклонов. Во внутренних точках интервала формулы имеют вид:

$$m_i = y'_i = L_i \frac{y_i - y_{i-1}}{h_{i-1}} + M_i \frac{y_{i+1} - y_i}{h_i} \quad (10.17)$$

где $i = 2, 3, \dots, (n-1)$

Для крайних точек формулы имеют вид:

$$m_1 = y'_1 = (1 + M_2) \frac{y_2 - y_1}{h_1} - M_2 \frac{y_3 - y_2}{h_2} \quad (10.18)$$

$$m_n = y'_n = L_{n-1} \frac{y_{n-1} - y_{n-2}}{h_{n-2}} + (1 + L_{n-1}) \frac{y_n - y_{n-1}}{h_{n-1}} \quad (10.19)$$

где

$$\left. \begin{aligned} L_i &= 1 - M_i \\ M_i &= \frac{h_{i-1}}{h_i + h_{i-1}} \end{aligned} \right\} \quad (10.20)$$

Алгоритм для вычисления Эрмита сплайна следующий:

1. определить функции Эрмита $F_1(t), F_2(t), F_3(t), F_4(t)$;
2. задать имя функции $Sp_Erm(x, XX, YY)$,
где x - скалярное значение точки,
 XX - массив узлов $x_1, x_2 \dots x_n$,
 YY - массив значений функции $y_1, y_2 \dots y_n$.
3. необходимо проверить, не совпадает ли x с одним из узлов $|x - x_i| < \zeta$.
- если совпадает, то возвращаем значение функции y_i и производим выход из процедуры (break);
- если нет, то
 - 3.1. организовываем цикл для перебора подинтервалов (до (n-1));
 - 3.2. в цикле определяем на какой из подинтервалов попадает x ;
 - 3.3. считаем коэффициенты L_i и M_i по формулам (10.20);
 - 3.4. если точка находится на первом интервале, то считаем m_1 по формуле (10.18), а m_2 по формуле (10.17);
 - 3.5. если точка находится на последнем интервале, то необходимо посчитать m_n по формуле (10.19), а m_{n-1} по формуле (10.17);
 - 3.6. если точка находится на внутреннем интервале, то считаем m_i и m_{i+1} по формуле (10.17);
 - 3.7. записываем формулу $S_{3,2}(x)$;
4. при выполнении лабораторной работы необходимо насчитать таблицу x_i и y_i , выбрав любую функцию;
5. используя процедуру, вычислить значения сплайна в промежуточных точках с шагом

$$\Delta h = \frac{\min|x_{i+1} - x_i|}{10};$$

6. на одном графике представить сходную функцию и приближающий ее сплайн.

18. Интерполяция кривых локальными сплайнами.

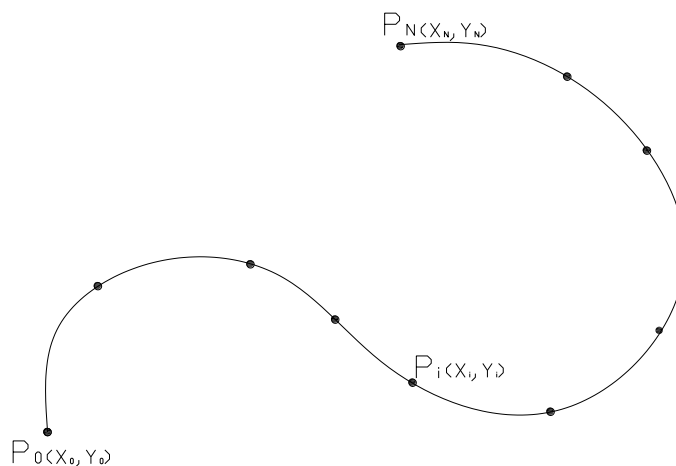


Рис.10.13 – Кривая и заданные точки на ней

Пусть имеем кривую, известны только координаты точек $P_i(x_i, y_i)$. В этом случае уже не выполняется условие упорядоченности абсцисс, которое было обязательным при построении приближающих полиномов или сплайнов для обычной функции. Однако и в этом случае можно развить аппарат интерполяции сплайнами плоских или пространственных кривых.

Поступим следующим образом, введем естественную параметризацию кривой.

$$\begin{cases} x = x(S) \\ y = y(S) \end{cases}$$

S – в данном случае это длина дуги, отсчитываемая от точки P_0 . Тогда углу P_i будет соответствовать единственное значение S_i .

$$P_i \longleftrightarrow S_i$$

$$P_0 \longleftrightarrow S_0 = 0$$

$$P_N \longleftrightarrow S_N = L$$

L - общая длина кривой.

Рассмотрим интерполяционный сплайн первой степени.

$$\begin{cases} S_0(x_i, S) = (1-t) \cdot x_i + t \cdot x_{i+1} \\ S_1(y_i, S) = (1-t) \cdot y_i + t \cdot y_{i+1} \end{cases} \quad (10.21)$$

$$\text{где } t = \frac{S-S_i}{l_i}$$

$$l_i = S_{i+1} - S_i$$

Геометрически такой сплайн представляет собой ломаную, состоящую из кривых, соединяющих между собой точки:

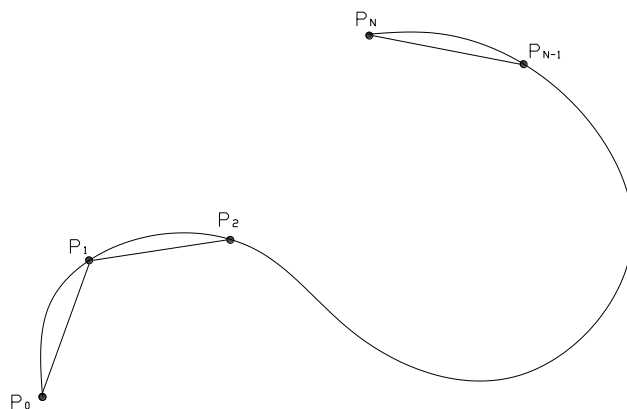


Рис.10.14 – Кривая приближенная отрезками

Из условия (10.21) можно получить

$$\frac{S_1'(y_i, S)}{S_1'(x_i, S)} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (10.22)$$

где $x \neq x_{i+1}$, а производная взята по параметру t .

Как видно, это отношение представляет собой тангенс угла наклона звена сплайна:

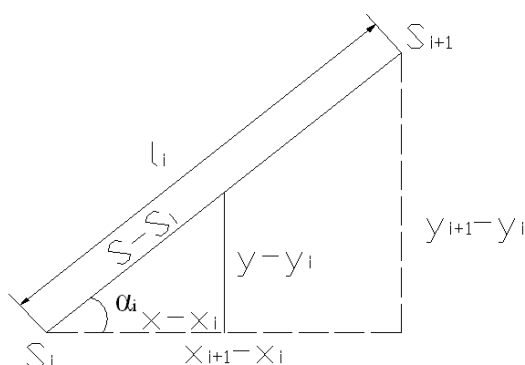


Рис.10.15 –Звено сплайна

$$tg \alpha = \frac{S_1'(y_i, S)}{S_1'(x_i, S)}$$

$$\frac{x_{i+1} - x_i}{l_i} = \frac{x - x_i}{S - S_i}$$

$$S_i(x_i, S) = x_i + \frac{S - S_i}{l_i} (x_{i+1} - x_i)$$

$$S_1(x_i, S) = x_i + t(x_{i+1} - x_i) = (1 - t) \cdot x_i + t \cdot x_{i+1}$$

В этом случае, если звено сплайна $x_i = x_{i+1}$ параллельно оси y .
 t - безразмерный параметр, который изменяется от 0 до 1.

$$t \in [0, 1]$$

$$t = 0 \Leftrightarrow S = S_i$$

$$t = 1 \Leftrightarrow S = S_{i+1}$$

Отметим интересное свойство сплайна, которое заключается в том, что тангенс угла наклона не зависит от S . Положение точки определяется параметром t , изменяя значение которого от 0 до 1 можно получать промежуточные значения на звене сплайна.

