

Федеральное агентство по образованию
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГОУВПО "АмГУ"
Факультет математики и информатики

УТВЕРЖДАЮ

Зав. кафедрой МАиМ

_____ Т.В. Труфанова

«__» _____ 2007 г.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
МЕХАНИКИ СПЛОШНЫХ СРЕД

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

для специальности 010501 – "Прикладная математика и информатика"

Составитель: А.В. Рыженко

Благовещенск

2007 г.

ББК

*Печатается по решению
редакционно-издательского
совета
факультета математики и
информатики
Амурского государственного
университета*

Рыженко А.В.

Математическое моделирование механики сплошных сред:
Учебно-методический комплекс по дисциплине для студентов АмГУ очной
формы обучения специальности 010501 "Прикладная математика и
информатика". – Благовещенск: Амурский гос. ун-т, 2007. – 171 с.

Учебно-методический комплекс по дисциплине "Математическое моделирование механики сплошных сред" предназначен для студентов специальности 010501 – "Прикладная математика и информатика" очной формы обучения, призван помочь ведущим преподавателям и студентам в организации процесса изучения дисциплины. Комплекс содержит рабочую программу дисциплины, план-конспект лекций, материалы для проведения семинарских занятий и лабораторных работ, контролирующие материалы для осуществления промежуточного и итогового контроля, справочный материал и библиографический список.

© Амурский государственный университет, 2007

© Кафедра математического анализа и моделирования, 2007

I. РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Рабочая программа по дисциплине "Математическое моделирование механики сплошных сред" для специальности 010501 – "Прикладная математика и информатика".

Курс 4, 5. Семестр 7,8,9. Лекции 98 (32+32+34) час. Экзамен 8,9 семестр. Практические (семинарские) занятия 49 (16+16+17) час. Зачет 7 семестр. Лабораторные занятия 49 (16+16+17) час. Курсовая работа 9 семестр. Самостоятельная работа 64 час. Всего 260 часов.

Составитель Г.С. Михайлов, к.ф.-м.н., доцент. Факультет математики и информатики. Кафедра математического анализа и моделирования. Благовещенск, 2006 г.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

Рабочая программа дисциплины "Математическое моделирование механики сплошных сред" составлена на основе авторских разработках и соответствует основным требованиям близких по тематике государственных образовательных стандартов высшего профессионального образования.

1.1. Цель преподавания дисциплины.

Цель курса – ознакомление с общими принципами построения математических моделей, технологией решения задач на ЭВМ, а также изучение и освоение методов исследования и анализа математических моделей.

1.2. Перечень дисциплин с указанием разделов (тем), усвоение которых студентами необходимо при изучении данной дисциплины.

Преподавание настоящей дисциплины связано с курсами классической математики, численных методов, методов решения обыкновенных и стохастических дифференциальных уравнений, методов оптимизации,

механики сплошных сред, современных информационных технологий и опирается на их содержание.

1.3. Задачи изучения дисциплины.

По завершении изучения дисциплины студент должен: знать и владеть технологической цепочкой проведения вычислительного эксперимента на ЭВМ; уметь применять для анализа и исследования конкретной математической модели общий МВН или его варианты МКР, МКЭ, МКО.

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Наименование тем, их содержание, объем в часах лекционных занятий.

7 семестр

Тема 1. Предмет математического моделирования. Формальная технология моделирования. Формулировка математической технологии моделирования. Общие принципы построения математических моделей. (4 часа)

Тема 2. Математический аппарат моделей, основанный на законах сохранения. Примеры математических моделей неживой, живой и мыслящей материи. (6 часов)

Тема 3. Характерные черты системного анализа. Имитационные системы – основные положения. Компромиссы Парето для оценки возможных стратегий. (4 часа)

Тема 4. Методы анализа математических моделей. Способы аппроксимации непрерывных математических моделей. Конечно-разностные аппроксимации и схемы интегрирования. Решения дифференциальных уравнений методом конечных разностей (МКР). (8 часов)

Тема 5. Основные положения метода взвешенных невязок (МВН). Аппроксимация с помощью МВН. Аппроксимация решений

дифференциальных уравнений МВН с использованием базисных функций. Одновременная аппроксимация решений дифференциальных уравнений и краевых условий. Слабая формулировка МВН и естественные краевые условия. Методы граничного решения в МВН. (10 часов)

8 семестр

Тема 6. Понятие конечного элемента. Локально определенные базисные функции с минимальными носителями. КЭ аппроксимация решений дифференциальных уравнений и требование гладкости. Кусочно-линейные базисные функции и методология ансамблирования. (6 часов)

Тема 7. Концепция аналогии в стандартных дискретных инженерных системах. Обобщение МКЭ на двумерные и трехмерные задачи. Двумерные и трехмерные КЭ линейного типа. Моделирование двумерной теплопроводности треугольными и прямоугольными КЭ. (8 часов)

Тема 8. Сопоставление аппроксимации конечными разностями и конечными элементами. (2 часа)

Тема 9. КЭ аппроксимации высших порядков. Степень многочленов базисных функций и скорость сходимости. Кусочное тестирование. (3 часа)

Тема 10. Стандартные базисные функции высших степеней для одномерных КЭ. (2 часа)

Тема 11. Иерархические формы высших степеней для одномерных КЭ. Иерархические многочлены. Иерархические многочлены почти ортогональной формы. (6 часов)

Тема 12. Двумерные базисные функции высших степеней для лагранжевых и сирендиповых КЭ. Иерархические базисные функции для прямоугольных и треугольных КЭ. (5 часов)

9 семестр

Тема 13. Понятие отображения областей. Параметрическое отображение. Процедуры численного интегрирования при отображении. (6 часов)

Тема 14. Отображение составной функцией. Отображение решением дополнительных уравнений. Отображение бесконечных областей. (6 часов)

Тема 15. Частичная дискретизация для нестационарных задач. Использование процедур аналитических решений для линейных задач: однородное уравнение 2-го и 1-го порядков, разложение решения по собственным векторам. (7 часов)

Тема 16 КЭ аппроксимации решения нестационарных задач во временной области: уравнения 1-го и 2-го порядков. Анализ конкретных схем интегрирования. (8 часов)

Тема 17. Обобщенный МКЭ. Погрешность дискретизации в численном решении. Мера погрешности дискретизации. Оценка, погрешности дискретизации. (7 часов)

2.2. Практические и семинарские занятия, их содержание и объем в часах.

Перечень темы практических занятий по семестрам:

7 семестр

1. Моделирование одномерных физических процессов МКР. (4 часа)
2. Моделирование двумерных физических процессов МКР. (4 часа)
3. Анализ нелинейных физических процессов. (4 часа)
4. Интерполяция МВН: метод коллокаций, метод Галеркина, метод наименьших квадратов. (4 часа)

8 семестр

1. Решение краевых задач МВН: ВН по расчетной области; ВН по расчетной области и по границе, ВН по границе. (4 часа)
2. МКЭ для решения краевых задач кусочно-линейными базисами функциями: одномерные и двумерные процессы. (6 часов)
3. Базисные функции высших степеней для одномерных КЭ. (2 часа)
4. Иерархические базисные функции высших степеней для одномерных КЭ. (4 часа)

9 семестр

1. Лагранжевы и сирендиповы двумерные базисные функции высших степеней: прямоугольные КЭ, треугольные КЭ. (4 часа)
2. Иерархические двумерные базисные функции (4 часа)
3. Процедуры отображения. (2 часа)
4. Частичная дискретизация для нестационарных задач. (5 часов)
5. Оценка погрешности дискретизации. (2 часа)

2.3. Лабораторные занятия, их наименование и объем в часах.

7 семестр

1. Разработка программы решения СЛАУ методом прогонки. (3 часа)
2. Моделирование смешанной задачи теплопроводности явной и неявной КР схемами. (4 часа)
3. Моделирование нелинейной краевой задачи МКР. (4 часа)
4. Аппроксимация решения краевой задачи МВН. (5 часов)

8 семестр

1. МВН в слабой формулировке для решения двумерных краевых задач. (4 часа)
2. Простейшие базисные функции для КЭ аппроксимации линейных и нелинейных задач. (4 часа)
3. Базисные функции высших степеней для КЭ аппроксимации одномерных краевых задач. (4 часа)
4. Билинейные базисные функции в КЭ моделировании двумерных задач. (4 часа)

9 семестр

1. КЭ моделирование распределения тепла иерархическими базисными функциями в одномерных изотропных элементах конструкций. (4 часа)
2. КЭ моделирование распределения тепла иерархическими базисными функциями в одномерных композитных элементах конструкций. (5 часов)
3. Моделирование двумерного упругого кручения прямоугольного бруса четырехугольными и треугольными иерархическими КЭ. (8 часов)

2.4. Курсовая работа.

Разработка ММ, алгоритма и программной реализации построения двумерных сеток алгебраическими отображениями.

ММ, алгоритм и программная реализация МКО для решения простейших задач двумерной гидрогазодинамики.

ММ построения адаптивных сеток методом эквираспределения и программная реализация.

ММ интерактивной процедуры построения структурных сеток и ее реализация на двумерных областях.

ММ и алгоритм реализации построения адаптивных расчетных сеток из ячеек Дирихле.

Математическое моделирование течения вязкой жидкости МКО на треугольных сетках.

Математическое моделирование нестационарных течений МКО.

Математическое моделирование течений вязкой жидкости МКО на компактных конечно-разностных схемах

Включение процедур расщепления в КО и КЭ методологии решения задач математической физики.

2.5. Самостоятельная работа студентов.

1. Метод конечных объемов (МКО) для математического моделирования задач математической физики.

2. Использование неортогональных, неструктурированных сеток для аппроксимации областей со сложной геометрией.

3. Основные этапы аппроксимации в МКО.

4. КО дискретизация основных уравнений: триангуляция и способ построения двойственной сетки; интегральная форма законов сохранения; аппроксимация потоковых слагаемых, источниковых слагаемых, матрицы массы КО; граничных условий.

5. Формирование итоговой системы алгебраических уравнений.

2.6. Перечень и темы промежуточных форм контроля знаний

В рамках практических занятий предусмотрен промежуточный и итоговый контроль. Название тем контрольных работ по семестрам:

1. Конечно-разностный метод аппроксимации – 7 семестр.
2. Аппроксимация решений дифференциальных уравнений МВН – 8 семестр.
3. КЭ методология решения на одномерных и двумерных задачах – 9 семестр.

2.7. Вопросы к зачету (7 семестр)

1. Принцип построения математических моделей и математическая технология моделирования. Компромиссы Парето.
2. Конечно-разностные аппроксимации и схемы интегрирования.
3. Виды МВН и способы их применения для решения конкретных задач.
4. Решения задач по перечисленным темам.

2.8. Вопросы к экзамену (8 семестр)

1. Предмет математического моделирования и общие принципы построения математических моделей.
2. Математический аппарат моделей, основанный на законах сохранения.
3. Математическая технология моделирования.
4. Конечно-разностная дискретизация и аппроксимация производных в одномерном случае.
5. Решение одномерных дифференциальных уравнений МКР на согласованных схемах.
6. МКР на двумерных областях правильной формы.
7. МКР на двумерных областях с криволинейной границей.
8. Основные положения и виды метода взвешенных невязок (МВН).
9. Метод Галеркина и наименьших квадратов в МВН.
10. Аппроксимация решений дифференциальных уравнений МВН.
11. Одновременная аппроксимация решений дифференциальных уравнений и краевых условий МВН.

12. Формулировка естественных краевых условий в МВН.
13. МВН с естественными краевыми условиями в двумерных задачах (теплопроводность).
14. Методы граничного решения в МВН.
15. Аппроксимация решения системы дифференциальных уравнений МВН.

Вопросы к экзамену (9 семестр)

1. Моделирование двумерных плоских задач теории упругости в напряжениях МВН.
2. Понятие конечного элемента (КЭ) и локальное определение базисных функций с минимальными носителями.
3. КЭ аппроксимация решений дифференциальных уравнений и требования гладкости.
4. Слабая формулировка МВН и кусочно-линейные базисные функции в одномерных краевых задачах.
5. Обобщение МКЭ на двумерные задачи билинейными базисными функциями.
6. МКЭ для двумерных задачах теплопроводности на произвольных линейных треугольниках.
7. Моделирование двумерных задач теории упругости в напряжениях линейными треугольниками.
8. Моделирование двумерных задачи теплопроводности МКЭ в виде билинейных прямоугольников.
9. КЭ аппроксимации высших порядков. Степень многочленов базисных функций и скорость сходимости.
10. Стандартные базисные функции высших степеней для одномерных КЭ.
11. Иерархические базисные функции высших степеней для одномерных КЭ.
12. Двумерные базисные функции высших степеней для Лагранжевых КЭ.
13. Двумерные базисные функции высших степеней для сирендиповых КЭ.
14. Иерархические базисные функции высших степеней для прямоугольных КЭ.

15. Иерархические базисные функции высших степеней для треугольных КЭ.

3. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

3.1. Перечень обязательной (основной) литературы.

1. Белолипецкий В.М., Шокин Ю.И. Математическое моделирование в задачах охраны окружающей среды. – Новосибирск, изд-во "ИНФОЛИО-пресс", 1997.
2. Краснощеков П.С., Петров А.А, Принципы построения моделей – М.: изд-во МГУ, 1983.
3. Математическое моделирование / Под ред. Дж. Эндрюса, Р. Мак-Лоуна - М.: Мир, 1979.
4. Самарский А.А., Михайлов А.П. Математическое моделирование. Идеи. Методы. Примеры. – М.: Физматлит, 2005. – 320 с.

3.2. Перечень дополнительной литературы.

5. Белоцерковский О.М. Численное моделирование в механике сплошной среды. – М.: Наука, 1984.
6. Емельянов В.В., Курейчик В.Н., Курейчик В.В. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 432 с.
7. Журналы РАН: Математическое моделирование, Вычислительные технологии, Журнал вычислительной математики и математической физики.
8. Зенкевич О., Морган К. КЭ и аппроксимация. – М.: Мир. 1986.
9. Зенкевич О. Метод КЭ в технике. – М.: Мир, 1975.
10. Искусственные иммунные системы и их применение / Под ред. Д. Дасгупты, пер. с англ. Под ред. А.А. Романюхи. – М.: Физматлит, 2006. – 344 с.
11. Кондратьев К.Я., Крапивин В.Ф. Моделирование глобального круговорота углерода. – М.: Физматлит, 2004. – 336 с.

- 12.Ковеня В.М., Яненко Н.Л. Метод расщепления в задачах газовой динамики. – Новосибирск, Наука, Сиб. отд-ие, 1981.
- 13.Ковеня В.М., Тарковский Г.А., Черный С.Г. Применение метода расщепления в задачах аэродинамики. – Новосибирск, Наука, 1990.
- 14.Моисеев Н.Л. Математика ставит эксперимент. – Н.: Наука, 1979.
- 15.Моделирование и управление процессами регионального развития / Под ред. С.Н. Васильева. – М.: Физматлит, 2001. – 432 с.
- 16.Математические модели природы и общества / Н.Н. Калиткин, Н.В. Карпенко, А.П. Михайлов, В.Ф. Тимкин, М.В. Черненко. – М.: Физматлит, 2005. – 360 с.
17. Михайлов А.П. Моделирование системы «власть - общество». – М.: Физматлит, 2006. – 144 с.
- 18.Смолин Д.В. Введение в искусственный интеллект: конспект лекций. – М.: Физматлит, 2004. – 208 с.
- 19.Тарко А.М, Антропогенные изменения глобальных биосферных процессов. Математическое моделирование. – М.: Физматлит, 2005. – 232 с.
- 20.Форрестер Дж. Мировая динамика. - М., Наука, 1978.
- 21.Фалин Г.И., Фалин А.И. Актуарная математика в задачах. – М.: Физматлит, 2003.-192с.
- 22.Флетчер К. Вычислительные методы в динамике жидкостей. - М.: Мир, т. 1,2, 1991.
- 23.Хренников А.Ю. Моделирование процессов мышления в p-адических координатах. – М.: Физматлит, 2004. – 296 с.
- 24.Чуличков А.И. Математические модели нелинейной динамики. – М.: Физматлит, 2003. – 296 с.
- 25.Шеннон Р. Имитационное моделирование систем – искусство и наука – М.: Мир, 1978.

3.3. Перечень наглядных и иных пособий.

1. Карточки с заданиями к лабораторным работам / *А.В. Рыженко*.

3.4. Средства обеспечения освоения дисциплины.

1. Один из перечисленных пакетов прикладных программ: MathCad фирмы Mathsoft Engineering & Education, Inc.; MatLab фирмы The MathWorks, Inc.; Matematica 5 фирмы Wolfram Research, Inc.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс кафедры МАиМ.

5. КРИТЕРИИ ОЦЕНКИ ЗНАНИЙ

Экзамен или зачет сдаются в конце семестра. Форма сдачи – устная. Необходимым допуском на экзамен (зачет) является сдача всех лабораторных работ, положительные оценки за промежуточные контрольные работы. В предлагаемый билет входят три вопроса: два теоретических и один практический, на которые студент должен дать развернутый ответ. Показать полное знание теории по данной части курса, продемонстрировать свободную ориентацию в материале, знание понятий и терминологии, ответить на дополнительные вопросы, уметь объяснить выполнение упражнения (практического задания).

II. ГРАФИК САМОСТОЯТЕЛЬНОЙ УЧЕБНОЙ РАБОТЫ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ НА КАЖДЫЙ СЕМЕСТР С УКАЗАНИЕМ ЕЕ СОДЕРЖАНИЯ, ОБЪЕМА В ЧАСАХ, СРОКОВ И ФОРМ КОНТРОЛЯ

График самостоятельной учебной работы студентов по дисциплине на каждый семестр (с указанием ее содержания, объема в часах, сроков и форм контроля) приведен в рабочей программе дисциплины.

III. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ ЗАНЯТИЙ. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ (ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ)

Форма проведения лабораторного занятия: а) приветствие студентов, 1 мин.; б) определение личного состава студенческой группы, 4 мин.; в) объявление тематики и вопросов лабораторного занятия, 1 мин.; г) выполнение заданий, 70 мин.; д) подведение итогов лабораторного занятия, 13 мин.; е) прощание со студентами, 1 мин.

Список рекомендуемой литературы (основной и дополнительной) приведен в рабочей программе.

IV. МАТЕРИАЛЫ ДЛЯ ЧТЕНИЯ ЛЕКЦИЙ ИЛИ ПЛАН-КОНСПЕКТ

Лекция 1: Математическое моделирование. Форма и принципы представления математических моделей

ЭВМ прочно вошла в нашу жизнь, и практически нет такой области человеческой деятельности, где не применялась бы ЭВМ. ЭВМ сейчас широко используется в процессе создания и исследования новых машин, новых технологических процессов и поиске их оптимальных вариантов; при решении экономических задач, при решении задач планирования и управления производством на различных уровнях. Создание же крупных объектов в ракетотехнике, авиастроении, судостроении, а также проектирование плотин, мостов, и др. вообще невозможно без применения ЭВМ.

Для использования ЭВМ при решении прикладных задач, прежде всего прикладная задача должна быть "переведена" на формальный математический язык, т.е. для реального объекта, процесса или системы должна быть построена его *математическая модель*.

Слово "Модель" происходит от латинского *modus* (копия, образ, очертание). Моделирование – это замещение некоторого объекта А другим объектом Б. Замещаемый объект А называется оригиналом или объектом моделирования, а замещающий Б - моделью. Другими словами, модель – это объект-заменитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала.

Целью моделирования являются получение, обработка, представление и использование информации об объектах, которые взаимодействуют между собой и внешней средой; а модель здесь выступает как средство познания свойств и закономерности поведения объекта.

Моделирование широко используется в различных сферах человеческой деятельности, особенно в сферах проектирования и управления, где особенными являются процессы принятия эффективных решений на основе получаемой информации.

Модель всегда строится с определенной целью, которая оказывает влияние на то, какие свойства объективного явления оказываются существенными, а какие - нет. Модель представляет собой как бы проекцию объективной реальности под определенным углом зрения. Иногда в зависимости от целей можно получить ряд проекций объективной реальности, вступающих в противоречие. Это характерно, как правило, для сложных систем, у которых каждая проекция выделяет существенное для определенной цели из множества несущественного.

Теорией моделирования является раздел науки, изучающий способы исследования свойств объектов-оригиналов, на основе замещения их другими объектами-моделями. В основе теории моделирования лежит теория подобия. При моделировании абсолютное подобие не имеет места и лишь стремится к тому, чтобы модель достаточно хорошо отображала исследуемую сторону функционирования объекта. Абсолютное подобие может иметь место лишь при замене одного объекта другим точно таким же.

Все модели можно разделить на два класса:

1. вещественные,
2. идеальные.

В свою очередь вещественные модели можно разделить на:

1. натурные,
2. физические,
3. математические.

Идеальные модели можно разделить на:

1. наглядные,
2. знаковые,
3. математические.

Вещественные натурные модели – это реальные объекты, процессы и системы, над которыми выполняются эксперименты научные, технические и производственные.

Вещественные физические модели - это макеты, муляжи, воспроизводящие физические свойства оригиналов (кинематические, динамические, гидравлические, тепловые, электрические, световые модели).

Вещественные математические – это аналоговые, структурные, геометрические, графические, цифровые и кибернетические модели.

Идеальные наглядные модели - это схемы, карты, чертежи, графики, графы, аналоги, структурные и геометрические модели.

Идеальные знаковые модели – это символы, алфавит, языки программирования, упорядоченная запись, топологическая запись, сетевое представление.

Идеальные *математические модели* – это аналитические, функциональные, имитационные, комбинированные модели.

В приведенной классификации некоторые модели имеют двойное толкование (например - аналоговые). Все модели, кроме натуральных, можно объединить в один класс мысленных моделей, т.к. они являются продуктом абстрактного мышления человека.

Остановимся на одном из наиболее универсальных видов моделирования - математическом, ставящем в соответствие моделируемому физическому процессу систему математических соотношений, решение которой позволяет получить ответ на вопрос о поведении объекта без создания физической модели, часто оказывающейся дорогостоящей и неэффективной.

Математическое моделирование - это средство изучения реального объекта, процесса или системы путем их замены *математической моделью*, более удобной для экспериментального исследования с помощью ЭВМ.

Математическая модель является приближенным представлением реальных объектов, процессов или систем, выраженным в математических терминах и сохраняющим существенные черты оригинала. *Математические модели* в количественной форме, с помощью логико-математических конструкций, описывают основные свойства объекта, процесса или системы, его параметры, внутренние и внешние связи.

В общем случае *математическая модель* реального объекта, процесса или системы представляется в виде системы функционалов

$$\Phi_i(X, Y, Z, t) = 0,$$

где X – вектор входных переменных, $X = [x_1, x_2, x_3, \dots, x_N]^T$,

Y – вектор выходных переменных, $Y = [y_1, y_2, y_3, \dots, y_N]^T$,

Z – вектор внешних воздействий, $Z = [z_1, z_2, z_3, \dots, z_N]^T$,

t – координата времени.

Построение *математической модели* заключается в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно и качественно связь между теми или иными процессами и явлениями, между интересующими специалиста физическими величинами, и факторами, влияющими на конечный результат.

Обычно их оказывается настолько много, что ввести в модель всю их совокупность не удастся. При построении *математической модели* перед

исследованием возникает задача выявить и исключить из рассмотрения факторы, несущественно влияющие на конечный результат (*математическая модель* обычно включает значительно меньшее число факторов, чем в реальной действительности). На основе данных эксперимента выдвигаются гипотезы о связи между величинами, выражающими конечный результат, и факторами, введенными в *математическую модель*. Такая связь зачастую выражается системами дифференциальных уравнений в частных производных (например, в задачах механики твердого тела, жидкости и газа, теории фильтрации, теплопроводности, теории электростатического и электродинамического полей).

Конечной целью этого этапа является формулирование математической задачи, решение которой с необходимой точностью выражает результаты, интересующие специалиста.

Форма и принципы представления *математической модели* зависит от многих факторов.

По принципам построения *математические модели* разделяют на:

1. аналитические;
2. имитационные.

В аналитических моделях процессы функционирования реальных объектов, процессов или систем записываются в виде явных функциональных зависимостей.

Аналитическая модель разделяется на типы в зависимости от математической проблемы:

1. уравнения (алгебраические, трансцендентные, дифференциальные, интегральные),
2. аппроксимационные задачи (интерполяция, экстраполяция, численное интегрирование и дифференцирование),
3. задачи оптимизации,
4. стохастические проблемы.

Однако по мере усложнения объекта моделирования построение аналитической модели превращается в трудноразрешимую проблему. Тогда исследователь вынужден использовать имитационное моделирование.

В имитационном моделировании функционирование объектов, процессов или систем описывается набором алгоритмов. Алгоритмы имитируют реальные элементарные явления, составляющие процесс или систему с сохранением их логической структуры и последовательности протекания во времени. Имитационное моделирование позволяет по исходным данным получить сведения о состояниях процесса или системы в определенные моменты времени, однако прогнозирование поведения объектов, процессов или систем здесь затруднительно. Можно сказать, что имитационные модели - это проводимые на ЭВМ вычислительные

эксперименты с *математическими моделями*, имитирующими поведение реальных объектов, процессов или систем.

В зависимости от характера исследуемых реальных процессов и систем *математические модели* могут быть:

1. детерминированные,
2. стохастические.

В детерминированных моделях предполагается отсутствие всяких случайных воздействий, элементы модели (переменные, математические связи) достаточно точно установленные, поведение системы можно точно определить. При построении детерминированных моделей чаще всего используются алгебраические уравнения, интегральные уравнения, матричная алгебра.

Стохастическая модель учитывает случайный характер процессов в исследуемых объектах и системах, который описывается методами теории вероятности и математической статистики.

По виду входной информации модели разделяются на:

1. непрерывные,
2. дискретные.

Если информация и параметры являются непрерывными, а математические связи устойчивы, то модель - непрерывная. И наоборот, если информация и параметры - дискретны, а связи неустойчивы, то и *математическая модель* - дискретная.

По поведению моделей во времени они разделяются на:

1. статические,
2. динамические.

Статические модели описывают поведение объекта, процесса или системы в какой-либо момент времени. Динамические модели отражают поведение объекта, процесса или системы во времени.

По степени соответствия между *математической моделью* и реальным объектом, процессом или системой *математические модели* разделяют на:

1. *изоморфные* (одинаковые по форме),
2. *гомоморфные* (разные по форме).

Модель называется изоморфной, если между нею и реальным объектом, процессом или системой существует полное поэлементное соответствие. *Гомоморфной* - если существует соответствие лишь между наиболее значительными составными частями объекта и модели.

В дальнейшем для краткого определения вида *математической модели* в приведенной классификации будем пользоваться следующими обозначениями:

Первая буква:

Д – детерминированная,

С – стохастическая.

Вторая буква:

Н – непрерывная,

Д – дискретная.

Третья буква:

А – аналитическая,

И – имитационная.

Согласно этим обозначениям, описанная в лекции 2, модель кривошипно-шатунного механизма обозначается как модель вида ДНА (детерминированная, непрерывная, аналитическая), так как:

1. Отсутствует (точнее не учитывается) влияние случайных процессов, т.е. модель детерминированная (Д).
2. Информация и параметры - непрерывные, т.е. модель - непрерывная (Н),
3. Функционирование модели кривошипно-шатунного механизма описано в виде нелинейных трансцендентных уравнений, т.е. модель - аналитическая (А)

Лекция 2: Особенности построения математических моделей

Для использования ЭВМ при решении прикладных задач прежде всего прикладная задача должна быть "переведена" на формальный математический язык, т.е. для реального объекта, процесса или системы должна быть построена его математическая модель.

Математические модели в количественной форме, с помощью логико-математических конструкций, описывает основные свойства объекта, процесса или системы, его параметры, внутренние и внешние связи.

Для построения математической модели необходимо:

1. тщательно проанализировать реальный объект или процесс;
2. выделить его наиболее существенные черты и свойства;
3. определить переменные, т.е. параметры, значения которых влияют на основные черты и свойства объекта;
4. описать зависимость основных свойств объекта, процесса или системы от значения переменных с помощью логико-математических соотношений (уравнения, равенства, неравенства, логико-математические конструкций);
5. выделить внутренние связи объекта, процесса или системы с помощью ограничений, уравнений, равенств, неравенств, логико-математических конструкций;
6. определить внешние связи и описать их с помощью ограничений, уравнений, равенств, неравенств, логико-математических конструкций.

Математическое моделирование, кроме исследования объекта, процесса или системы и составления их математического описания, также включает:

1. построение алгоритма, моделирующего поведение объекта, процесса или системы;
2. проверка адекватности модели и объекта, процесса или системы на основе вычислительного и натурного эксперимента;
3. корректировка модели;
4. использование модели.

Математическое описание исследуемых процессов и систем зависит от:

1. природы реального процесса или системы и составляется на основе законов физики, химии, механики, термодинамики, гидродинамики, электротехники, теории пластичности, теории упругости и т.д.
2. требуемой достоверности и точности изучения и исследования реальных процессов и систем.

На этапе выбора математической модели устанавливаются: линейность и нелинейность объекта, процесса или системы, динамичность или статичность, стационарность или нестационарность, а также степень детерминированности исследуемого объекта или процесса. При математическом моделировании сознательно отвлекаются от конкретной физической природы объектов, процессов или систем и, в основном, сосредотачиваются на изучении количественных зависимостей между величинами, описывающими эти процессы.

Математическая модель никогда не бывает полностью тождественна рассматриваемому объекту, процессу или системе. Основанная на упрощении, идеализации она является приближенным описанием объекта. Поэтому результаты, полученные при анализе модели, носят приближенный характер. Их точность определяется степенью адекватности (соответствия) модели и объекта.

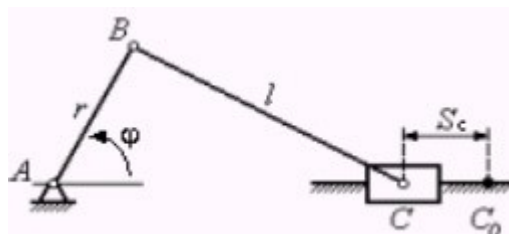
Построение математической модели обычно начинается с построения и анализа простейшей, наиболее грубой математической модели рассматриваемого объекта, процесса или системы. В дальнейшем, в случае необходимости, модель уточняется, делается ее соответствие объекту более полным.

Возьмем простой пример. Нужно определить площадь поверхности письменного стола. Обычно для этого измеряют его длину и ширину, а затем перемножают полученные числа. Такая элементарная процедура фактически обозначает следующее: реальный объект (поверхность стола) заменяется абстрактной математической моделью – прямоугольником. Прямоугольнику приписываются размеры, полученные в результате измерения длины и ширины поверхности стола, и площадь такого прямоугольника приближенно принимается за искомую площадь стола.

Однако модель прямоугольника для письменного стола – это простейшая, наиболее грубая модель. При более серьезном подходе к задаче прежде, чем воспользоваться для определения площади стола моделью прямоугольника, эту модель нужно проверить. Проверки можно осуществить следующим образом: измерить длины противоположных сторон стола, а также длины его диагоналей и сравнить их между собой. Если, с требуемой степенью точности, длины противоположных сторон и длины диагоналей попарно равны между собой, то поверхность стола действительно можно рассматривать как прямоугольник. В противном случае модель прямоугольника придется отвергнуть и заменить моделью четырехугольника общего вида. При более высоком требовании к точности может возникнуть необходимость пойти в уточнении модели еще дальше, например, учесть закругления углов стола.

С помощью этого простого примера было показано, что математическая модель не определяется однозначно исследуемым объектом, процессом или системой. Для одного и того же стола мы можем принять либо модель прямоугольника, либо более сложную модель четырехугольника общего вида, либо четырехугольника с закругленными углами. Выбор той или иной модели определяется требованием точности. С повышением точности модель приходится усложнять, учитывая новые и новые особенности изучаемого объекта, процесса или системы.