Как видно на этом примере (формула 1) параметрический сплайн первой степени в случае плоской кривой представляет собой пару обычных сплайнов. Один для координаты x , а второй для координаты y . В качестве независимой переменной выступает S . Если кривая пространственная, то добавится такая же формула для координаты z .

19. Параметрический Эрмитов кубический сплайн.

Этот сплайн представляет собой совокупность двух Эрмитовых сплайнов.

$$\begin{aligned} S_{3,2}(x, S) \\ S_{3,2}(y, S) \end{aligned}$$

$$\begin{cases} S_{3,2}(x, S) = x_i \cdot (1-t)^2(1+2t) + x_{i+1} \cdot t^2(3-2t) + x'_i t(1-t)^2 \cdot l_i - x'_{i+1} \cdot t^2(1-t) \cdot l_i \\ S_{3,2}(y, S) = y_i \cdot (1-t)^2(1+2t) + y_{i+1} \cdot F_2(t) + y'_i \cdot F_3(t) \cdot l_i - y'_{i+1} \cdot F_4(t) \cdot l_i \end{cases} \quad (10.23)$$

x'_i - производная по S от $x(S)$ в точке S_i .

Для возможности вычисления по формуле (10.23) необходимо определить

$$l_i, x'_i, x'_{i+1}, y'_i, y'_{i+1}.$$

Поскольку в реальных задачах информация о наклонах обычно отсутствует, то, как и в случае обычного Эрмитова сплайна, заменим их приближенными значениями.

Поскольку точное значение параметра вычислить невозможно, то будем строить Эрмитов сплайн близкий к сплайну (10.23) в некотором смысле.

Во-первых, для описания сплайна введем параметризацию по суммарной длине хорд.

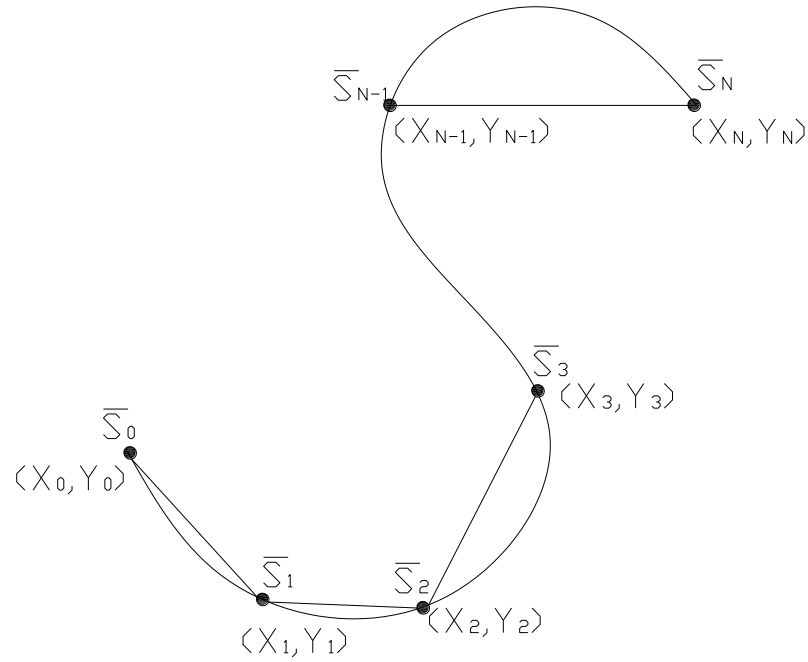


Рис.10.16 – Параметризация по суммарной длине хорд

$$\tilde{S}_0 < \tilde{S}_1 < \dots < \tilde{S}_N$$

$$\tilde{S}_i = \sum_{k=0}^{i-1} dk$$

$$dk = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}$$

$$\tilde{S}_0 = 0$$

$$\tilde{S}_1 = d_0$$

$$\tilde{S}_2 = d_0 + d_1$$

$$\tilde{S} = [0; S_N]$$

$$S_{3,2}(x, \tilde{S}) = x_i \cdot F_1(t) + x_{i+1} \cdot F_2(t) + x'_i \cdot F_3(t) \cdot d_i - x'_{i+1} \cdot F_4(t) \cdot d_i$$

где $t = \frac{\tilde{S} - \tilde{S}_i}{d_i}$

Во-вторых, точные значения производных заменим по приближенным разностным формам.

$$\tilde{x}'_0 = \frac{(1 + \tilde{\mu}_1)(x_1 - x_0)}{d_0} - \frac{\tilde{\mu}_1(x_2 - x_1)}{d_1} \quad (10.24)$$

$$\tilde{x}'_i = \frac{\tilde{\mu}_i(x_{i+1} - x_i)}{d_i} + \frac{\tilde{\lambda}_i(x_i - x_{i-1})}{d_{i-1}} \quad (10.25)$$

$$\tilde{x}'_N = \frac{(1 + \tilde{\lambda}_{N-1})(x_N - x_{N-1})}{d_{N-1}} - \frac{\tilde{\lambda}_{N-1}(x_{N-1} - x_{N-2})}{d_{N-2}} \quad (10.26)$$

где $\tilde{\mu}_i = \frac{d_{i-1}}{d_{i-1} + d_i}$
 $\tilde{\lambda}_i = 1 - \tilde{\mu}_i$

Эти формулы используются в том случае, если кривая не замкнута. Если кривая замкнута, то вместо формул (10.24) и (10.26) используем:

$$\widetilde{x}_0 = \widetilde{x}_N = \frac{d_{N-1}}{d_0+d_{N-1}} \cdot \frac{(x-x_0)}{d_0} + \frac{d_0}{d_0+d_{N-1}} \cdot \frac{(x_N-x_{N-1})}{d_{N-1}}$$

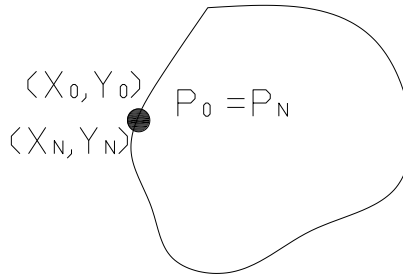


Рис.10.16 Замкнутая кривая

Рекомендации по выбору узлов:

1. Следует выбирать узлы так, чтобы $d_{i-1} = d_i + O(d_i^2)$ (то есть, чтобы длины звеньев были практически одинаковы);
2. В точках излома кривой следует вводить по два близких узла. В этом случае будет снижена осцилляция, заключающаяся в том, что сплайн может сильно уклониться от истинной кривой.

2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

Лабораторная работа Модели колебательных систем

Исследуйте поведение одномассовой колебательной системы, расчетная схема которой представлена на рисунке ниже при различных законах действия внешней нагрузки $P(t)$ и сочетаниях параметров k , c , m .