Рассмотрим пример: исследование движения кривошипно-шатунного механизма:



Для кинематического анализа этого механизма, прежде всего, необходимо построить его кинематическую модель. Для этого:

1. Заменяем механизм его кинематической схемой, где все звенья заменены жесткими связями;
2. Пользуясь этой схемой, мы выводим уравнение движения механизма;
3. Дифференцируя последнее, получаем уравнения скоростей и ускорения, которые представляют собой дифференциальные уравнения 1-го и 2-го порядка.

Запишем эти уравнения:

$$\begin{cases} S_c = r(1 - \cos \varphi + \frac{\lambda}{2} \sin^2 \varphi); \\ V_c = (\frac{d\varphi}{dt})r(\sin \varphi + \frac{\lambda}{2} \sin 2\varphi); \\ A_c = (\frac{d^2\varphi}{dt^2})r(\cos \varphi + \lambda \cos 2\varphi); \end{cases}$$

где C_0 – крайнее правое положение ползуна C:

r – радиус кривошипа AB;

l – длина шатуна BC;

φ – угол поворота кривошипа;

$$\lambda = r/l.$$

Полученные трансцендентные уравнения представляют математическую модель движения плоского аксиального кривошипно-шатунного механизма, основанную на следующих упрощающих предположениях:

1. нас не интересовали конструктивные формы и расположение масс, входящих в механизм тел, и все тела механизма мы заменили отрезками прямых. На самом деле, все звенья механизма имеют массу и довольно сложную форму. Например, шатун – это сложное сборное соединение, форма и размеры которого, конечно, будут влиять на движение механизма;
2. при построении математической модели движения рассматриваемого механизма мы также не учитывали упругость входящих в механизм тел, т.е. все звенья

рассматривали как абстрактные абсолютно жесткие тела. В действительности же, все входящие в механизм тела – упругие тела. Они при движении механизма будут как-то деформироваться, в них могут даже возникнуть упругие колебания. Это все, конечно, также будет влиять на движение механизма;

3. мы не учитывали погрешность изготовления звеньев, зазоры в кинематических парах А, В, С и т.д.

Таким образом, важно еще раз подчеркнуть, что, чем выше требования к точности результатов решения задачи, тем больше необходимость учитывать при *построении математической модели* особенности изучаемого объекта, процесса или системы. Однако, здесь важно во время остановиться, так как сложная математическая модель может превратиться в трудно разрешимую задачу.

Наиболее просто строится модель, когда хорошо известны законы, определяющие поведение и свойства объекта, процесса или системы, и имеется большой практический опыт их применения.

Более сложная ситуация возникает тогда, когда наши знания об изучаемом объекте, процессе или системе недостаточны. В этом случае при *построении математической модели* приходится делать дополнительные предположения, которые носят характер гипотез, такая модель называется гипотетической. Выводы, полученные в результате исследования такой гипотетической модели, носят условный характер. Для проверки выводов необходимо сопоставить результаты исследования модели на ЭВМ с результатами натурного эксперимента. Таким образом, вопрос применимости некоторой математической модели к изучению рассматриваемого объекта, процесса или системы не является математическим вопросом и не может быть решен математическими методами.

Основным критерием истинности является эксперимент, практика в самом широком смысле этого слова.

Построение математической модели в прикладных задачах – один из наиболее сложных и ответственных этапов работы. Опыт показывает, что во многих случаях правильно выбрать модель – значит решить проблему более, чем наполовину. Трудность данного этапа состоит в том, что он требует соединения математических и специальных знаний. Поэтому очень важно, чтобы при решении прикладных задач математики обладали специальными знаниями об объекте, а их партнеры, специалисты, – определенной математической культурой, опытом исследования в своей области, знанием ЭВМ и программирования.

Лекция 3: Компьютерное моделирование и вычислительный эксперимент. Решение математических моделей

Компьютерное моделирование как новый метод научных исследований основывается на:

1. построении математических моделей для описания изучаемых процессов;
2. использовании новейших вычислительных машин, обладающих высоким быстродействием (миллионы операций в секунду) и способных вести диалог с человеком.

Суть компьютерного моделирования состоит в следующем: на основе математической модели с помощью ЭВМ проводится серия вычислительных экспериментов, т.е. исследуются свойства объектов или процессов, находятся их оптимальные параметры и режимы работы, уточняется модель. Например, располагая уравнением, описывающим

протекание того или иного процесса, можно изменяя его коэффициенты, начальные и граничные условия, исследовать, как при этом будет вести себя объект. Более того, можно спрогнозировать поведение объекта в различных условиях.

Вычислительный эксперимент позволяет заменить дорогостоящий натурный эксперимент расчетами на ЭВМ. Он позволяет в короткие сроки и без значительных материальных затрат осуществить исследование большого числа вариантов проектируемого объекта или процесса для различных режимов его эксплуатации, что значительно сокращает сроки разработки сложных систем и их внедрение в производство.

Компьютерное моделирование и вычислительный эксперимент как новый метод научного исследования заставляет совершенствовать математический аппарат, используемый при построении математических моделей, позволяет, используя математические методы, уточнять, усложнять математические модели. Наиболее перспективным для проведения вычислительного эксперимента является его использование для решения крупных научно-технических и социально-экономических проблем современности (проектирование реакторов для атомных электростанций, проектирование плотин и гидроэлектростанций, магнитогидродинамических преобразователей энергии, и в области экономики – составление сбалансированного плана для отрасли, региона, для страны и др.).

В некоторых процессах, где натурный эксперимент опасен для жизни и здоровья людей, вычислительный эксперимент является единственно возможным (термоядерный синтез, освоение космического пространства, проектирование и исследование химических и других производств).

Для проверки адекватности математической модели и реального объекта, процесса или системы результаты исследований на ЭВМ сравниваются с результатами эксперимента на опытном натурном образце. Результаты проверки используются для корректировки математической модели или решается вопрос о применимости построенной математической модели к проектированию либо исследованию заданных объектов, процессов или систем.

В заключение подчеркнем еще раз, что компьютерное моделирование и вычислительный эксперимент позволяют свести исследование "нематематического" объекта к решению математической задачи. Этим самым открывается возможность использования для его изучения хорошо разработанного математического аппарата в сочетании с мощной вычислительной техникой. На этом основано применение математики и ЭВМ для познания законов реального мира и их использования на практике.

В задачах проектирования или исследования поведения реальных объектов, процессов или систем математические модели, как правило, нелинейны, т.к. они должны отражать реальные физические нелинейные процессы, протекающие в них. При этом параметры (переменные) этих процессов связаны между собой физическими нелинейными законами. Поэтому в задачах проектирования или исследования поведения реальных объектов, процессов или систем чаще всего используются математические модели типа ДНА.

Согласно классификации приведенной в лекции 1:

Д – модель детерминированная, отсутствует (точнее не учитывается) влияние случайных процессов.

Н – модель непрерывная, информация и параметры непрерывны.

A – модель аналитическая, функционирование модели описывается в виде уравнений (линейных, нелинейных, систем уравнений, дифференциальных и интегральных уравнений).

Итак, мы построили математическую модель рассматриваемого объекта, процесса или системы, т.е. представили прикладную задачу как математическую. После этого наступает второй этап решения прикладной задачи – поиск или разработка метода решения сформулированной математической задачи. Метод должен быть удобным для его реализации на ЭВМ, обеспечивать необходимое качество решения.

Все методы решения математических задач можно разделить на 2 группы:

1. точные методы решения задач;
2. численные методы решения задач.

В точных методах решения математических задач ответ удается получить в виде формул.

Например, вычисление корней квадратного уравнения:

$$ax^2 + bx + c = 0,$$
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

или, например, вычисление производных функций:

$$y = \sin(x), y' = \cos(x);$$
$$y = x^n, y' = nx^{n-1};$$

или вычисление определенного интеграла:

$$\int_b^a \cos(x) dx = \sin(b) - \sin(a);$$

Однако, подставляя числа в формулу в виде конечных десятичных дробей, мы все равно получаем приближенные значения результата.

Для большинства задач, встречающихся на практике, точные методы решения или неизвестны, или дают очень громоздкие формулы. Однако, они не всегда являются необходимыми. Прикладную задачу можно считать практически решенной, если мы сумеем ее решить с нужной степенью точности.

Для решения таких задач разработаны численные методы, в которых решение сложных математических задач сводится к последовательному выполнению большого числа простых арифметических операций. Непосредственная разработка численных методов относится к вычислительной математике.

Примером численного метода является метод прямоугольников для приближенного интегрирования, не требующий вычисления первообразной для подынтегральной функции. Вместо интеграла вычисляется конечная квадратурная сумма:

$$\int_b^a f(x) dx \approx \sum_{i=1}^{n+1} f(x_i) \Delta x_i;$$

где

$x_1=a$ – нижний предел интегрирования;

$x_{n+1}=b$ – верхний предел интегрирования;

n – число отрезков, на которые разбит интервал интегрирования (a, b) ;

Δx_i – длина элементарного отрезка;

$f(x_i)$ – значение подынтегральной функции на концах элементарных отрезков интегрирования.

Чем больше число отрезков n , на которые разбит интервал интегрирования, тем ближе приближенное решение к истинному, т.е. тем точнее результат.

Таким образом, в прикладных задачах и при применении точных методов решения, и при применении численных методов решения результаты вычислений носят приближенный характер. Важно только добиться того, чтобы ошибки укладывались в рамки требуемой точности.

Численные методы решения математических задач известны давно, еще до появления ЭВМ, но ими пользовались редко и только в сравнительно простых случаях в силу чрезвычайной трудоемкости вычислений. Широкое применение численных методов стало возможным благодаря ЭВМ.

Лекция 4: Компьютерное имитационное моделирование. Статистическое имитационное моделирование

Компьютерное моделирование как новый метод научных исследований основывается на:

1. построении математических моделей для описания изучаемых процессов;
2. использовании новейших вычислительных машин, обладающих высоким быстродействием (миллионы операций в секунду) и способных вести диалог с человеком.

Суть компьютерного моделирования состоит в следующем: на основе математической модели с помощью ЭВМ проводится серия вычислительных экспериментов, т.е. исследуются свойства объектов или процессов, находятся их оптимальные параметры и режимы работы, уточняется модель. Например, располагая уравнением, описывающим протекание того или иного процесса, можно изменяя его коэффициенты, начальные и граничные условия, исследовать, как при этом будет вести себя объект. *Имитационные модели* Имитационные модели – это проводимые на ЭВМ вычислительные эксперименты с математическими моделями, имитирующими поведение реальных объектов, процессов или систем.

Реальные процессы и системы можно исследовать с помощью двух типов математических моделей: аналитических и имитационных.

В аналитических моделях поведение реальных процессов и систем (РПС) задается в виде явных функциональных зависимостей (уравнений

линейных или нелинейных, дифференциальных или интегральных, систем этих уравнений). Однако получить эти зависимости удастся только для сравнительно простых РПС. Когда явления сложны и многообразны исследователю приходится идти на упрощенные представления сложных РПС. В результате аналитическая модель становится слишком грубым приближением к действительности. Если все же для сложных РПС удастся получить аналитические модели, то зачастую они превращаются в трудно разрешимую проблему. Поэтому исследователь вынужден часто использовать *имитационное моделирование*.

Имитационное моделирование представляет собой численный метод проведения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими поведение реальных объектов, процессов и систем во времени в течении заданного периода. При этом функционирование РПС разбивается на элементарные явления, подсистемы и модули. Функционирование этих элементарных явлений, подсистем и модулей описывается набором алгоритмов, которые имитируют элементарные явления с сохранением их логической структуры и последовательности протекания во времени.

Имитационное моделирование – это совокупность методов алгоритмизации функционирования объектов исследований, программной реализации алгоритмических описаний, организации, планирования и выполнения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими функционирование РПС в течении заданного периода.

Под алгоритмизацией функционирования РПС понимается пооперационное описание работы всех ее функциональных подсистем отдельных модулей с уровнем детализации, соответствующем комплексу требований к модели.

"*Имитационное моделирование*" (ИМ) – это двойной термин. "Имитация" и "моделирование" - это синонимы. Фактически все области науки и техники являются моделями реальных процессов. Чтобы отличить математические модели друг от друга, исследователи стали давать им дополнительные названия. Термин "*имитационное моделирование*" означает, что мы имеем дело с такими математическими моделями, с помощью которых нельзя заранее вычислить или предсказать поведение системы, а для предсказания поведения системы необходим вычислительный эксперимент (имитация) на математической модели при заданных исходных данных.

Основное достоинство ИМ:

1. возможность описания поведения компонент (элементов) процессов или систем на высоком уровне детализации;
2. отсутствие ограничений между параметрами ИМ и состоянием внешней среды РПС;
3. возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы;

Эти достоинства обеспечивают имитационному методу широкое распространение.

Рекомендуется использовать *имитационное моделирование* в следующих случаях:

1. Если не существует законченной постановки задачи исследования и идет процесс познания объекта моделирования. *Имитационная модель* служит средством изучения явления.
2. Если аналитические методы имеются, но математические процессы сложны и трудоемки, и *имитационное моделирование* дает более простой способ решения задачи.
3. Когда кроме оценки влияния параметров (переменных) процесса или системы желательно осуществить наблюдение за поведением компонент (элементов) процесса или системы (ПС) в течение определенного периода.
4. Когда *имитационное моделирование* оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальных условиях (реакции термоядерного синтеза, исследования космического пространства).
5. Когда необходимо контролировать протекание процессов или поведение систем путем замедления или ускорения явлений в ходе имитации.
6. При подготовке специалистов новой техники, когда на *имитационных моделях* обеспечивается возможность приобретения навыков в эксплуатации новой техники.
7. Когда изучаются новые ситуации в РПС. В этом случае имитация служит для проверки новых стратегий и правил проведения натурных экспериментов.
8. Когда особое значение имеет последовательность событий в проектируемых ПС и модель используется для предсказания узких мест в функционировании РПС.

Однако ИМ наряду с достоинствами имеет и недостатки:

1. Разработка хорошей ИМ часто обходится дороже создания аналитической модели и требует больших временных затрат.
2. Может оказаться, что ИМ неточна (что бывает часто), и мы не в состоянии измерить степень этой неточности.
3. Зачастую исследователи обращаются к ИМ, не представляя тех трудностей, с которыми они встретятся и совершают при этом ряд ошибок методологического характера.

И тем не менее ИМ является одним из наиболее широко используемых методов при решении задач синтеза и анализа сложных процессов и систем.

Одним из видов *имитационного моделирования* является статистическое *имитационное моделирование*, позволяющее воспроизводить на ЭВМ функционирование сложных случайных процессов.

При исследовании сложных систем, подверженных случайным возмущениям используются вероятностные аналитические модели и вероятностные *имитационные модели*.

В вероятностных аналитических моделях влияние случайных факторов учитывается с помощью задания вероятностных характеристик случайных процессов (законы распределения вероятностей, спектральные плотности или корреляционные функции). При этом построение вероятностных аналитических моделей представляет собой сложную вычислительную

задачу. Поэтому вероятностное аналитическое моделирование используют для изучения сравнительно простых систем.

Подмечено, что введение случайных возмущений в *имитационные модели* не вносит принципиальных усложнений, поэтому исследование сложных случайных процессов проводится в настоящее время, как правило, на *имитационных моделях*.

В вероятностном *имитационном моделировании* оперируют не с характеристиками случайных процессов, а с конкретными случайными числовыми значениями параметров ПС. При этом результаты, полученные при воспроизведении на *имитационной модели* рассматриваемого процесса, являются случайными реализациями. Поэтому для нахождения объективных и устойчивых характеристик процесса требуется его многократное воспроизведение, с последующей статистической обработкой полученных данных. Именно поэтому исследование сложных процессов и систем, подверженных случайным возмущениям, с помощью *имитационного моделирования* принято называть статистическим моделированием.

Статистическая модель случайного процесса - это алгоритм, с помощью которого имитируют работу сложной системы, подверженной случайным возмущениям; имитируют взаимодействие элементов системы, носящих вероятностный характер.

При реализации на ЭВМ статистического *имитационного моделирования* возникает задача получения на ЭВМ случайных числовых последовательностей с заданными вероятностными характеристиками. Численный метод решающий задачу генерирования последовательности случайных чисел с заданными законами распределения получил название "метод статистических испытаний" или "метод Монте-Карло".