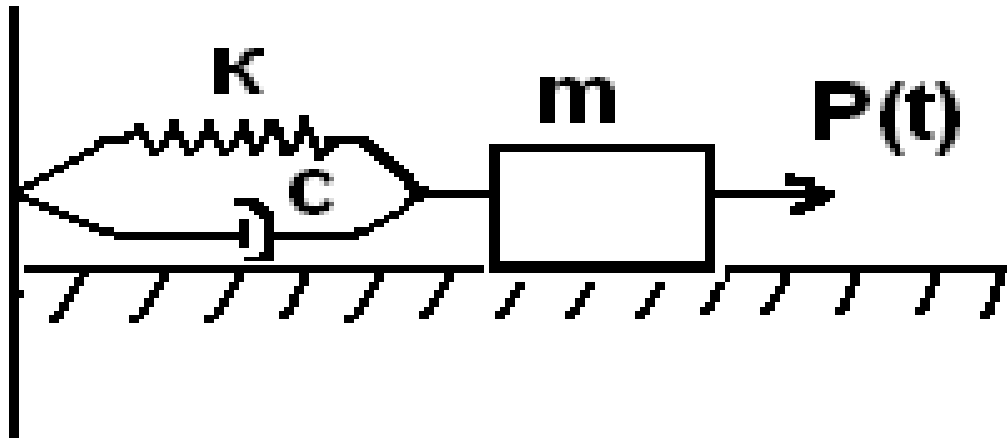


Рисунок П2.1 Колебательная одномассовая система.

1. Свести ДУ второго порядка вида

$$m \cdot \ddot{x} + c \cdot \dot{x} + k \cdot x = p(t)$$

к системе ДУ первого порядка.

2. При $p(t)=0$ получить $x(t)$, $\dot{x}(t)$ при разных c .
3. При $c=0$ для разных законов $p(t)$ получить $x(t)$, $\dot{x}(t)$
 - $p(t)=\text{const}$;
 - $p(t)=\sin(2\omega t)$, $\omega = \omega_0$, $\omega_0 = \sqrt{\frac{k}{m}}$;
 - $p(t)=\sin(\omega t)$, $\omega = \omega_0$, $\omega_0 = \sqrt{\frac{k}{m}}$;
 - $p(t)=t$;
4. При $c \neq 0$ для разных законов $p(t)$ получить $x(t)$, $\dot{x}(t)$
 - $p(t)=\text{const}$;
 - $p(t)=\sin(2\omega t)$, $\omega = \omega_0$, $\omega_0 = \sqrt{\frac{k}{m}}$;

- $p(t) = \sin(\omega t)$, $\omega = \omega_0$, $\omega_0 = \sqrt{\frac{k}{m}}$;
- $p(t) = t$.

Вы можете использовать в качестве примера следующий документ MATHCAD

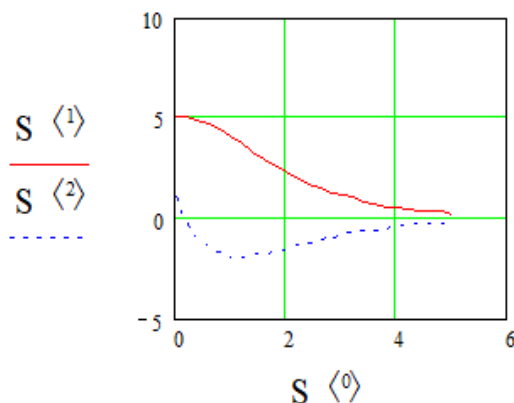
$$t_0 := 0 \quad t_1 := 5 \quad h := 0.2$$

$$N := \frac{t_1 - t_0}{h} \quad M := 1 \quad K := 1 \quad p(t) := 0$$

$$f(t, z) := \begin{pmatrix} z_1 \\ \frac{p(t) - C \cdot z_1 - K \cdot z_0}{M} \end{pmatrix} \quad R := \begin{pmatrix} 5 \\ 1 \end{pmatrix} \text{ вектор начальных значений}$$

$$S := \text{Rkadapt}(R, t_0, t_1, N, f)$$

метод Рунге-Кутты с фиксированным шагом



Лабораторная работа Модель «хищник жертва»

Исследуйте поведение системы «хищник жертва» на основе системы ОДУ

$$\begin{aligned} \dot{x}_1 &= (a - bx_2)x_1 \\ \dot{x}_2 &= (-c + dx_1)x_2 \end{aligned}$$

Задавайте разные начальные условия и значения параметров a , b , c , d .
Данные возьмите из таблицы, представленной ниже.

Таблица П2.1. Начальные условия и значения параметров a , b , c , d
Вариант 1.

Параметр	Львы	Антилопы
----------	------	----------

Начальная численность	30	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

Вариант 2.

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

Вариант 3.

Параметр	Львы	Антилопы
Начальная численность	100	200
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

Вариант 4.

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	4
Межвидовое взаимодействие	0,001	0,01

Вариант 5.

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,04	2
Межвидовое взаимодействие	0,001	0,01

Вариант 6.

Параметр	Львы	Антилопы
Начальная численность	200	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,004	0,01

Вы можете использовать в качестве примера следующий документ MATHCAD.

Параметр	Львы	Антилопы
Начальная численность	30	100
Прирост/Смертность	0,02	2
Межвидовое взаимодействие	0,001	0,01

ORIGIN := 1

$x_{begin} := \begin{pmatrix} 100 \\ 30 \end{pmatrix}$ начальная численность Антилоп
 начальная численность Львов

$c := 0.02$ Прирост (смертность) Львов

$a := 2$ Прирост (смертность) Антилоп

$d := 0.001$ Межвидовое взаимодействие Львы-Антилопы (коэффициент защищенности жертв от хищников)

$b := 0.01$ Межвидовое взаимодействие Антилопы-Львы (коэффициент уничтожения хищниками жертв).

$t_{begin} := 0$ Начальный момент наблюдения

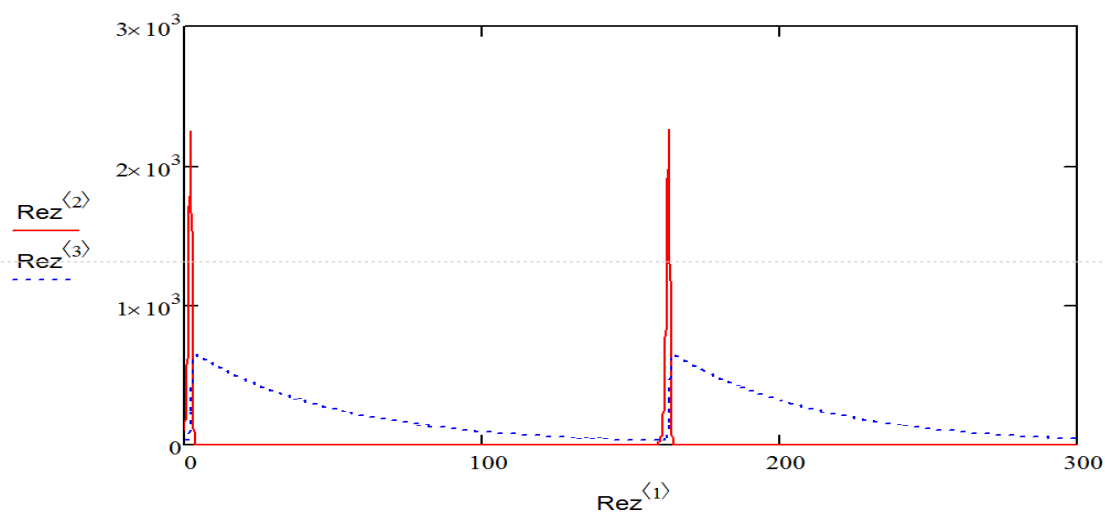
$t_{end} := 300$ Конечный момент наблюдения

$N := 4000$ Количество шагов по времени

$$P(t, x) := \begin{bmatrix} x_1 \cdot (a - b \cdot x_2) \\ x_2 \cdot (-c + d \cdot x_1) \end{bmatrix} \quad \begin{array}{l} + \\ \text{Правые части системы ОДУ} \\ \text{модели "хищник-жертва"} \end{array}$$

Rez := rkfixed(xbegin, tbegin, tend, N, P)

График изменения количества хищников и количества жертв от времени



Лабораторная работа Случайные величины в моделировании

Задание 1.

Для разных вариантов вычислите значения площадей фигур, точки которых удовлетворяют одновременно двум ограничениям вида

$$f(x, y) > 0$$

$$g(x, y) < 0$$

Предварительно изобразите кривые, описывающие границы:

$$f(x, y) = 0$$

$$g(x, y) = 0$$

Номер варианта	$f(x, y)$	$g(x, y)$
1	$-(x + 5)^2 - y + 6$	$\sin(x) - y - 7$
2	$\sqrt{ x + 1} - y$	$ x ^3 - y$
3	$x - y^2$	$x^2 + y^2 - 1$
4	$\cos(x) - y$	$x^2 - y - 5$
5	$-y^2 - x^2 + 5$	$\cos(x - 1) + 1.1 - y$
7	$-y^2 - x^2 + 7$	$e^{-x} - y$
8	$y - x - 2 $	$y - \sin(x - 1)$

Используйте пример документа MATHCAD для вычисления площади фигуры, ограниченной двумя кривыми с уравнениями:

$$-(x + 5)^2 - y + 6 = 0$$

$$\sin(x) - y - 7 = 0$$

Исходные функции: $f(x, y) := -(x + 5)^2 + 6 - y$ $g(x, y) := \sin(x) - 7 - y$

Вспомогательные функции $ab(x, y) := x$ $cd(x, y) := y$

Начальные "затравочные" значения $x := 0$ $y := 0$

Поиск границ области:

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Minimize}(ab, x, y)$ $a := ab(z_0, z_1)$ $a = -8.697$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Maximize}(ab, x, y)$ $b := ab(z_0, z_1)$ $b = -1.264$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Minimize}(cd, x, y)$ $c := cd(z_0, z_1)$ $c = -8$

Given

$f(x, y) \geq 0$ $g(x, y) \leq 0$ $z := \text{Maximize}(cd, x, y)$ $d := cd(z_0, z_1)$ $d = 6$

Функция, вычисляющая количество точек, попавших в область и их координаты:

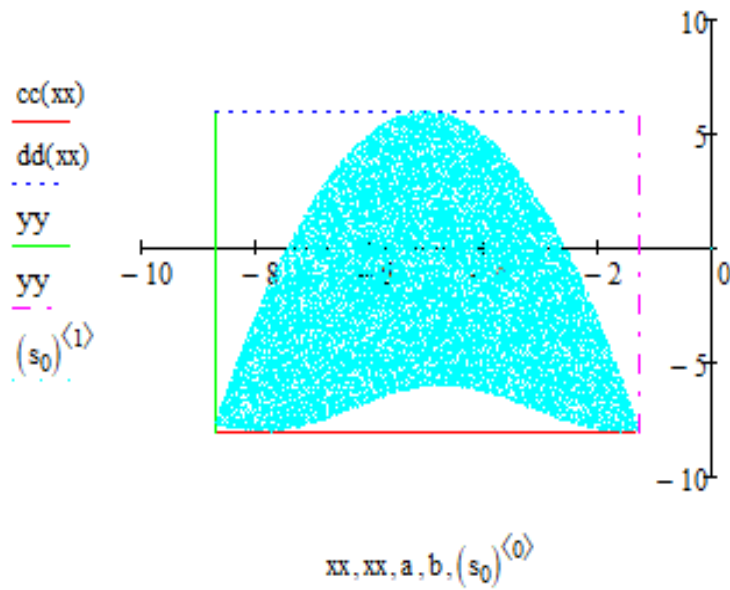
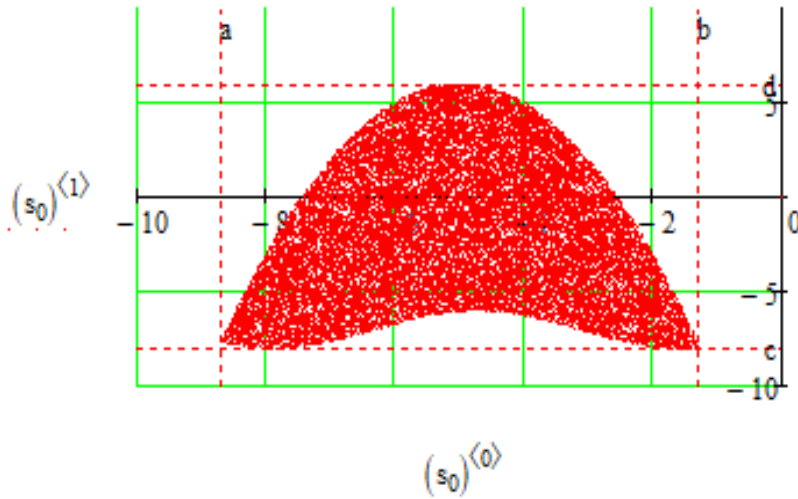
```

p(n) :=
  n1 ← 0
  for i ∈ 1..n
    x ← md(b - a) + a
    y ← md(d - c) + c
    if 1 = (g(x, y) ≤ 0) · (f(x, y) ≥ 0)
      n1 ← n1 + 1
      ti,0 ← x
      ti,1 ← y
  rez ← ( t
          n1 )
  rez
  n := 40000
  s := p(n)

```

$cc(x) := c$ $dd(x) := d$ $aa(x) := x$ $bb(x) := x$
 $xx := a, a + 0.1.. b$
 $yy := c, c + 0.1.. d$

Графическое отображение:



Сравнение полученного значения с значением вычисленным интегралом

$$s := (b - a) \cdot (d - c) \cdot \frac{s_1}{n} \quad s = 63.999$$

$$s1 := \int_a^b 13 - \sin(x) - (x + 5)^2 dx \quad s1 = 63.452$$

На рисунке красным цветом отображены точки, попавшие в область.

Задание 2.

На основе изложенной методики вычислите объем тела, ограниченного цилиндрической поверхностью с радиусом 2 единицам. Подошва цилиндра расположена на плоскости OXY. Высота цилиндрической поверхности равна 3 единицам. Сверху цилиндр накрыт полусферой.

Лабораторная работа Конкурирующие стратегии эксплуатации автобуса

Для разных вариантов заданных вероятностей поломок a , b и заданном числе предполагаемых рейсов N в день вычислите количество выполненных рейсов в день для двух стратегий эксплуатации автобуса.