Так как метод Монте-Карло кроме статистического моделирования имеет приложение к ряду численных методов (взятие интегралов, решение уравнений), то целесообразно иметь различные термины.

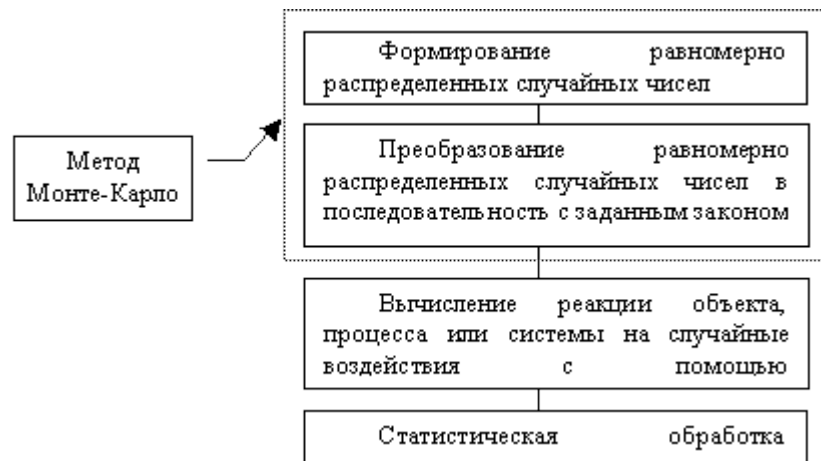
Итак, статистическое моделирование - это способ изучения сложных процессов и систем, подверженных случайным возмущениям, с помощью *имитационных моделей*.

Метод Монте-Карло – это численный метод, моделирующий на ЭВМ псевдослучайные числовые последовательности с заданными вероятностными характеристиками.

Методика статистического моделирования состоит из следующих этапов:

1. Моделирование на ЭВМ псевдослучайных последовательностей с заданной корреляцией и законом распределения вероятностей (метод Монте-Карло), имитирующих на ЭВМ случайные значения параметров при каждом испытании;
2. Преобразование полученных числовых последовательностей на *имитационных математических моделях*.
3. Статистическая обработка результатов моделирования.

Обобщенный алгоритм метода статистических испытаний представлен на рисунке:



Лекция 5: Метод Монте-Карло

При реализации на ЭВМ статистического моделирования возникает задача получения (генерирования) на ЭВМ случайных числовых последовательностей с заданными вероятностными характеристиками, в которых каждое число – это имитация случайного значения какого-либо параметра реального процесса или системы, подверженного случайным возмущениям.

Генерирование на ЭВМ таких случайных числовых последовательностей получило название "*метод Монте-Карло*".

В математической литературе часто используется термины "последовательность случайных чисел" или просто "случайные числа".

Однако, если проанализировать эти термины с философской точки зрения, то можно спросить: а есть ли такой объект как случайное число? Число 2 – это случайное число? Или число 17 – это случайное число? Конечно, нет. Если использовать точные термины, то можно говорить только о случайной последовательности чисел или о случайном значении параметров. Однако, в литературе широко используется термины "случайные числа" и "последовательность случайных чисел", и это означает, что каждое число было получено самым произвольным образом, без всякой связи с другими членами последовательности, и что у него есть определенная вероятность оказаться в заданном интервале.

Раньше ученые, нуждавшиеся для своей работы в случайных числах, раскладывали карты, бросали кости или вытаскивали шары из урны, которую предварительно как следует трясли. В 1927 году Л. Типпетт опубликовал таблицы, содержащие свыше 40 000 случайных чисел, взятых произвольно из отчетов по переписи. Позже были сконструированы специальные машины, механически вырабатывающие случайные числа. В 1955 году компания RAND Corporation опубликовала хорошо известные таблицы с миллионом случайных чисел, полученных одной из таких машин.

После создания ЭВМ начались поиски эффективных алгоритмов получения (генерирования на ЭВМ) последовательностей случайных чисел, пригодных для программной реализации.

Последовательности случайных чисел, вырабатываемые детерминистскими способами (т. е. с помощью специальных алгоритмов) называются псевдослучайными или квазислучайными. В дальнейшем мы будем их называть просто случайными последовательностями, понимая, что они просто производят впечатление случайных.

Задачу *генерирования случайных чисел* на ЭВМ с заданным законом распределения решают в несколько этапов:

Вначале получают последовательность равномерно распределенных на интервале $[0, 1]$ псевдослучайных чисел.

Из равномерно распределенной последовательности получают последовательность псевдослучайных чисел с заданным законом распределения в заданном интервале.

Равномерным называется такое распределение, при котором каждое возможное случайное число равновероятно. Обычно, если специально не оговорен закон распределения случайных чисел, то обычно имеют в виду равномерное распределение.

Сущность алгоритмических методов получения равномерно распределенных псевдослучайных чисел заключается в том, что псевдослучайные числа получают с помощью некоторой рекуррентной формулы $x_{i+1} = f(x_i)$, где каждое следующее $(i+1)$ -е значение образуется из предыдущего (или группы предыдущих) путем применения некоторого алгоритма, содержащего логические и арифметические операции.

Известно большое количество имитации равномерного распределения (методы вычетов, суммирования, усечения, перемешивания). Общими для всех этих методов являются требования:

1. Количество операций для получения каждого псевдослучайного числа должно быть минимальным;
2. Случайные числа генерируются как можно менее коррелированными, а их распределение – близким к равномерному

5.1 Метод середины квадрата

Первым алгоритмический метод получения равномерно распределенных псевдослучайных чисел предложил Джон фон Нейман (один из основоположников кибернетики). *Метод получил название "метод середины квадрата"*.

Суть метода: предыдущее случайное число возводится в квадрат, а затем из результата извлекаются средние цифры.

Например:

пусть $x_0 = 0.2061$, тогда $x_0^2 = 0.04|2477|21$;

$x_1 = 0.2477$, $x_1^2 = 0.06|1355|29$;

$x_2 = 0.1355$, $x_2^2 = 0.01|8360|25$

и т.д.

Как видно *метод середины квадрата* довольно хорошо должен "перемешивать" предыдущее число. Однако он имеет недостатки:

1. Если какой-нибудь член последовательности окажется равным нулю, то все последующие члены также будут нулями.
2. Последовательности имеют тенденцию "зацикливаться", т. е. в конце концов, образуют цикл, который повторяется бесконечное число раз.

Свойство "зацикливаться" присуще всем последовательностям, построенных по рекуррентной формуле $x_{i+1} = f(x)$.

Повторяющийся цикл называется периодом. Длина периода у различных последовательностей разная. Чем больше, тем лучше.

5.2 Линейный конгруэнтный метод

Наилучшие из известных сегодня методов имитации случайных чисел представляют собой частные случаи схемы, предложенные в 1948 году Д.Х.Лемером.

Суть метода: Выбираем четыре "магических числа":

x_0 – начальное значение, $x_0 \geq 0$;

a – множитель, $a \geq 0$;

c – приращение, $c \geq 0$;

m – модуль, $m > x_0, m > a, m > c$.

Тогда искомая последовательность случайных чисел получается из соотношения:

$$x_{i+1} = (ax_i + c)(m), n \geq 0, \quad (5.1)$$

т. е. каждое случайное число – это остаток, при делении $(ax_i + c)$ на m (операция mod - "определение остатка", термин взят от слова "modulo" – в переводе "остаток").

Последовательность, полученная из соотношения (5.1) называется линейной конгруэнтной последовательностью.

Пример: $x_0 = a = c = 7, m = 10$.

Тогда последовательность имеет вид: 7, 6, 9, 0, 7, 6, 9, 0,

Как видно, при выбранных значениях "магических чисел" последовательность почти сразу "зациклилась", длина периода = 4.

Из этого примера видно, что "магические числа" нельзя выбирать произвольно. Проведено много исследований и доказано теорем по вопросу "как правильно выбирать" "магические числа".

Метод получения случайных чисел при $c=0$ называется "мультипликативный конгруэнтный метод", при $c \neq 0$ - "смешанный конгруэнтный метод". При $c=0$, выработка последовательностей происходит быстрее, но при этом уменьшается длина периода последовательностей.

Первоначально в методе Лемера было принято $c=0$. Идея получения более длинных последовательностей за счет $c \neq 0$ принадлежит Томпсону и независимо Ротенбергу.

Выбор модуля m . Для получения длинных последовательностей и для увеличения скорости вычисления рекомендуется m выбирать равным размеру машинного слова. Для 32^* разрядного машинного слова $m = 2^{31} = 2147483648$, (левый нулевой бит слова отведен под знак числа).

При этом в 32^* разрядном машинном слове, максимальное целое число, размещающее в машинном слове, равно $\omega = 2^{31} - 1 = 2147483647 \dots$

Тогда $m = \omega + 1$.

Значение множителя также влияет на длину периода последовательностей. По этому вопросу также проведено много исследований.

Линейные конгруэнтные последовательности – не единственный из предложенных источников случайных чисел. Его можно обобщить, превратив его, например, в квадратичный конгруэнтный метод

$$x_{i+1} = (dx_i^2 + a \cdot x_i + c) \text{Mod}(m).$$

Известен квадратичный метод, предложенный Р. Ковэю:

$$x_{i+1} = x_i(x_i + 1) \text{Mod}(2^c).$$

Известен метод получения случайных чисел, где реализуется последовательность Фибоначчи:

$$x_{i+1} = (x_i + x_{i-1}) \text{Mod}(m).$$

Известен также метод получения случайных чисел, предложенный Грином:

$$x_{i+1} = (x_i + x_{i-k}) \text{Mod}(m),$$

где k - большое число.

Имеются еще, так называемые, аддитивные методы, где не требуются операции умножения и деления, и другие методы.

Лекция 6: Генерирование на ЭВМ последовательностей равномерно распределенных случайных чисел. Моделирование нормально распределенной случайной величины

В стандартном математическом и программном обеспечении различных типов ЭВМ имеются специальные процедуры и подпрограммы для *генерации равномерно распределенных последовательностей псевдослучайных чисел*.

В настоящее время все языки высокого уровня имеют программные генераторы равномерно распределенных последовательностей псевдослучайных чисел. Их называют датчиками случайных чисел. Датчики случайных чисел, как правило имеют имена: RAN, RAND, RANDU, RND, RANDOM, RANDOMIZE и т.д. Эти имена получены от английского слова random, означающего случайный или выбранный наугад. Поэтому датчики случайных чисел иногда называют рандомизаторами.

Датчики случайных чисел обычно генерируют последовательность действительных чисел u_n , равномерно распределенных между нулем и единицей, т.е. последовательность случайных дробей в интервале $[0, 1]$. Но вначале датчики генерируют последовательность целых случайных чисел x_i в интервале от нуля до m , где m – размер машинного слова (на единицу больше максимального целого числа, размещающегося в машинном слове).

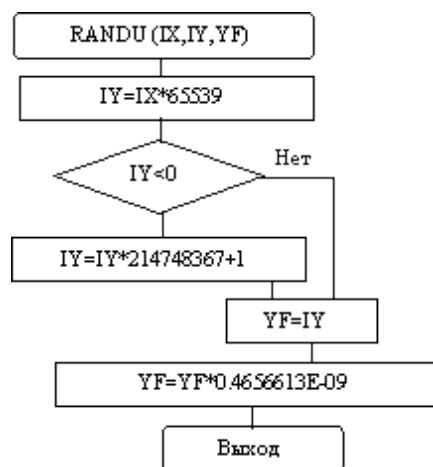
Как уже говорилось, для 32-х разрядного машинного слова:

$$m = 2^{31} = 2147483648;$$

$$1/m = 0.4656613 \text{ E-}9.$$

Приведем датчик равномерно распределенных случайных чисел RANDU.

Этот датчик специально предназначен для системы IBM и образует 2^{29} значений перед началом повторений, т.е. длина цикла равна 2^{29} (при 32-х разрядном машинном слове). Алгоритм датчика реализует метод остатка степеней:



Назначение датчика RANDU:

Вычисление равномерно распределенных случайных действительных чисел YF в интервале $[0, 1]$ и случайных целых чисел IY в интервале $[0, 2^{31}]$. В качестве входа служит целое случайное число IX , на выходе образуется новое целое число IY и вещественное YF .

Обращение к датчику:

`RANDU (IX, IY, YF).`

Описание параметров:

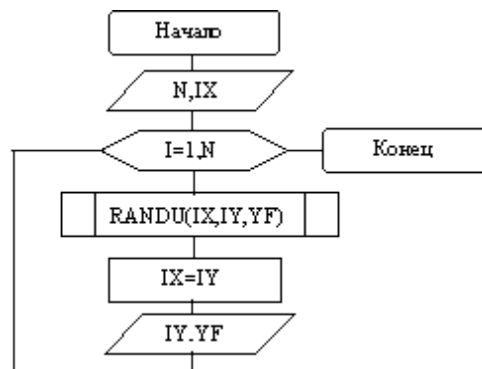
IX – при первом обращении – нечетное целое число с числом цифр ≤ 9 .

После первого обращения $IX=IY$, где IY – целое число, вычисленное при предыдущем обращении.

IY – полученное в результате целое случайное число, требуемое при последующих обращениях. IY находится в интервале $[0, 2^{31}]$.

YF – полученное в результате равномерно распределенное действительное число в интервале $[0, 1]$, представленное в форме с плавающей запятой.

Используя датчик `RANDU`, построим алгоритм вычисления последовательности равномерно распределенных случайных чисел и ее печать:



Алгоритм вычисления последовательности равномерно распределенных случайных чисел.

Исходные данные:

N - длина последовательности (количество испытаний).

IX - начальное значение, нечетное целое число с числом цифр < 9 .

Например: $IX = 1234567$.

Различные начальные значения параметра IX позволяют формировать различные последовательности равномерно распределенных псевдослучайных чисел.

Для генерирования равномерно распределенных случайных чисел на отрезке $[0, 1]$ используется отделение дробной части от сложного арифметического выражения, содержащего предшествующее число.

$$V_{i+1} = \text{FRAC}(k V_i)$$

FRAC – оператор выделения дробной части,

V_i - предшествующее случайное число,

V_{i+1} - последующее случайное число,

$$k - 8t = 3,$$

t - нечетное целое число.

Задание различных начальных значений v_0 позволяет формировать различные последовательности случайных чисел.

Количество случайных чисел в одном периоде от нескольких тысяч до сотен тысяч.

Для перевода равномерно распределенных случайных чисел из отрезка $[0, 1]$ в отрезок $[a, b]$ можно использовать:

$$x_{i+1} = a + (b-a) V_{i+1}.$$

Задачу моделирования случайных величин с нормальным законом распределения решают в несколько этапов:

1. Вначале имитируют равномерное распределение и получают последовательность псевдослучайных чисел, равномерно распределенных на интервале $[0, 1]$.
2. Затем, используя равномерно распределенную псевдослучайную величину, получают последовательность псевдослучайных чисел с нормальным законом распределения (чаще всего в нормированном виде, т.е. $M(X) = 0$, $\sigma = 1$).

Пусть y – равномерно распределенная случайная величина на интервале $[0, 1]$. Необходимо получить случайную величину X с нормальным законом распределения.

Различают три основных способа формирования последовательности нормально распределенных случайных величин:

1. Прямое преобразование псевдослучайного числа y являющегося реализацией случайной величины y , равномерно распределенной на интервале $[0, 1]$, с помощью некоторой функции w в число x , которое может рассматриваться как реализация случайной величины x , имеющей нормальный закон распределения.
2. Отсеивание псевдослучайных чисел из первоначальной последовательности y равномерно распределенной на интервале $[0, 1]$, таким образом, чтобы оставшиеся числа были распределены по нормальному закону.
3. Моделирование условий, соответствующих центральной предельной теореме теории вероятности.

Рассмотрим некоторые методы моделирования нормально распределенной случайной величины.

Метод полярных координат относится к первому способу получения последовательности псевдослучайных чисел с нормальным законом распределения. Метод вычисляет две независимые нормально распределенные случайные величины x_1 и x_2 со средним значением, равным нулю, и среднеквадратичным отклонением, равным единице, по двум

заданным независимым равномерно распределенным случайным числам y_1 и y_2 .