Номер варианта	a	b	N
1	0.1	0.7	50
2	0.2	0.1	100
3	0.3	0.5	150
4	0.4	0.2	50
5	0.5	0.4	100
6	0.6	0.3	150
7	0.7	0.8	50

Вы можете использовать в качестве примера следующий документ MATHCAD, где приведено решение для трех ситуаций при

$$a=0, b=1, N=100.$$

$$a=1, b=0, N=100$$

$$a=1, b=1, N=100$$

ORIGIN:- 1

Процедура вычисления числа передач при стратегиях α и β , а также разницы в рейсах при данных стратегиях

```

Func(a,b,N,Dmax) :-
  D ← 0      Текущий номер моделируемого дня
  SRA ← 0    Общее число передач за D
  SRB ← 0    дней при стратегиях  $\alpha$  и  $\beta$ 
  SZ ← 0     Разница в числе передач в стратегиях  $\alpha$  и  $\beta$ 
  SG2 ← 1    Флаг обозначает хорошее состояние стратегии  $\beta$ .
  SB2 ← 0    НЕ критическое состояние при стратегии  $\beta$ 
  while D < Dmax
    RA ← 0    Количество действительно выполненных рейсов
    RB ← 0    в течении дня при стратегиях  $\alpha$  и  $\beta$ 
    RM ← 0    соответственно.
    SG1 ← 1
    if SB2 = 1 ПОЛНЫЙ ремонт ПРОПУСК всех рейсов дня стратегии  $\beta$ .
      SB2 ← 0  НЕ критическое состояние при стратегии  $\beta$ 
      SG2 ← 1  Хорошее состояние стратегии  $\beta$ .
    while RM < N
      RM ← RM + 1
      if SG1 = 1 Хорошее состояние при стратегии  $\alpha$ 
        RA ← RA + 1 Счётчик рейсов в день при стратегии  $\alpha$ 
        R $\alpha$  ← runif(1,0,1)
        SG1 ← 0 if R $\alpha$  <  $\alpha$  Есть переход в ухудшенное состояние
                  при стратегии  $\alpha$ .
        SG1 ← 1 otherwise
      if SB2 = 0 НЕ критическое состояние при стратегии  $\beta$ 
        R $\beta$  ← runif(1,0,1)
        if SG2 = 1 Хорошее состояние стратегии  $\beta$ .
          SG2 ← 0 if R $\alpha$  <  $\alpha$  Есть переход в ухудшенное состояние
                    при стратегии  $\beta$ .
          RB ← RB + 1 Счётчик рейсов в день при стратегии  $\beta$ 
        otherwise
          SB2 ← 1 if R $\beta$  <  $\beta$  Переход в критическое состояние при стратегии
                     $\beta$ 
          RB ← RB + 1 otherwise Рейс выполняется, если не было
                    перехода в критическое состояние при
                    стратегии  $\beta$ 
      SRA ← SRA + RA
      SRB ← SRB + RB
    Z ← RA - RB
    SZ ← SZ + Z
    D ← D + 1      Увеличиваем номер дня
  AVERA ←  $\frac{SRA}{Dmax}$ 
  AVERB ←  $\frac{SRB}{Dmax}$  Среднее число передач в день при стратегиях
                     $\alpha$  и  $\beta$  соответственно.
  AVERZ ←  $\frac{SZ}{Dmax}$  Средняя разница в числе передач
                    между стратегиями  $\alpha$  и  $\beta$  за один день.
  return (AVERA AVERB AVERZ)

```

Полученные результаты тестирования

Пример при вероятности перехода в ухудшенное и совсем плохое состояние =1

$$\text{Func}(1,1,100,4) = (50 \ 1 \ 49)$$

Пример при вероятности перехода в ухудшенное =1 и совсем плохое состояние =0

$$\text{Func}(1,0,100,4) = (50 \ 100 \ -50)$$

Пример при вероятности перехода в ухудшенное =0 и совсем плохое

$$\text{Func}(0,1,100,4) = (100 \ 100 \ 0)$$

Пример при вероятности перехода в ухудшенное =0 и совсем плохое

$$\text{Func}(0,0,100,4) = (100 \ 100 \ 0)$$

Дополнительные задания. Моделирование систем массового обслуживания

Задание 1. Стационарный режим, открытая СМО.

Задача 1 (По вариантам)

Имеется n телефонных линий. Вызов, пришедший в момент, когда все линии заняты, получает отказ. Интенсивность потока вызовов в минуту равна L . Средняя продолжительность одного разговора равна t минут. Определить основные характеристики данной системы массового обслуживания в стационарном режиме.

ЗАМЕЧАНИЕ. Эта СМО без очереди, в отличие от рассмотренного ниже демонстрационного примера. Внимательно прочтите теорию!

Таблица 7.1

Номер варианта	n	L	t	Номер варианта	n	L	t	Номер варианта	n	L	t
1	1	0,8	1,5	13	2	0,7	1,5	25	3	0,85	1,5
2	10	0,6	1,6	14	9	0,65	1,6	26	7	0,65	1,1
3	2	0,8	1,5	15	12	0,8	1,3	27	6	0,8	1,1
4	8	0,7	1,4	16	8	0,6	1,4	28	8	0,7	1,6
5	1	0,5	1,8	17	1	0,5	1,4	29	2	0,3	1,4
6	10	0,6	1,4	18	5	0,6	1,4	30	10	0,7	1,5

7	5	0,2	2	19	5	0,3	2	31	5	0,4	2
8	4	0,3	1,9	20	4	0,3	2	32	4	0,5	1,4
9	7	0,2	1,3	21	8	0,5	1,3	33	7	0,4	1,3
10	6	0,1	1,5	22	7	0,5	1,5	34	6	0,2	1,6
11	4	0,3	1,5	23	5	0,3	1,6	35	4	0,4	1,3
12	5	0,4	1,2	24	4	0,3	1,3	36	5	0,5	1,2

Задание 2. Нестационарный режим, замкнутая СМО с очередью

Задача 2 (По вариантам)

n рабочих обслуживают k станков. Каждый станок останавливается в среднем 2 раза в час. Процесс наладки одного станка занимает у рабочего в среднем 10 мин. Определить:

1. предельные вероятности состояний данной системы;
2. промежуток времени, в течение которого система переходит в стационарный режим;
3. среднее число неисправных станков.

Таблица 7.2

Номер варианта	n	k	Номер варианта	n	k	Номер варианта	n	k
1	1	3	13	2	6	25	3	7
2	2	6	14	3	4	26	3	19
3	3	8	15	5	8	27	2	17
4	4	10	16	4	10	28	4	15
5	5	12	17	1	12	29	2	11
6	1	6	18	5	14	30	5	10
7	2	10	19	5	16	31	5	8
8	3	14	20	4	18	32	4	20
9	4	16	21	3	20	33	3	16
10	5	20	22	2	12	34	5	18
11	3	12	23	5	14	35	4	14
12	5	6	24	4	16	36	1	12

Задача 3 (Для всех вариантов)

Рассматривается круглосуточная работа пункта проведения профосмотра

танков подразделения, в котором **18** единиц боевой техники и **8** групп технического осмотра танков. На осмотр и выявление дефектов каждого танка затрачивается в среднем одной группой техосмотра **6** часов. На осмотр в среднем поступает **3** танка за **12** часов. Найти основные характеристики данной системы массового обслуживания. Определить также максимально возможную интенсивность поступления танков на профилактику с тем, чтобы среднее число машин в очереди не превышало **50%** от общего числа поступающих машин. Построить графики вероятностей нахождения системы в различных состояниях в зависимости от времени t .

3. РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

Контрольные вопросы

1. Модель и расчётная схема. Виды моделирования.
2. Математическое моделирование. Основные этапы и источники ошибок.
3. Необходимость тестирования компьютерной модели. Способы тестирования.
4. Сравнение некоторых пакетов инженерных расчётов и моделирования.
5. Понятие о системах с сосредоточенными и распределёнными параметрами.
6. Примеры систем, модель для которых описывается дифференциальными уравнениями
7. Сведение произвольной системы ОДУ к системе ОДУ I-го порядка в нормализованном виде Коши. Пример.
8. Получение модели колебательной системы со сосредоточенными параметрами на основе уравнений Лагранжа 2-го порядка.
9. Пример получения уравнений движения для двухмассовой системы с поступательным движением масс.
10. Пример получения уравнений, описывающих колебания механической системы со сосредоточенными параметрами при наличии вращательных степеней свободы.
11. Необходимость статистического анализа при моделировании технических объектов.
12. Модель «хищник-жертва».
13. Метод Какутани для решения граничных задач.
14. Моделирование конкурирующих стратегий.
15. Пример построения алгоритмической модели для одноканальной системы массового обслуживания и заявок без приоритета.
16. Моделирование процессов систем массового обслуживания.
17. Терминология для СМО. Понятие о простейшем потоке событий
18. Уравнения Колмогорова.
19. Открытые системы массового обслуживания.
20. Замкнутые системы массового обслуживания.

4. ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

УТВЕРЖДАЮ

Проректор по учебной работе

УТВЕРЖДАЮ

Проректор по учебной работе
Белорусского национального
технического университета


Ю.А. Николайчик

19.06.2024

Регистрационный № УД ФУТР 49-15 /уч.

КОМПЬЮТЕРНЫЕ МЕТОДЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности
6-05-0611-01 "Информационные системы и технологии"
профилизация "Информационные системы и технологии в проектировании
и производстве"**

Минск 2024 г.

Учебная программа составлена на основе образовательного стандарта ОСВО 6-05-0611-01-2023 и учебного плана специальности 6-05-0611-01 "Информационные системы и технологии" профилизация "Информационные системы и технологии в проектировании и производстве" (рег. №ФИТР 49д-4/уч. утв. 30.04.2024 г.).

СОСТАВИТЕЛЬ:

В. В. Напрасников, доцент кафедры «Программное обеспечение информационных систем и технологий» Белорусского национального технического университета, кандидат технических наук, доцент.

РЕЦЕНЗЕНТЫ:

А.Л. Кочеров, доцент кафедры управления информационными ресурсами Академии управления при Президенте Республики Беларусь, кандидат технических наук, доцент.

В.А. Казакевич, доцент кафедры «Информационные системы и технологии» Белорусского национального технического университета, кандидат физ. мат. наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой «Программное обеспечение информационных систем и технологий» Белорусского национального технического университета (протокол №___ от ___04___ 2024г.)

Заведующий кафедрой _____ Ю.В. Полозков

Методической комиссией факультета информационных технологий и робототехники Белорусского национального технического университета (протокол №___ от ___05___ 2024г.)

Председатель методической комиссии _____ С. В. Васильев

Научной библиотекой БНТУ _____ Т.И. Бирюкова

Научно-методическим советом Белорусского национального технического университета (протокол №___ секции №__ от _____ 2024г.)

• ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа учебной дисциплине «Компьютерные методы математического моделирования» разработана для специальности 6-05-0611-01 "Информационные системы и технологии" профилизации "Информационные системы и технологии в проектировании и производстве".

Целью изучения дисциплины является формирование навыков и знаний по использованию численных методов при расчетах и моделировании и разработке соответствующей программной поддержки для реализации этих методов.

Задачами изучения дисциплины являются:

1. Изучение взаимосвязи характера модели с упрощающими предположениями.
2. Изучение способов построения моделей в составе систем автоматизированного проектирования для конкретных классов технических объектов.
3. Изучение особенностей работы с моделями в составе систем автоматизированного проектирования разного класса при сведении их к последовательности простейших операций.

В результате освоения учебной дисциплины студент должен:

знать:

- взаимосвязь характера модели с упрощающими предположениями;
- способы построения моделей в составе систем автоматизированного проектирования для конкретных классов технических объектов;
- особенности работы с моделями в составе систем автоматизированного проектирования разного класса при сведении их к последовательности простейших операций;

уметь:

- выбирать методы решения уравнений, соответствующих моделям для разных объектов и явлений;
- создавать математические модели для систем со сосредоточенными параметрами;
- создавать специальные математические модели;

иметь навык:

- использования встроенных средств современных компьютерных систем для реализации различного класса математических моделей;
- создания собственных программных средств на основе современных компьютерных систем для реализации различного класса математических моделей.

Освоение данной учебной дисциплины обеспечивает формирование следующих компетенций:

СК-10. Применять современные программные средства для реализации различного класса математических моделей

Согласно учебному плану для очной формы получения высшего образования на изучение учебной дисциплины отведено всего 118 часов, в том числе 68 часов аудиторных занятий.

Распределение аудиторных часов по семестрам приведено в таблице 1.

Таблица 1

Очная форма получения высшего образования					
Курс	Семестр	Лекции.	Лабораторные занятия	Форма текущей аттестации	Форма промежуточной аттестации
2	4	34	34	Лабораторная работа	зачет

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Математическое моделирование и численные методы

Понятие о моделировании. Виды моделей. Математическое моделирование. Численные методы как средство математического моделирования. Источники погрешностей. Необходимость тестирования компьютерной модели. Способы тестирования компьютерной модели.

Тема 2. Начало работы в среде MATHCAD

Обзор возможностей программ MATHCAD, MATLAB, MAPLE. Особенности работы с программой MATHCAD. Простейшие приемы работы в MATHCAD. Основное меню. Основные палитры инструментов. Ввод формул, вычисление значений по формулам, отображение подсчитанных значений в различных форматах. Задание матриц и векторов, действия с квадратными матрицами, извлечение из матрицы строк и столбцов. Графика в MATHCAD.

Тема 3. Численные методы решения нелинейных уравнений

Отделение корня уравнения и уточнение корня уравнения. Метод бисекции. Метод хорд. Метод касательных. Метод секущих. Метод простых итераций. Особенности реализации численных методов решения нелинейных уравнений в MATHCAD. Встроенные процедуры решения нелинейных уравнений в MATHCAD.

Тема 4. Численные методы решения систем линейных алгебраических уравнений

Решение систем линейных алгебраических уравнений методом Гаусса. Решение систем линейных алгебраических уравнений итерационными методами. Решение переопределенных систем линейных алгебраических уравнений. Решение систем линейных алгебраических уравнений в MATHCAD.

Тема 5. Моделирование и численные методы решения обыкновенных дифференциальных уравнений (ОДУ)

Примеры задач, приводящих к ОДУ. Модели, описывающие колебания мобильных транспортных средств, на примере движения автомобиля по дороге с заданным микропрофилем и подрессоренным грузом на кузове. Механические системы с сосредоточенными параметрами. Модель «хищник-жертва». Классификация методов решения задачи Коши. Одношаговые и многошаговые, явные и неявные методы. Сведение произвольной системы ОДУ к системе ОДУ первого порядка. Примеры. Метод Эйлера и его модификации для решения ОДУ. Методы Рунге-Кутты. Встроенные процедуры решения систем ОДУ в MATHCAD. Особенности описания правых частей в системе ОДУ в виде Коши в системе MATHCAD. Реализация метода Рунге-Кутты на встроенном языке MATHCAD.

Методы прогноза и коррекции для решения ОДУ. Понятие о граничной задаче для ОДУ. Задача об изгибе балки переменного сечения, как пример граничной задачи для ОДУ. Решение граничных задач методом стрельб. Метод стрельбы для решения граничных задач. Реализация метода стрельб в среде MATHCAD.

Тема 6. Моделирование и методы типа Монте-Карло

Исторические корни возникновения методов типа Монте – Карло. Общая идея моделирования на основе методов типа Монте-Карло. Пример задачи Бюффона как моделирования детерминированной задачи с использованием поведения случайных величин. Скорость сходимости методов Монте – Карло. Равномерное распределение как основа моделирования случайных величин с другими законами распределения. Примеры формул для моделирования случайных величин с разными законами распределения. Использование случайных величин в задачах вычисления площадей и объемов в многомерном пространстве. Преимущества методов Монте – Карло для решения подобных задач. Метод Судзуо Какутани для решения граничных задач теории потенциала как пример метода решения детерминированной задачи. Понятие о граничной задаче. Определяющие уравнения и граничные условия. Примеры различных процессов, описываемых эллиптическими уравнениями. Случайные блуждания при моделировании температурных полей. Алгоритм метода. Особенности создания собственных процедур в MATHCAD, использующих идеи методов Монте-Карло.

Тема 7. Метод наихудшего случая и статистический анализ

Исследование работоспособности объекта на основе математической модели. Необходимость статистического анализа при моделировании технических объектов. Метод наихудшего случая на примере задачи определения напряженно-деформированного состояния резца токарного станка в упрощенной постановке. Анализ чувствительности. Матрица чувствительности. Метод приращений для получения матрицы чувствительности, его достоинства и недостатки. Статистический анализ работоспособности технического объекта на основе математической модели.

Тема 8. Моделирование конкурирующих стратегий

Понятие о конкурирующих стратегиях. Преимущества алгоритмических моделей в задачах определения рационального подхода при выборе вариантов периодичности ремонтов станочного парка. Схема для программной реализации двух простейших стратегий эксплуатации станка. Реализация в среде MATHCAD.

Тема 9. Моделирование систем массового обслуживания (СМО)

Моделирование систем массового обслуживания (СМО). Общая структура СМО. Критерии качества и цели моделирования СМО. Алгоритмическая модель простейшей системы массового обслуживания. Особенности программной реализации моделей СМО. Марковские процессы. Состояние СМО. Вероятности состояний СМО. Открытые СМО. Замкнутые СМО. Уравнения Колмогорова. Размеченный граф СМО. Стационарный режим как предельный режим функционирования СМО. Уравнения для стационарного режима. Реализация моделей СМО в среде MATHCAD.

Тема 10. Методы приближения инженерных и экономических данных

Постановка задачи о вычислении геометрических характеристик плоских сечений. Связь с общей задачей по приближению данных. Различные виды приближений. Отличие интерполяции и аппроксимации. Использование интерполяции при выполнении инженерных расчетов. Различные виды интерполяции. Особенности применения встроенных процедур MATHCAD для приближения данных на основе интерполяции. Сплайны - современный математический аппарат для приближения данных. Исторические корни возникновения сплайнов. Общие сведения. Определение сплайна. Сплайны первой степени. Эрмитовы кубические сплайны. Алгоритм построения эрмитова кубического сплайна по табличным экспериментальным данным с использованием приближенной замены производных на основе разностных представлений. Особенности создания собственных процедур в MATHCAD для приближения данных на основе интерполяции сплайнами.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
очная форма получения высшего образования

Номер раздела, темы	Название раздела, темы, занятия	Количество аудиторных часов			Количество часов СР	Форма контроля знаний
		Лекции	Практические занятия	Лабораторные занятия		
1	2	3	4	6	8	9
	4 семестр					
1.	Математическое моделирование и численные методы	2				
2.	Начало работы в среде MATHCAD	2				
	Лабораторные занятия №1. Основные операции в среде MATHCAD			4		Защита лабораторной работы
3.	Численные методы решения нелинейных уравнений	2				Защита лабораторной работы
	Лабораторные занятия №2. Методы решения нелинейных уравнений			2		Защита лабораторной работы
4.	Численные методы решения систем линейных алгебраических уравнений	2				
	Лабораторные занятия №3. Методы решения систем линейных алгебраических уравнений			2		Защита лабораторной работы
5.	Моделирование и численные методы решения обыкновенных дифференциальных уравнений (ОДУ)	8				
	Лабораторные занятия №4. Метод Эйлера, метод Рунге-Кутты. Модели технических и экологических систем. Метод пристрелки в моделировании.			8		Защита лабораторной работы
6.	Моделирование и методы типа Монте-Карло	4				

	Лабораторные занятия №5. Вычисление площадей и интегралов			4		Защита лабораторной работы
7.	Метод наилучшего случая и статистический анализ	2				
	Лабораторные занятия №6. Метод наилучшего случая и статистический анализ			2		Защита лабораторной работы
8.	Моделирование конкурирующих стратегий	2				
	Лабораторные занятия №7. Выбор стратегии эксплуатации автобуса			2		Защита лабораторной работы
9.	Моделирование систем массового обслуживания (СМО)	4				
	Лабораторные занятия №8 Модели систем массового обслуживания			6		Защита лабораторной работы
10.	Методы приближения инженерных и экономических данных	6				
	Лабораторные занятия №9. Интерполяция и аппроксимация			4		Защита лабораторной работы
	Итого за семестр	34		34		зачет
	Всего аудиторных часов		68			

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Электронный учебно-методический комплекс «Компьютерное моделирование прикладных задач» для специальностей: I – 40 01 01 «Программное обеспечение информационных технологий» 1-40 05 01 «Информационные системы и технологии» [Электронный ресурс] / Белорусский национальный технический университет, Кафедра «Информационные системы и технологии»; сост.: В. В. Напрасников, И. Л. Ковалева. – Минск : БНТУ, 2022 <https://rep.bntu.by/handle/data/125146>
2. Алгоритмическая реализация численных методов [Электронный ресурс] : учебно-методический комплекс для специальности 1-31 81 12 "Прикладной компьютерный анализ данных" / Белорусский национальный технический университет, Кафедра "Системы автоматизированного проектирования"; сост.: В. М. Волков, И. Л. Ковалева, В. В. Напрасников. – Минск : БНТУ, 2018. <https://rep.bntu.by/handle/data/48295>
3. Волков, В.М. Численные методы для обыкновенных дифференциальных уравнений [Электронный ресурс]: учебно-методическое пособие для магистрантов специальности 1-31 81 12 "Прикладной компьютерный анализ данных" / В.М. Волков, Ирина Львовна Ковалева; кол. авт. Белорусский национальный технический университет, Кафедра "Системы автоматизированного проектирования". – Электрон. дан. – Минск: БНТУ, 2016. – 1 электрон. опт. диск (CD-ROM). – Режим доступа: <https://rep.bntu.by/handle/data/27855>.
4. Краков, Михаил Самуилович. Численные методы и обработка данных: пособие / М. С. Краков, С. Г. Погирницкая; Министерство образования Республики Беларусь, Белорусский национальный технический университет, Кафедра ЮНЕСКО "Энергоснабжение и возобновляемые источники энергии". – Минск: БНТУ, 2021. – 86, [1] с.: ил., табл., схемы. – Режим доступа : <https://rep.bntu.by/handle/data/102987>.
5. Компьютерное конечно-элементное моделирование : пособие для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» направления специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)» : в 2 ч. / В. В. Напрасников [и др.]. – Минск : БНТУ, 2021. – Ч. 1. – 83 с.
6. Компьютерное конечно-элементное моделирование : пособие для студентов специальности 1-40 05 01 «Информационные системы и