Алгоритм метода:

1. Выработать два независимых случайных числа y_1 и y_2 , равномерно распределенных в интервале $[0, 1]$.
2. Установить:

$$V_1 = 2 * y_1 - 1;$$

$$V_2 = 2 * y_2 - 1.$$

Теперь величины V_1 и V_2 равномерно распределены в интервале $[-1; +1]$ и их удобно представить в форме с плавающей запятой.

1. Установить:

$$S = V_1^2 + V_2^2$$

2. Проверить условие: $S \geq 1$ Если "да", то необходимо вернуться к шагу 1. Если "нет", то переходим к шагу 5.

3. Вычисляем x_1 и x_2 :

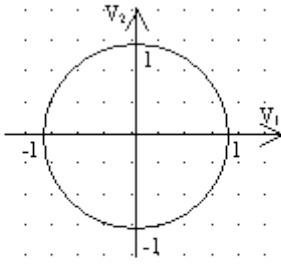
$$x_1 = V_1 \cdot \sqrt{\frac{-2 \cdot \ln S}{S}}, \quad (6.1)$$

4. $x_2 = V_2 \cdot \sqrt{\frac{-2 \cdot \ln S}{S}}.$

5. Полученные величины x_1 и x_2 – это требующиеся значения нормально распределенных случайных величин со средним значением равным нулю, и среднеквадратичным отклонением $\sigma = 1$.
6. При других значениях среднего α и среднеквадратичного отклонения σ делаем пересчет:

$$x_i = \alpha + x_i \cdot \sigma.$$

Метод полярных координат легко доказать, воспользовавшись аналитической геометрией. Рассмотрим плоскость, определенную декартовыми координатами V_1 и V_2 . С помощью шагов 1 и 2 метода мы получаем на плоскости равномерно распределенные случайные точки с декартовыми координатами (V_1, V_2) и полярными координатами $V_1 = R \cdot \cos \theta, V_2 = R \cdot \sin \theta$, где $R^2 = S$. Далее, с помощью шагов 3 и 4 метода, мы из этих случайных точек оставляем только те точки, которые находятся внутри единичного круга.



При этом попадание точек внутрь единичного круга подчиняется закону нормального распределения со средним значением равным нулю, и среднеквадратичным отклонением равным единице.

Переходя к полярным координатам точек, которые равномерно распределены внутри единичного круга имеем

$$\begin{aligned} x_1 &= \sqrt{-2 \cdot \ln y_1} \cdot \cos(2\pi \cdot y_1), \\ x_2 &= \sqrt{-2 \cdot \ln y_2} \cdot \cos(2\pi \cdot y_2). \end{aligned} \quad (6.2)$$

Метод, основанный на центральной предельной теореме

Этот метод моделирования относится к третьему способу получения последовательности чисел с нормальным законом распределения. Метод основан на приближенном воспроизводстве условий, при которых справедлива центральная предельная теорема теории вероятности.

Согласно центральной предельной теореме, при сложении достаточно большого независимых случайных величин с произвольным законом распределения получается случайная величина, распределенная по нормальному закону. Опыт показывает, что при сложении всего шести ($k=6$) случайных величин равномерно распределенных на интервале $[0, 1]$, получается случайная величина, которая с точностью, достаточной для большинства прикладных задач, может считаться нормальной.

Рассмотрим метод аппроксимации нормально распределенной случайной величины x , основанный на использовании двенадцати ($k=12$) равномерно распределенных случайных величин.

Алгоритм метода:

1. Сложить 12 равномерно распределенных псевдослучайных чисел y_i .
2. Пронормировать полученную сумму, т.е. получить случайную величину T с $M(T)=0$ и $\sigma = 1$, где T – нормально распределенная случайная величина.
3. Результат привести в соответствие с заданным математическим ожиданием и среднеквадратичным отклонением σ .

Нормально распределенная случайная величина x с требуемыми значениями математического ожидания и среднеквадратичного отклонения σ определяется как:

$$x = \sigma \cdot T + a.$$

Пусть

$$Z = \sum_{i=1}^{12} y_i,$$

где y_i – независимые равномерно распределенные на интервале $[0, 1]$ случайные величины.

Ранее было показано, что математическое ожидание и дисперсия равномерно распределенной на интервале $[0, 1]$ случайной величины y соответственно равны:

$$M(Y) = \frac{1}{2}; D(Y) = \sigma_y^2 = \frac{1}{12}.$$

Тогда математическое ожидание суммы z равно:

$$M(Z) = \sum_{i=1}^{12} M(Y) = 12 \frac{1}{2} = 6.$$

а ее дисперсия $D(Z)$ равна:

$$D(Z) = \sigma_z^2 = \sum_{i=1}^{12} D(Y) = 12 \frac{1}{12} = 1.$$

Пронормируем сумму z , т.е. перейдем от нее к величине:

$$T = (Z - M(Z)) / \sigma_z = (Z - 6) / 1 = Z - 6.$$

Переходя к требуемым математическому ожиданию a и среднеквадратичному отклонению σ , окончательно имеем:

$$x = \sigma \cdot T + a = \sigma(Z - 6) + a.$$

Таким образом, чтобы определить значение нормально распределенной случайной величины с математическим, равным нулю, и среднеквадратичным отклонением, равным единице, необходимо взять 12 равномерно распределенных чисел, сложить их, а из суммы вычесть 6, т.е.:

$$T = \sum_{i=1}^{12} y_i - 6. \quad (6.4)$$

Чтобы определить значение нормально распределенной случайной величины с требуемым математическим ожиданием a и требуемым среднеквадратичным отклонением σ необходимо из суммы двенадцати

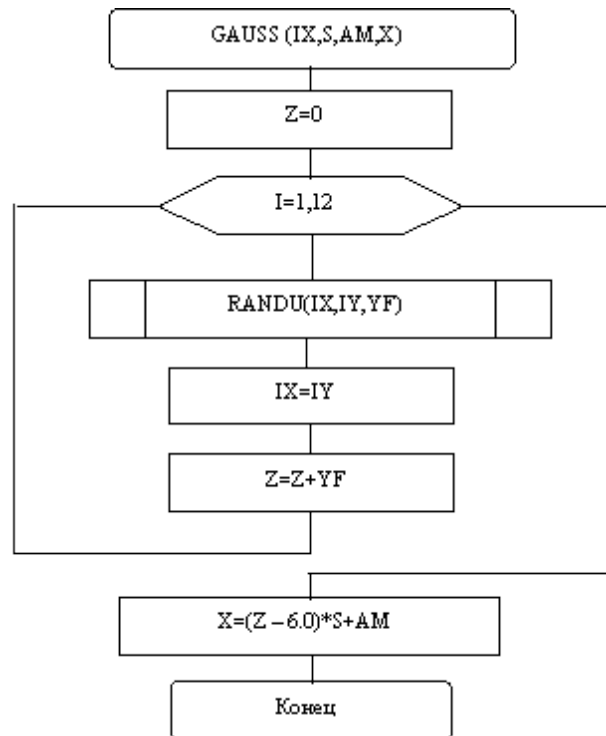
равномерно распределенных чисел вычесть 6, а результат умножить на σ и прибавить a , т.е.

$$x = \left(\sum_{i=1}^{12} y_i - 6 \right) \sigma + a. \quad (6.5)$$

Теперь перейдем к генерированию последовательности нормально распределенных случайных чисел.

Рассмотрим датчик нормально распределенных случайных чисел.

Алгоритм датчика (рис. 6.3) реализует *метод получения последовательностей псевдослучайных чисел с нормальным законом распределения, основанный на центральной предельной теореме вероятностей*. Алгоритм датчика требует обращения к RANDU для вычисления равномерно распределенных случайных чисел.



Назначение датчика GAUSS:

Вычисление нормально распределенного псевдослучайного числа x с заданным математическим ожиданием AM и среднеквадратичным отклонением S .

Обращение к датчику: GAUSS (IX, S, AM, X),

Описание параметров:

IX – параметр необходимый для обращения к `RANDU`. При первом обращении, IX – целое число с числом цифр ≤ 9 . После первого обращения $IX=IY$, где IY – целое равномерно распределенное случайное число, вычисленное с помощью равномерно распределенных случайных чисел `RANDU`.

s – требуемое среднеквадратичное отклонение нормального распределения.

AM – требуемое математическое ожидание нормального распределения.

x – значение вычисленной нормально распределенной случайной величины.

Требуемые подпрограммы:

`RANDU` – датчик равномерно распределенных случайных чисел.

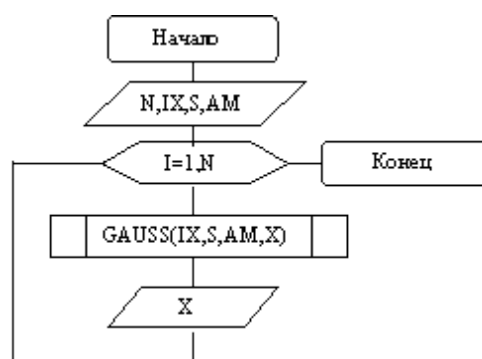
Обращение к `RANDU`:

`RANDU (IX, IY, YF),`

где

YF – полученное в результате обращения случайное равномерно распределенное число в интервале $[0, 1]$ и представленное в форме с плавающей запятой.

Используя датчик `GAUSS`, построим алгоритм (рис. 6.4) вычисления последовательности нормально распределенных случайных чисел x с требуемым математическим ожиданием AM и среднеквадратичным отклонением s .



Исходные данные:

N - длина последовательности нормально распределенных чисел (количество испытаний).

IX - начальное значение, нечетное целое число с числом цифр ≤ 9 .

Например .

σ , μ — характеристики нормального распределения: среднеквадратичное отклонение и математическое ожидание.

Результат:

x — значение нормального распределения псевдослучайного числа.

Используя различные начальные значения параметра μ , можно формировать различные последовательности нормального распределенных псевдослучайных чисел.

Лекция 7: Компьютерное моделирование при обработке опытных данных

Любому специалисту в своей практической деятельности приходится изучать зависимости между различными параметрами исследуемых объектов, процессов и систем.

Например: зависимость числа оборотов двигателя от нагрузки, т.е. $n=f(M_{кр.})$; зависимость силы резания при обработке детали на металлорежущем станке от глубины резания, т.е. $P=f(t)$, и т.д.

Из всех способов задания зависимостей наиболее удобным является аналитический способ задания зависимости в виде функции $n=f(M_{кр.})$, $P=f(t)$, $y=f(t)$.

Однако на практике специалист чаще всего получает зависимости между исследуемыми параметрами экспериментально. В этом случае ставится натурный эксперимент, изменяются значения параметров на входе системы, измеряются значения параметров на выходе системы. Результаты измерений заносятся в таблицу.

Таким образом, в результате проведения натурального эксперимента получаем зависимости между исследуемыми параметрами в виде таблицы, т.е. получаем, так называемую, табличную функцию.

Далее с этой табличной функцией необходимо вести научно-исследовательские расчеты. Например, необходимо проинтегрировать или продифференцировать табличную функцию и т.д.

Рассмотрим две задачи по обработке опытных данных:

1. задачу интерполирования,

2. задачу аппроксимации.

Интерполирование функций

Дана табличная функция, т.е. дана таблица, в которой для некоторых дискретных значений аргумента x_i , расположенных в порядке возрастания, заданы соответствующие значения функции y_i :

i	x	y
0	x_0	y_0
1	x_1	y_1
2	x_2	y_2
...	...	...
i	x_i	y_i
...	...	...
n	x_n	y_n

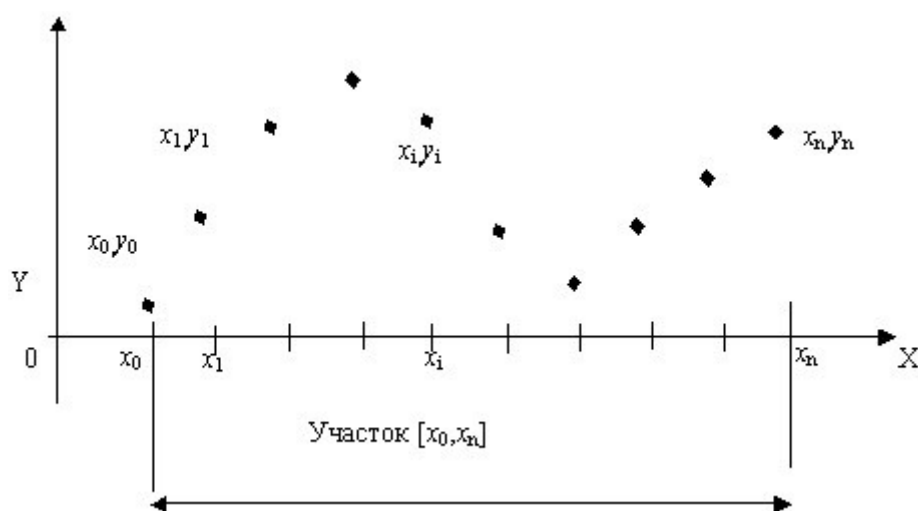
$$y_i = f(x_i), i = \overline{0, n}. \quad (7.1)$$

Точки с координатами (x_i, y_i) называются узловыми точками или узлами.

Количество узлов в табличной функции равно

$$N = n + 1.$$

На графике табличная функция представляется в виде совокупности узловых точек:



Длина участка $[x_0, x_n]$ равна $(x_n - x_0)$.

В расчетной практике инженера часто возникают задачи найти значение функции для аргументов, которые отсутствуют в таблице. Такие задачи называются задачами интерполирования или экстраполирования.

Задача *интерполирования функции* (или задача интерполяции) состоит в том, чтобы найти значения y_k табличной функции в любой промежуточной точке x_k , расположенной внутри интервала $[x_0, x_n]$, т.е.

$$x_1 < x_k < x_{i+1}$$

и

$$x_k \in [x_0, x_n].$$

Задача *экстраполирования функции* (или задача экстраполяции) состоит в том, чтобы найти значения y_1 табличной функции в точке x_1 , которая не входит в интервал $[x_0, x_n]$, т.е.

$$x_l < x_0; x_l > x_n.$$

Такую задачу часто называют задачей прогноза.

Обе эти задачи решаются при помощи нахождения аналитического выражения некоторой вспомогательной функции $F(x)$, которая приближала бы заданную табличную функцию, т.е. в узловых точках принимала бы значение табличных функций

$$F(x_i) = y_i, i = 0, 1, 2, \dots, n.$$

Для определенности задачи искомую функцию $F(x)$ будем искать из класса алгебраических многочленов:

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x^1 + a_n x^0. \quad (7.2)$$

Этот многочлен должен пройти через все узловые точки, т.е.

$$P_n(x_i) = y_i. \quad (7.3)$$

Поэтому степень многочлена n зависит от количества узловых точек N и равна количеству узловых точек минус один, т.е. $n=N-1$.

Многочлен вида (7.2), который проходит через все узловые точки табличной функции называется интерполяционным многочленом.

Интерполирование с помощью алгебраических многочленов называется параболическим интерполированием.

Таким образом, для решения задачи интерполирования прежде всего необходимо решить задачу, которую можно сформулировать следующим образом:

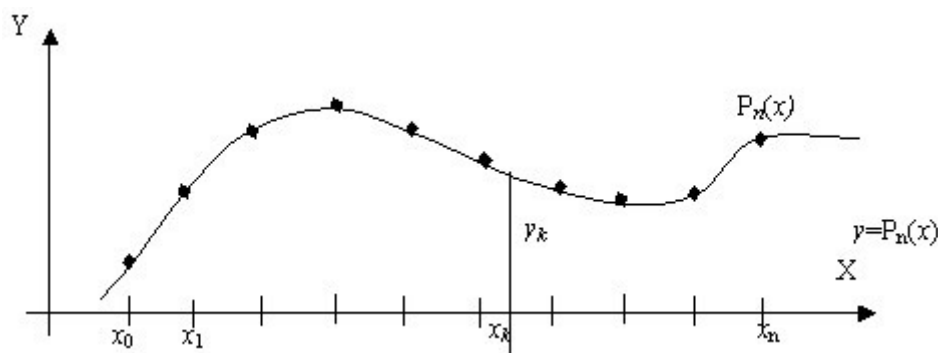
для функции $F(x_i) = y_i, i = 0, 1, 2, \dots, n$, заданной таблично, *построить интерполяционный многочлен* степени n , который проходит через все узловые точки таблицы:

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x^1 + a_n x^0 + a_n,$$

где

n -степень многочлена, равная количеству узловых точек N минус один, т.е. $n=N-1$.