- технологии (по направлениям)» направления специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)» : в 2 ч. / В. В. Напрасников [и др.] ; Белорусский национальный технический университет, Кафедра «Программное обеспечение информационных систем и технологий». – Минск : БНТУ, 2021. – Ч. 2. – 79 с.
7. Мулярчик, С.Г. Численные методы: учебное пособие для студентов учреждений высшего образования по специальностям "Радиофизика", "Физическая электроника", "Компьютерная безопасность (по направлениям)", "Прикладная информатика (по направлениям)" / С.Г. Мулярчик. – Минск: РИВШ, 2017. – 317 с.: ил.
 8. Кремень, Е.В. Численные методы: практикум в MathCad: учебное пособие для студентов учреждений высшего образования по математическим специальностям / Е. В. Кремень, Ю. А. Кремень, Г. А. Расолько. – Минск: Вышэйшая школа, 2019. – 254, [1] с.: ил., табл.
 9. Напрасников, В. В. Оптимизационные расчеты на основе командного файла в ANSYS : учебно-методическое пособие / В. В. Напрасников, Ю. В. Напрасникова. – Минск: БНТУ, 2014. – 20 с.

Дополнительная литература

1. Информационные технологии в автоматизированном проектировании машиностроительных конструкций: учебно-методическое пособие для студентов специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)»: в 3 ч. / А. В. Бородуля [и др.]; Белорусский национальный технический университет, Кафедра «Программное обеспечение информационных систем и технологий». – Минск: БНТУ, 2020. – Ч. 1. – 98 с.
2. Напрасников, В. В. Построение конечно-элементной модели на основе языка APDL : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2009. – 51 с.
3. Напрасников, В. В. Создание конечно-элементной модели для расчета контейнера в процессе прессования порошковой заготовки : лабораторный практикум / В. В. Напрасников [и др.]. – Минск: БНТУ, 2008. – 89 с.
4. Напрасников, В. В. Конечно-элементное моделирование в ANSYS в режиме удаленного доступа к суперкомпьютеру «СКИФ»: учебно-методическое пособие / В. В. Напрасников, А. В. Бородуля, В. А. Кочуров. – Минск: БНТУ, 2008. – 65 с.

5. Напрасников, В. В. Моделирование колебаний рамной конструкции на основе метода конечных элементов : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2010. – 43 с.
6. Напрасников, В. В. Создание 3D конечно-элементной модели в среде ANSYS : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2010. – 37 с.
7. Напрасников, В. В. Расчет температурных полей на основе конечно-элементного моделирования : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2011. – 22 с.
8. Напрасников, В. В. Особенности расчета конструкций с шарнирами в ANSYS / В. В. Напрасников, Ю. В. Напрасникова, А. Н. Соловьев. – Минск: БНТУ, 2013. – 37 с

Средства диагностики результатов учебной деятельности

Оценка уровня знаний студента производится по десятибалльной шкале в соответствии с критериями, утвержденными Министерством образования Республики Беларусь.

Для оценки достижений студента используется следующий диагностический инструментарий:

- защита выполненных на лабораторных занятиях работ;
- собеседование при проведении индивидуальных и групповых консультаций;
- сдача зачета по дисциплине.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов

1. Понятие о моделировании.
2. Модель и расчётная схема. Виды моделирования.
3. Математическое моделирование. Основные этапы и источники ошибок.
4. Необходимость тестирования компьютерной модели. Способы тестирования.
5. Сравнение некоторых пакетов инженерных расчётов и моделирования.
6. Понятие о системах с сосредоточенными и распределёнными параметрами.
7. Примеры систем, модель для которых описывается дифференциальными уравнениями

8. Сведение произвольной системы ОДУ к системе ОДУ I-го порядка в нормализованном виде Коши. Пример.
9. Получение модели колебательной системы со сосредоточенными параметрами на основе уравнений Лагранжа 2-го порядка.
10. Пример получения уравнений движения для двухмассовой системы с поступательным движением масс.
11. Пример получения уравнений, описывающих колебания механической системы со сосредоточенными параметрами при наличии вращательных степеней свободы.
12. Классификация методов решения задачи Коши для ОДУ. Одношаговые и многошаговые, явные и неявные методы.
13. Метод Эйлера и его модификации для решения ОДУ.
14. Основная идея методов Рунге-Кутты для решения ОДУ.
15. Метод Рунге-Кутты 4-го порядка точности для решения ОДУ и его реализация в MathCad.
16. Способ Рунге для оценки погрешности ОДУ.
17. Методы прогноза и коррекции для решения ОДУ.
18. Понятие о жестких дифференциальных уравнениях.
19. Метод стрельбы для решения граничных задач для ОДУ.
20. Задача Бюффона, как пример использования методов Монте-Карло при решении детерминированных задач.
21. Использование методов Монте-Карло для вычисления площадей и объёмов.
22. Метод Какутани для решения граничных задач.
23. Моделирование конкурирующих стратегий.
24. Пример построения алгоритмической модели для одноканальной системы массового обслуживания и заявок без приоритета.
25. Моделирование процессов систем массового обслуживания.
26. Терминология для СМО. Понятие о простейшем потоке событий
27. Уравнения Колмогорова.
28. Приближение инженерных данных
29. Виды приближение инженерных данных. Аппроксимация, интерполяция, равномерное приближения.
30. Разложение аппроксиматора по системы базисных функций.
31. Интегральная квадратичная аппроксимация на отрезке.
32. Пример аппроксимации функции 2-х переменных в MATHCAD, значения которой заданы таблично.
33. Алгебраическое интерполирование для приближения инженерных данных.
34. Интерполяционный полином Лагранжа.
35. О наилучшем выборе узлов интерполирования.

36. Тригонометрическая интерполяция.
37. Понятие о сплайнах.
38. Области использования интерполирования.
39. Эрмитов кубический сплайн.
40. Кубический сплайн дефекта 1. Алгоритм его построения.
41. Решение системы уравнений, определяющих сплайн, методом прогонки.
42. Обобщенные кубические сплайны.
43. Параметрические сплайны для приближения кривых.
44. Открытые системы массового обслуживания.
45. Замкнутые системы массового обслуживания.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

При изучении дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- подготовка сообщений, тематических докладов, презентаций по заданным темам;
- составление тематической подборки литературных источников, интернет-источников;
- проработка тем (вопросов), вынесенных на самостоятельное изучение.

<p>Название учебной дисциплины, с которой требуется согласование</p>	<p>Название кафедры</p>	<p>Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине</p>	<p>Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола заседания кафедры)</p>
<p>Согласование не требуется</p>	<p>Кафедра «Программное обеспечение информационных систем и технологий»</p>		<p>Содержание данной учебной программы не требует согласования с другими учебными дисциплинами специальности. Протокол № _ от 04. .2024</p>