В результате, в любой другой промежуточной точке x_k , расположенной внутри отрезка $[x_0, x_n]$ выполняется приближенное равенство $P_n(x_k) \approx f(x_k) \approx y_k$.



Построение интерполяционного многочлена в явном виде

Для построения интерполяционного многочлена вида (7.2) необходимо определить его коэффициенты $a_0, a_1, \dots, a_n$, т.е. $a_i, i=0, 1, 2, \dots, n$. Количество неизвестных коэффициентов равно

$$n+1=N,$$

где

n -степень многочлена (7.2),

N -количество узловых точек табличной функции (7.1).

Для нахождения коэффициентов, используем свойство (7.3) интерполяционного многочлена (7.2). На основании этого свойства интерполяционный многочлен должен пройти через каждую узловую точку (x_i, y_i) таблицы (7.1), т.е.,

$$a_0 x_i^n + a_1 x_i^{n-1} + \dots + a_{n-1} x_i + a_n = y_i, i = 0, 1, \dots, n. \quad (7.4)$$

Подставляя в (7.4) каждую узловую точку таблицы (7.1) получаем систему линейных уравнений:

[illegible]

Неизвестными системы (7.5) являются $a_0, a_1, a_2, \dots, a_n$ т.е. коэффициенты многочлена (7.2). Коэффициенты при неизвестных системы (7.5)

$x_i^n, x_i^{n-1}, \dots, x_i^0, i = 0, 1, \dots, n, i = 0, 1, \dots, n$ легко могут быть определены на основании данных таблицы (7.1).

Интерполяция по Лагранжу

Интерполяционный многочлен может быть построен при помощи специальных интерполяционных формул Лагранжа, Ньютона, Стерлинга, Бесселя и др.

Интерполяционный многочлен по формуле Лагранжа имеет вид:

$$\begin{aligned}
 L_n(x) = & \frac{(x-x_1)(x-x_2)(x-x_3)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)\dots(x_0-x_n)} \cdot y_0 + \\
 & + \frac{(x-x_0)(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)} \cdot y_1 + \\
 & + \frac{(x-x_0)(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)} \cdot y_2 + \dots \\
 & + \frac{(x-x_0)(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\dots(x_n-x_{n-1})} \cdot y_n.
 \end{aligned}
 \tag{7.6}$$

Докажем, что многочлен Лагранжа является интерполяционным многочленом, проходящим через все узловые точки, т.е. в узлах интерполирования x_i выполняется условие $L_n(x_i) = y_i$. Для этого будем последовательно подставлять значения координат узловых точек таблицы (7.1) в многочлен (7.6). В результате получим:

если $x=x_0$, то $L_n(x_0) = y_0$,

если $x=x_1$, то $L_n(x_1) = y_1$,

.....

если $x=x_n$, то $L_n(x_n) = y_n$.

Это достигнуто за счет того, что в числителе каждой дроби при соответствующем значении y_j , $j=0, 1, 2, \dots, n$ отсутствует сомножитель $(x-x_i)$, в котором $i=j$, а знаменатель каждой дроби получен заменой переменной x на соответствующее значение x_j .

Таким образом, интерполяционный многочлен Лагранжа приближает заданную табличную функцию, т.е. $L_n(x_i) = y_i$ и мы можем использовать его в

качестве вспомогательной функции для решения задач интерполирования, т.е. $L_n(x_k) \approx y_k$.

Чем больше узлов интерполирования на отрезке $[x_0, x_n]$, тем точнее интерполяционный многочлен приближает заданную табличную функцию (7.1), т.е. тем точнее равенство:

$$f(x_k) \approx L_n(x_k).$$

Однако с увеличением числа узлов интерполирования возрастает степень интерполяционного многочлена n и в результате значительно возрастает объем вычислительной работы. Поэтому при большом числе узлов необходимо применять ЭВМ. В этом случае удобно находить значения функции в промежуточных точках, не получая многочлен в явном виде.

При решении задачи экстраполирования функции с помощью интерполяционного многочлена вычисление значения функции за пределами отрезка $[x_0, x_n]$ обычно производят не далее, чем на один шаг h , равный наименьшей величине

$$|x_{i+1} - x_i|,$$

так как за пределами отрезка $[x_0, x_n]$ погрешности, как правило, увеличиваются.

Программирование формулы Лагранжа

Свернем формулу Лагранжа (7.6). В результате получим

$$L_n(x) = \sum_{j=0}^n B_j \cdot y_j,$$

где

$$B_j = \prod_{i=0}^n \frac{x - x_i}{x_j - x_i},$$

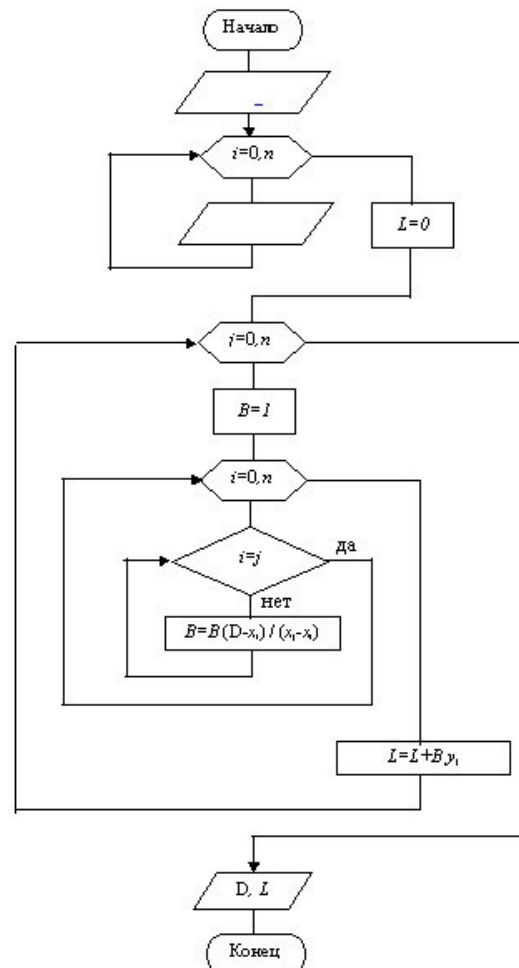
но при этом обязательно выполнение условия $i \neq j$.

При построении алгоритма используют конструкцию из двух включенных циклов:

Внешним циклом накапливаем сумму $L = \sum_{j=0}^n B_j \cdot y_j$.

Внутренним циклом накапливаем произведение $B_j = \prod_{i=0}^n \frac{x-x_i}{x_j-x_i}, i \neq j$.

Алгоритм не предусматривает получение интерполяционного многочлена в явном виде, а сразу решает задачу *интерполирования функции* в заданной точке, $x=D$.



Обозначения в алгоритме:

n – степень интерполяционного многочлена Лагранжа (7.6), равная количеству узловых точек N минус один, т.е. $n=N-1$.

D – значение аргумента в точке, для которой решается задача интерполирования табличной функции (7.1).

L – значение многочлена (7.6).

Интерполяция по Ньютону

Дана табличная функция:

i	x_i	y_i
-----	-------	-------

0	x_0	y
1	x_1	0
2	x_2	y
...	...	1
n	x_n	y
		2
		...
		y
		n

или

$$y_i = f(x_i), i = \overline{0, n}. \quad (7.1)$$

Точки с координатами (x_i, y_i) называются узловыми точками или узлами.

Количество узлов в табличной функции равно

$$N=n+1.$$

Необходимо найти значение этой функции в промежуточной точке, например, $x=D$, причем $D \in [x_0, x_n]$.

Для решения задачи строим интерполяционный многочлен.

Интерполяционный многочлен по формуле Ньютона имеет вид:

$$\begin{aligned} L_n(x) = & f(x_0) + (x-x_0) \cdot f(x_0; x_1) + \\ & + (x-x_0) \cdot (x-x_1) \cdot f(x_0; x_1; x_2) + \\ & + (x-x_0) \cdot (x-x_1) \cdot (x-x_2) \cdot f(x_0; x_1; x_2; x_3) + \dots + \\ & + (x-x_0) \cdot (x-x_1) \cdot \dots \cdot (x-x_{n-1}) \cdot f(x_0; x_1; \dots; x_n), \end{aligned} \quad (7.7)$$

где

n – степень многочлена,

$f(x_0), f(x_0; x_1), f(x_0; x_1; x_2), f(x_0; x_1; \dots; x_n)$ - разделенные разности 0-го, 1-го, 2-го, ..., n -го порядка, соответственно.

Разделенные разности

Значения $f(x_0), f(x_1), \dots, f(x_n)$, т.е. значения табличной функции в узлах, называются разделенными разностями нулевого порядка ($k=0$).

Отношение $f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ называется разделенной разностью первого порядка ($k=1$) на участке $[x_0, x_1]$ и равно разности разделенных

разностей нулевого порядка на концах участка $[x_0, x_1]$, разделенной на длину этого участка.

Для произвольного участка $[x_i, x_{i+1}]$ разделенная разность первого порядка ($k=1$) равна

$$f(x_i; x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

Отношение $f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0}$ называется разделенной разностью второго порядка ($k=2$) на участке $[x_0, x_2]$ и равно разности разделенных разностей первого порядка, разделенной на длину участка $[x_0, x_2]$.

Для произвольного участка $[x_i, x_{i+2}]$ разделенная разность второго порядка ($k=2$) равна

$$f(x_i; x_{i+1}; x_{i+2}) = \frac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i}$$

Таким образом, разделенная разность k -го порядка на участке $[x_i, x_{i+k}]$ может быть определена через *разделенные разности* $(k-1)$ -го порядка по рекуррентной формуле:

$$f(x_i; x_{i+1}; x_{i+2}; \dots; x_{i+k}) = \frac{f(x_{i+1}; x_{i+2}; \dots; x_{i+k}) - f(x_i; x_{i+1}; \dots; x_{i+k-1})}{x_{i+k} - x_i} \quad (7.8)$$

где

$$k = \overline{1, n},$$

$$k = \overline{0, n - k},$$

n – степень многочлена.

Максимальное значение k равно n . Тогда $i = 0$ и разделенная разность n -го порядка на участке $[x_0, x_n]$ равна

$$f(x_0; x_1; \dots; x_n) = \frac{f(x_1; x_2; \dots; x_n) - f(x_0; x_1; \dots; x_{n-1})}{x_n - x_0}, \text{ т.е. равна разности}$$

разделенных разностей $(n-1)$ -го порядка, разделенной на длину участка $[x_0, x_n]$.

Разделенные разности $f(x_0; x_1), f(x_0; x_1; x_2), \dots, f(x_0; x_1; \dots; x_n)$ являются вполне определенными числами, поэтому выражение (11.7) действительно является алгебраическим многочленом n -й степени. При этом в многочлене (7.7) все *разделенные разности* определены для участков $[x_0, x_{0+k}]$, $k = \overline{1, n}$.

Лемма: алгебраический многочлен (7.7), построенный по формулам Ньютона, действительно является интерполяционным многочленом, т.е. значение многочлена в узловых точках равно значению табличной функции

$$L_n(x_i) = f(x_i) = y_i; i = 0, 1, \dots, n.$$

Докажем это. Пусть $x=x_0$, тогда многочлен (7.7) равен

$$L_n(x_0) = f(x_0) = y_0.$$

Пусть $x=x_1$, тогда многочлен (7.7) равен

$$L_n(x_1) = f(x_0) + (x_1 - x_0) \cdot f(x_0; x_1) = f(x_0) + (x_1 - x_0) \cdot \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f(x_1) = y_1$$

Пусть $x=x_2$, тогда многочлен (7.7) равен

$$L_n(x_2) = f(x_0) + (x_2 - x_0) \cdot f(x_0; x_1) + (x_2 - x_0) \cdot (x_2 - x_1) \cdot f(x_0; x_1; x_2) = \\ = f(x_0) + (x_2 - x_0) \cdot \frac{f(x_1) - f(x_0)}{x_1 - x_0} + (x_2 - x_0) \cdot (x_2 - x_1) \cdot \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0} = f(x_2) = y_2$$

Заметим, что решение задачи *интерполяции по Ньютону* имеет некоторые преимущества по сравнению с решением задачи *интерполяции по Лагранжу*. Каждое слагаемое интерполяционного многочлена Лагранжа зависит от всех значений табличной функции $y_i, i=0, 1, \dots, n$. Поэтому при изменении количества узловых точек N и степени многочлена n ($n=N-1$) интерполяционный многочлен Лагранжа требуется строить заново. В многочлене Ньютона при изменении количества узловых точек N и степени многочлена n требуется только добавить или отбросить соответствующее число стандартных слагаемых в формуле Ньютона (7.7). Это удобно на практике и ускоряет процесс вычислений.

Программирование формулы Ньютона

Для построения многочлена Ньютона по формуле (7.7) организуем циклический вычислительный процесс по $k = \overline{1, n}$. При этом на каждом шаге поиска находим *разделенные разности* k -го порядка. Будем помещать *разделенные разности* на каждом шаге в массив y .

Тогда рекуррентная формула (7.8) будет иметь вид:

$$y_i = \frac{y_{i+1} - y_i}{x_{i+k} - x_i} \\ k = \overline{1, n}; \\ i = \overline{0, n - k}. \quad (7.9)$$

В формуле Ньютона (7.7) используются *разделенные разности* k -го порядка, подсчитанные только для участков $[x_0, x_{0+k}]$, т.е. *разделенные разности* k -го

порядка для $i=0$. Обозначим эти *разделенные разности* k -го порядка как y_0 . А разделенные разности, подсчитанные для $i > 0$, используются для расчетов разделенных разностей более высоких порядков.

Используя (7.9), свернем формулу (7.7). В результате получим

$$L_n(x) = y_0 + \sum_{k=1}^n P \cdot y_0^* \quad (7.10)$$

где

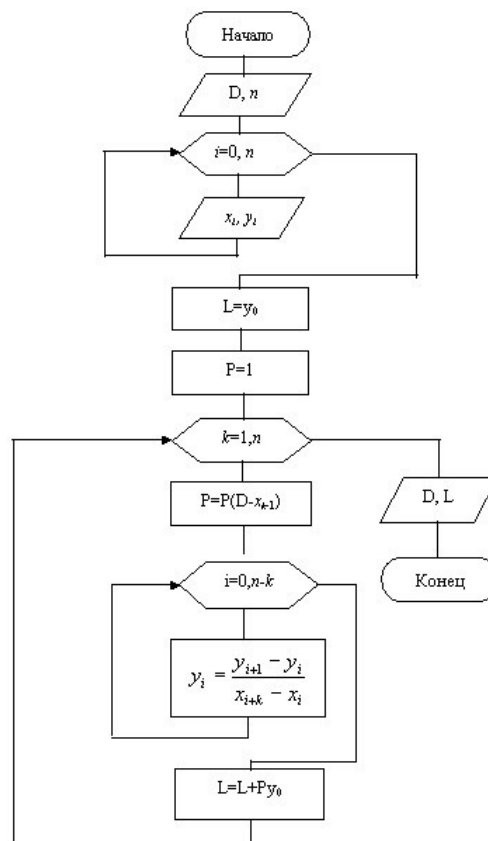
y_0 – значение табличной функции (7.1) для $x=x_0$.

y_0^* – разделенная разность k -го порядка для участка $[x_0, x_{0+k}]$.

$$P = (x-x_0)(x-x_1) \dots (x-x_{k-1}) = \prod_{j=0}^{k-1} (x-x_j).$$

Для вычисления P удобно использовать рекуррентную формулу $P = P(x - x_{k-1})$ внутри цикла по k .

Схема алгоритма *интерполяции по Ньютону* представлена на рис.7.4.



Пример интерполяции по Ньютону

Дана табличная функция:

i	x_i	y_i
0	2	0,693147
1	3	1,098613
2	4	1,386295
3	5	1,609438

Вычислить *разделенные разности* 1-го, 2-го, 3-го порядков ($n=3$) и занести их в диагональную таблицу.

Разделенные разности первого порядка:

$$f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1,098613 - 0,693147}{3 - 2} = 0,405466.$$

$$f(x_1; x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{1,386295 - 1,098613}{4 - 3} = 0,287682.$$

$$f(x_2; x_3) = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{1,609438 - 1,386295}{5 - 4} = 0,223143.$$

Разделенные разности второго порядка:

$$f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0} = \frac{0,287682 - 0,405466}{4 - 2} = -0,058892.$$

$$f(x_1; x_2; x_3) = \frac{f(x_2; x_3) - f(x_1; x_2)}{x_3 - x_1} = \frac{0,223143 - 0,287682}{5 - 3} = -0,0322695.$$

Разделенная разность третьего порядка:

$$f(x_0; x_1; x_2; x_3) = \frac{f(x_1; x_2; x_3) - f(x_0; x_1; x_2)}{x_3 - x_0} = \frac{-0,0322695 - (-0,058892)}{5 - 2} = 0,00887416$$

i	x_i	Разделенная разность			
		0-го пор.	1-го пор.	2-го пор.	3-го пор.
0	2	0,693147			
			0,405466		
1	3	1,098613		-0,058892	
			0,287682		0,00887416
2	4	1,386295		-0,0322695	
			0,223143		
3	5	1,60943			

Интерполяционный многочлен Ньютона для заданной табличной функции имеет вид:

$$L_3(x) = f(x_0) + (x - x_0) \cdot f(x_0; x_1) + (x - x_0)(x - x_1) \cdot f(x_0; x_1; x_2) + (x - x_0)(x - x_1)(x - x_2) \cdot f(x_0; x_1; x_2; x_3) = 0,693147 + (x - 2) \cdot 0,405466 + (x - 2)(x - 3) \cdot (-0,058892) + (x - 2)(x - 3)(x - 4) \cdot 0,00887416.$$

Далее полученный интерполяционный многочлен Ньютона можно привести к нормальному виду

$$L_3(x) = a_0x^3 + a_1x^2 + a_2x + a_3$$

и использовать его для решения задач интерполирования или прогноза.

Сплайн-интерполяция

Сплайны стали широко использоваться в вычислительной математике сравнительно недавно. В машиностроительном черчении они применяются уже давно, так как сплайны – это лекала или гибкие линейки, деформация которых позволяет провести кривую через заданные точки (x_i, y_i) .

Используя теорию изгиба бруса при малых деформациях, можно показать, что сплайн – это группа кубических многочленов, в местах сопряжения которых первая и вторая производные непрерывны. Такие функции называются кубическими сплайнами. Для их построения необходимо задать коэффициенты, которые единственным образом определяют многочлен в промежутке между данными точками.

Например, для некоторых функций (рис.7.5) необходимо задать все кубические функции $q_1(x)$, $q_2(x)$, ..., $q_n(x)$.

В наиболее общем случае эти многочлены имеют вид:

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3, i = \overline{1, n},$$

где k_{ij} - коэффициенты, определяемые описанными ранее условиями, количество которых равно $4n$. Для определения коэффициентов k_{ij} необходимо построить и решить систему порядка $4n$.

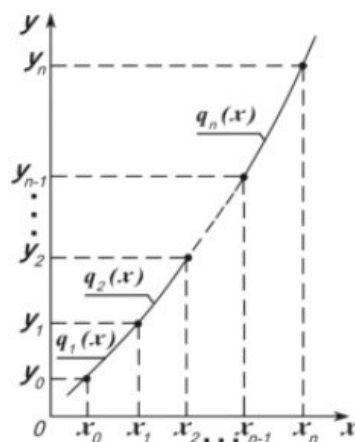


Рис. 7.5.

Первые 2п условий требуют, чтобы сплайны соприкасались в заданных точках:

$$q_i(x_i) = y_i, i = \overline{1, n};$$

$$q_{i+1}(x_i) = y_i, i = \overline{0, n-1}.$$

Следующие (2п-2) условий требуют, чтобы в местах соприкосновения сплайнов были равны первые и вторые производные:

$$q'_{i+1}(x_i) = q'_i(x_i), i = \overline{1, n-1};$$

$$q''_{i+1}(x_i) = q''_i(x_i), i = \overline{0, n-1}.$$

Система алгебраических уравнений имеет решение, если число уравнений соответствует числу неизвестных. Для этого необходимо ввести еще два уравнения. Обычно используются следующие условия:

$$q''_1(x_0) = 0; q''_n(x_n) = 0.$$

При построении алгоритма метода первые и вторые производные удобно аппроксимировать разделенными разностями соответствующих порядков.

Полученный таким образом сплайн называется естественным кубическим сплайном. Найдя коэффициенты сплайна, используют эту кусочно-гладкую полиномиальную функцию для представления данных при интерполяции.

Аппроксимация опытных данных

В результате проведения натурального эксперимента получена табличная функция:

i	x	y
0	x ₀	y ₀
1	x ₁	y ₁
2	x ₂	y ₂
3	x ₃	y ₃
...	...	...
n	x _n	y _n

где

N-количество узловых точек в таблице,

$$n=N-1.$$

Задача аппроксимации заключается в отыскании аналитической зависимости $y=f(x)$ полученной табличной функции.

В настоящее время существует 2 способа *аппроксимации опытных данных*:

Первый способ. Этот способ требует, чтобы аппроксимирующая кривая $F(x)$, аналитический вид которой необходимо найти, проходила через все узловые точки таблицы. Эту задачу можно решить с помощью *построения интерполяционного многочлена степени n* :

$$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_n. \quad (7.12)$$

Однако этот способ *аппроксимации опытных данных* имеет недостатки:

1. Точность аппроксимации гарантируется в небольшом интервале $[x_0, x_n]$ при количестве узловых точек не более 7-8.
2. Значения табличной функции в узловых точках должны быть заданы с большой точностью.

Известно, что как бы точно не проводился эксперимент, результаты эксперимента содержат погрешности. Дело в том, что на самом деле исследуемая величина зависит не только от одного аргумента x , но и от других случайных факторов, которые от опыта к опыту колеблются по своим собственным случайным законам. Этим самым обуславливается случайная колеблемость исследуемой функции.

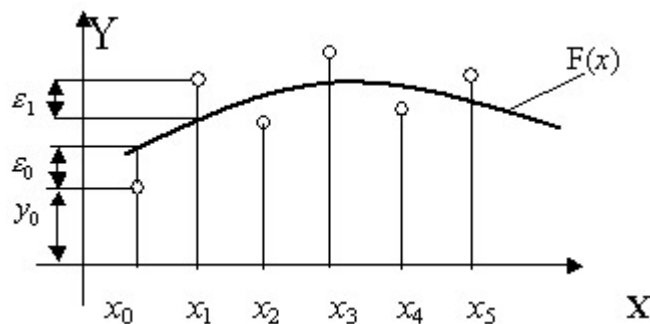
В результате аппроксимировать опытные данные с помощью интерполяционного многочлена, который проходил бы через все узловые точки таблицы, не всегда удастся. Более того, стремясь пройти через все узловые точки таблицы и увеличивая порядок многочлена, мы тем самым начинаем воспроизводить не только закономерные изменения снимаемой функции, но и ее случайные помехи.

Второй способ. На практике нашел применение другой способ *аппроксимации опытных данных* - сглаживание опытных данных. Сущность этого метода состоит в том, что табличные данные аппроксимируют кривой $F(x)$, которая не обязательно должна пройти через все узловые точки, а должна как бы сгладить все случайные помехи табличной функции.

Сглаживание опытных данных методом наименьших квадратов

В этом методе при сглаживании опытных данных аппроксимирующей кривую $F(x)$ стремятся провести так, чтобы ее отклонения ε_i от табличных данных (уклонения) по всем узловым точкам были минимальными, т.е.

$$\varepsilon_i = |F(x_i) - y_i| \rightarrow \min. \quad (7.6)$$



Избавимся от знака уклонения. Тогда условие (7.6) будет иметь вид:

$$\varepsilon_i^2 = |(F(x_i) - y_i)|^2 \rightarrow \min. \quad (7.7)$$

Суть метода наименьших квадратов заключается в следующем: для табличных данных, полученных в результате эксперимента, отыскать аналитическую зависимость $F(x)$, сумма квадратов уклонений которой от табличных данных по всем узловым точкам была бы минимальной, т.е.

$$\sum_{i=0}^n \varepsilon_i^2 = \sum_{i=0}^n (F(x_i) - y_i)^2 \rightarrow \min \quad (7.8)$$

Для определенности задачи искомую функцию $F(x)$ будем выбирать из класса алгебраических многочленов степени m :

$$P_m(x) = a_0 x^m + a_1 x^{m-1} + a_2 x^{m-2} + \dots + a_{m-1} x^1 + a_m. \quad (7.9)$$

Назовем многочлен (7.9) аппроксимирующим многочленом.

Аппроксимирующий многочлен не проходит через все узловые точки таблицы. Поэтому его степень m не зависит от числа узловых точек. При этом всегда $m < n$. Степень m может меняться в пределах $1 \leq m \leq N - 2$.

Если $m=1$, то мы аппроксимируем табличную функцию прямой линией. Такая задача называется линейной регрессией.

Если $m=2$, то мы аппроксимируем табличную функцию квадратичной параболой. Такая задача называется квадратичной аппроксимацией.

Если $m=3$, то мы аппроксимируем табличную функцию кубической параболой. Такая задача называется кубической аппроксимацией.

Уточним метод наименьших квадратов: для табличной функции, полученной в результате эксперимента, построить аппроксимирующий многочлен (7.9) степени m , для которого сумма квадратов уклонений по всем узловым точкам минимальна, т.е.

$$S = \sum_{i=1}^n (P_m(x_i) - y_i) \rightarrow \min. \quad (7.10)$$

Изменим вид многочлена P_m . Поставим на последнее место слагаемые, содержащие x^m . На предпоследнее - слагаемые, содержащие x^{m-1} и т.д. В результате получим:

$$P_m(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_mx^m. \quad (7.11)$$

ИЛИ

$$P_m(x) = \sum_{j=0}^m a_j x^j$$

При этом изменим индексы коэффициентов многочлена. Тогда условие (7.8) будет иметь вид:

$$S = \sum_{i=0}^n (a_0 x_i^0 + a_1 x_i^1 + a_2 x_i^2 + \dots + a_m x_i^m - y_i)^2 \rightarrow \min,$$

где

x_i и y_i — координаты узловых точек таблицы,

$a_j, j = \overline{0, m}$ — неизвестные коэффициенты многочлена (7.11).

Необходимым условием существования минимума функции S является равенство нулю ее частных производных по каждой a_j .

$$\left\{ \begin{array}{l} \frac{\delta s}{\delta a_0} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^{m-1} - y_i) x_i^0) = 0 \\ \frac{\delta s}{\delta a_1} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^{m-1} - y_i) x_i^1) = 0 \\ \frac{\delta s}{\delta a_2} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^{m-1} - y_i) x_i^2) = 0 \\ \dots\dots\dots \\ \frac{\delta s}{\delta a_m} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^{m-1} - y_i) x_i^m) = 0 \end{array} \right.$$

В результате получили систему линейных уравнений. Раскрывая скобки и перенося свободные члены в правой части уравнений, получим в нормальной форме систему линейных уравнений:

$$\left\{ \begin{array}{l} c_0 a_0 + c_1 a_1 + c_2 a_2 + \dots + c_m a_m = d_0, \\ c_1 a_0 + c_2 a_1 + c_3 a_2 + \dots + c_{m+1} a_m = d_1, \\ c_2 a_0 + c_3 a_1 + c_4 a_2 + \dots + c_{m+2} a_m = d_0, \\ \dots\dots\dots \\ c_m a_0 + c_{m+1} a_1 + c_{m+2} a_2 + \dots + c_{2m} a_m = d_m, \end{array} \right. \quad (7.12)$$

a_j - неизвестные системы линейных уравнений (7.12),

$$d_j = \sum_{i=0}^n y_i x_i^k, j = \overline{0, m} \quad - \text{свободные члены системы линейных уравнений} \quad (7.12),$$

Порядок системы равен $m+1$.

При ручном счете коэффициенты c_k и свободные члены d_j удобно определять, пользуясь таблицей 7.2:

i	x_i^0	x_i^1	x_i^2	$\dots$	x_i^{2m}	x_i^0	y_i	x_i^1	y_i	$\dots$	x_i^m
0	1										
1	1										
2	1										
$\dots$	$\dots$										
N	1										
$\sum_{i=0}^n$	C_0	C_1	C_2	$\dots$	C_{2m}	d_0		d_1		$\dots$	d_m

Изменим индексацию в системе (7.12). В результате получим:

где

$$c_{k,j} = \sum_{i=1}^N x_i^{k+j-2}, k = \overline{1, (m+1)}, j = \overline{1, (m+1)}$$

- коэффициенты системы
линейных уравнений (7.13),

$$d_k = \sum_{i=1}^N y_i x_i^{j-1}, j = \overline{1, (m+1)}$$

- свободные члены системы линейных уравнений (7.13),

(x_i, y_i) – координаты узловых точек табличной функции, $i = \overline{1, N}$,

N – количество узловых точек,

m – степень аппроксимирующего многочлена вида:

$$P_m(x) = a_1 x^0 + a_2 x^1 + a_3 x^2 + \dots + a_{m+1} x^m. \quad (7.14)$$

Алгоритм задачи:

1. Строим систему линейных уравнений (7.13). Определяем коэффициенты $c_{k,j}$ и свободные члены d_k . Т.к. система (7.13) симметрична относительно главной диагонали, то достаточно определить только наддиагональные элементы системы.
2. Решаем систему (7.13) методом Гаусса. Находим коэффициенты a_j многочлена (7.14).
3. Строим аппроксимирующий многочлен (7.14) и определяем его значение в каждой узловой точке $P_i = P_m(x_i)$.
4. Находим уклонение каждой узловой точки $\varepsilon_i = P_i - y_i$.
5. Находим сумму квадратов уклонений по всем узловым точкам $S = \sum_{i=1}^N \varepsilon_i^2$.
6. Находим остаточную дисперсию $D = \frac{S}{N-(m+1)}$.

Для построения аппроксимирующего многочлена (7.11) и вычисления его значения в каждой узловой точке используем рациональную форму многочлена:

$$P_m(x) = a_1 + x \cdot (a_2 + x \cdot (a_3 + \dots + x \cdot (a_m + x \cdot a_{m+1}) \dots)) \quad (7.15)$$

Тогда для вычисления значения многочлена (7.15) удобно пользоваться *схемой Горнера*. Рекуррентная формула по схеме Горнера имеет вид:

$$\begin{aligned} P &= a_{m+1}, \\ P &= a_j + x_i P, j = \overline{m, 1, -1}, i = \overline{1, N}. \end{aligned}$$

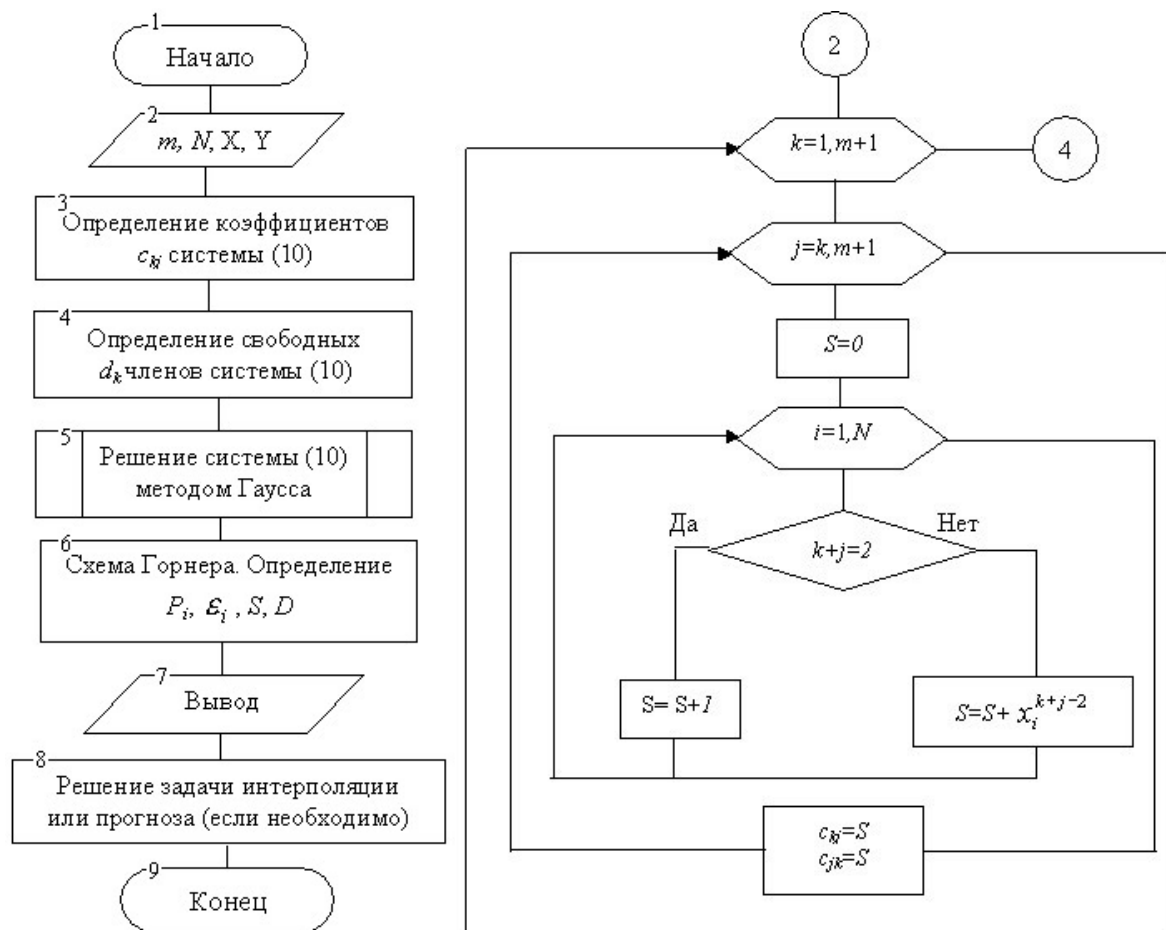
Укрупненная схема алгоритма МНК представлена на рисунке. Схемы алгоритмов основных блоков представлены на рисунках ниже.

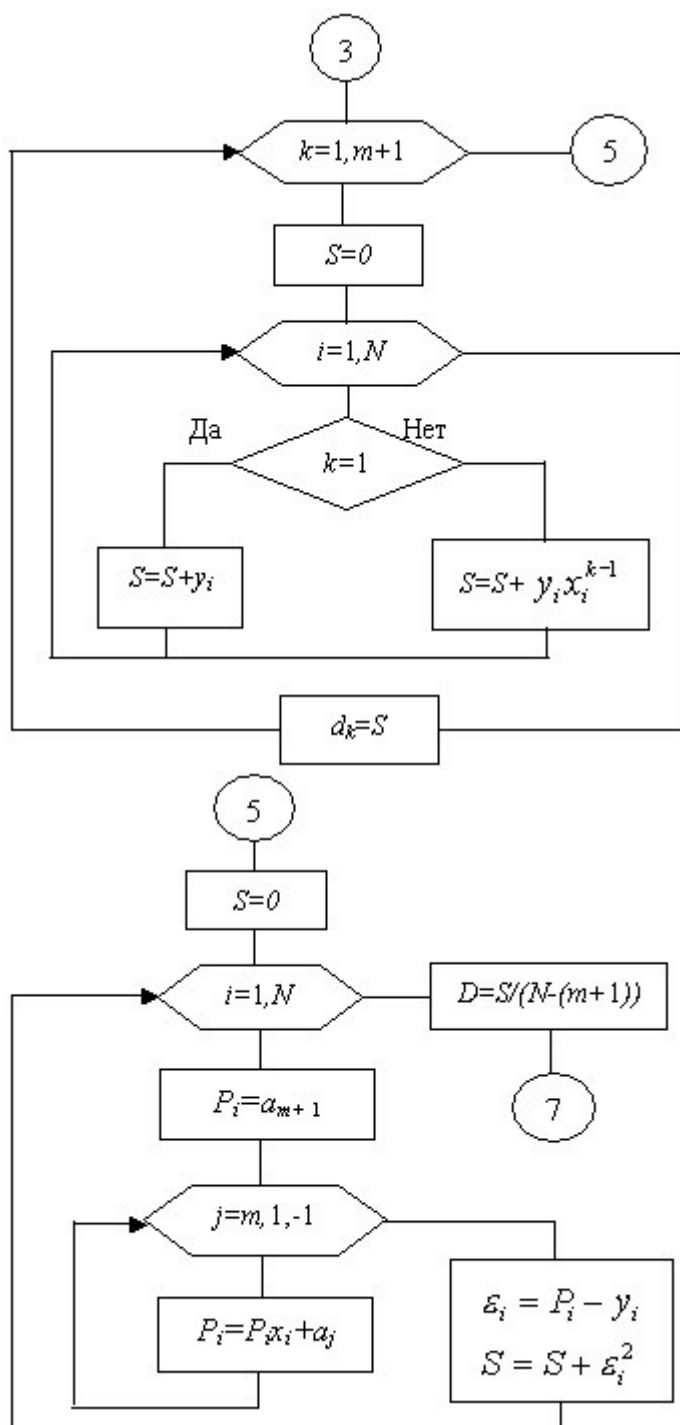
Обозначения в блоке 2:

m – степень аппроксимирующего многочлена,

N – количество узловых точек таблицы (7.2),

x, y – массивы значений x и y таблицы (7.2).





Общие вопросы моделирования

Предмет теории моделирования.

Моделирование – это замещение одного объекта (оригинала) другим (моделью) и фиксация и изучение свойств модели. Замещение производится с целью упрощения, удешевления, ускорения изучения свойств оригинала.

В общем случае объектом-оригиналом может быть естественная или искусственная, реальная или воображаемая система. Она имеет множество параметров S_0 и характеризуется определёнными свойствами. Количественной мерой свойств системы служит множество характеристик Y_0 , система проявляет свои свойства под влиянием внешних воздействий X .

Множество параметров S и их значений отражает её внутреннее содержание - структуру и принципы функционирования. Характеристики S – это в основном её внешние признаки, которые важны при взаимодействии с другими S .

Характеристики S находятся в функциональной зависимости от её параметров. Каждая характеристика системы $y_0 \in Y_0$ определяется в основном ограниченным числом параметров $\{S_{0k}\} \subset S_0$. Остальные параметры не влияют на значение данной характеристики S . Исследователя интересуют, как правило, только некоторые характеристики S : $\{y\} \subset Y_0$ при конкретных воздействиях на систему $\{x_{mn}\} \subset X$.

Модель – это тоже система со своими множествами параметров S_m и характеристик Y_m . Оригинал и модель сходны по одним параметрам и различны по другим. Замещение одного объекта другим правомерно если интересующие исследователя характеристики оригинала и модели определяются однотипными подмножествами параметров и связаны одинаковыми зависимостями с этими параметрами:

$$y_{ok} = f(\{S_{oi}\}, \{x_{on}\}, T); \quad (1.1)$$

$$y_{mn} = f(\{S_{mi}\}, \{x_{mn}\}, T_m) \quad (1.2)$$

где, y_{mn} - k -ая характеристика модели, $y_{mn} \in Y_m$

x_{mn} - внешнее воздействие на модель, $x_{mn} \in X$

T_m - модельное время.

При этом $s_{oi} = \Psi(S_{mi})$; $x_{on} = \omega(x_{mn})$, $T = mT_m$ (где m - масштабный коэффициент) на всём интервале $[0-T_m]$ или в отдельные периоды времени. Тогда с некоторым приближением можно сделать вывод о том, что характеристики O_p , связаны с характеристиками M зависимостями $y_{ok} = \varphi(y_{mk})$. Множество характеристик модели $Y_{mk} = \{y_{mk}\}$ является отображением множества интересующих характеристик оригинала $y_{ok} = \{y_{ok}\}$, т.е. $\varphi: Y_{ok} \rightarrow Y_{mk}$, т.е. $\varphi: Y_{ok} \rightarrow Y_{mk}$.

При исследовании сложных естественных S , у которых известны Y_{ok} , но мало изучен состав элементов и принципы их взаимодействия с помощью моделирования может решаться обратная задача. Строят предположительную модель, определяющая её характеристики Y_{mk} при эквивалентных внешних воздействиях $\{x_{mn}\}$ ($\omega: \{x_{on}\} \rightarrow \{x_{mn}\}$) и, если

оказывается, что имеет место отображение $\varphi: Y_{ok} \rightarrow Y_{mk}$ с некоторой известной функцией φ , то считается, что система-оригинал имеет такие же параметры.

Моделирование целесообразно, когда у модели отсутствуют те признаки оригинала, которые препятствуют его исследованию.

Теория моделирования — взаимосвязанная совокупность положений, определений, методов и средств создания моделей. Сами модели являются предметом теории моделирования.

Теория моделирования является основной составляющей общей теории систем - системологии, где в качестве главного принципа постулируются осуществимые модели: система представима конечным множеством моделей, каждая из которых отражает определённую грань её сущности.

Роль и место моделирования в исследовании систем.

Познание любой системы (S) сводится по существу к созданию её модели. Перед изготовлением каждого устройства или сооружения разрабатывается его модель - проект. Любое произведение искусства является моделью, фиксирующее действительность.

Достижения математики привели к распространению математических моделей различных объектов и процессов. Подмечено, что динамика функционирования разных по физической природе систем однотипными зависимостями, что позволяет моделировать их на ЭВМ.

На качественно новую ступень поднялась моделирование в результате разработки методологии имитационного моделирования на ЭВМ.

Сейчас трудно указать область человеческой деятельности, где бы применялось моделирование. Разработаны модели производства автомобилей, выращивания пшеницы, функционирования отдельных органов человека, жизнедеятельности Азовского моря, атомного взрыва, последствий атомной войны.

Специалисты считают, что моделирование становится основной функцией ВС. На практике широко используются АСУ технологическими процессами организационно-экономическими комплексами, процессами проектирования, банки данных и знаний. Но любая из этих систем нуждается в информации об управляемом объекте и модели управляемого объекта, в моделировании тех или иных управляющих решений.

Сами ВС как сложные и дорогостоящие технические системы могут являться объектами моделирования.

Обычно процесс разработки сложной системы осуществляется итерационно с использованием моделирования проектных решений. Если характеристики не удовлетворяют предъявленным требованиям, то по результатам анализа производят корректировку проекта, затем снова проводят моделирование. При анализе действующих систем с помощью моделирования определяют границы работоспособности системы, выполняют имитацию экспериментальных условий, которые могут возникнуть в процессе функционирования системы. Искусственное создание таких условий на

действительной системе затруднено и может привести к катастрофическим последствиям.

Применение моделирования может быть полезным при разработке стратегии развития ВС, её усовершенствования при создании сетей ЭВМ.

Классификация моделей.

Физические модели. В основу классификации положена степень абстрагирования модели от оригинала. Предварительно все модели можно подразделить на 2 группы — физические и абстрактные (математические).

Ф.М. обычно называют систему, эквивалентную или подобную оригиналу, но возможно имеющую другую физическую природу. Виды Ф.М.:

- натуральные;
- квазинатуральные;
- масштабные;
- аналоговые;

Натуральные модели — это реальные исследуемые системы (макеты, опытные образцы). Имеют полную адекватность (соответствия) с системой оригиналом, но дороги.

Квазинатуральные модели — совокупность натуральных и математических моделей. Этот вид используется тогда, когда модель части системы не может быть математической из-за сложности её описания (модель человека оператора) или когда часть системы должна быть исследована во взаимодействии с другими частями, но их ещё не существует или их включение очень дорого. (вычислительные полигоны, АСУ)

Масштабная модель — это система той же физической природы, что и оригинал, но отличается от него масштабами. Методологической основой масштабного моделирования является теория подобия. При проектировании ВС масштабные модели могут использоваться для анализа вариантов компоновочных решений.

Аналоговые модели называют системы, имеющие физическую природу, отличающуюся от оригинала, но сходные с оригиналом процессы функционирования. Для создания аналоговой модели требуется наличие математического описания изучаемой системы. В качестве аналоговых моделей используются механические, гидравлические, пневматические и электрические системы. Аналоговое моделирование использует при исследовании средства ВТ на уровне логических элементов и электрических цепей, а так же на системном уровне, когда функционирование системы описывается например, дифференциальными или алгебраическими уравнениями.

Математические модели. Математические модели представляют собой формализованное представление системы с помощью абстрактного языка, с помощью математических соотношений, отражающих процесс функционирования системы. Для составления математических моделей можно использовать любые математические средства — алгебраическое,

дифференциальное, интегральное исчисления, теорию множеств, теорию алгоритмов и т.д. По существу вся математика создана для составления и исследования моделей объектов и процессов.

К средствам абстрактного описания систем относятся также языки химических формул, схем, чертежей, карт, диаграмм и т.п. Выбор вида модели определяется особенностями изучаемой системы и целями моделирования, т.к. исследование модели позволяет получить ответы на определённую группу вопросов. Для получения другой информации может потребоваться модель другого вида. Математические модели можно классифицировать на детерминированные и вероятностные, аналитические, численные и имитационные.

Аналитической моделью называется такое формализованное описание системы, которое позволяет получить решение уравнения (1.2) в явном виде, используя известный математический аппарат.

Численная модель характеризуется зависимостью (1.2) такого вида, который допускает только частные решения для конкретных начальных условий и количественных параметров моделей.

Имитационная модель – это совокупность описания системы и внешних воздействий, алгоритмов функционирования системы или правил изменения состояния системы под влиянием внешних и внутренних возмущений. Эти алгоритмы и правила не дают возможности использования имеющихся математических методов аналитического и численного решения, но позволяют имитировать процесс функционирования системы и производить вычисления интересующих характеристик. Имитационные модели могут быть созданы для гораздо более широкого класса объектов и процессов, чем аналитические и численные. Поскольку для реализации имитационных моделей служат ВС, средствами формализованного описания ИМ служат универсальные и специальные алгоритмические языки. ИМ в наибольшей степени подходят для исследования ВС на системном уровне.

Математические схемы моделирования систем.

Основные подходы к построению ММ систем.

Исходной информацией при построении ММ процессов функционирования систем служат данные о назначении и условиях работы исследуемой (проектируемой) системы S . Эта информация определяет основную цель моделирования, требования к ММ, уровень абстрагирования, выбор математической схемы моделирования.

Понятие **математическая схема** позволяет рассматривать математику не как метод расчёта, а как метод мышления, средства формулирования понятий, что является наиболее важным при переходе от словесного описания к формализованному представлению процесса её функционирования в виде некоторой ММ.

При пользовании мат. схемой в первую очередь исследователя системы должен интересовать вопрос об адекватности отображения в виде

конкретных схем реальных процессов в исследуемой системе, а не возможность получения ответа (результата решения) на конкретный вопрос исследования.

Например, представление процесса функционирования ИВС коллективного пользования в виде сети схем массового обслуживания даёт возможность хорошо описать процессы, происходящие в системе, но при сложных законах входящих потоков и потоков обслуживания не даёт возможности получения результатов в явном виде.

Математическую схему можно определить как звено при переходе от содержательного к формализованному описанию процесса функционирования системы с учётом воздействия внешней среды. Т.е. имеет место цепочка: описательная модель – математическая схема – имитационная модель.

Каждая конкретная система S характеризуется набором свойств, под которыми понимаются величины, отображающие поведение моделируемого объекта (реальной системы) и учитываются условия её функционирования во взаимодействии с внешней средой (системой) E .

При построении ММ системы S необходимо решить вопрос о её полноте. Полнота моделирования регулируется, в основном, выбором границ "Система S — среда E ". Также должна быть решена задача упрощения ММ, которая помогает выделить основные свойства системы, отбросив второстепенные в плане цели моделирования.

ММ объекта моделирования, т.е. системы S можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

- совокупность X - входных воздействий на S $x_i \in X$, $i=1 \dots n_x$;
- совокупность воздействий внешней среды $v_l \in V$, $l=1 \dots n_v$;
- совокупность внутренних (собственных) параметров системы $h_k \in H$, $k=1 \dots n_h$;
- совокупность выходных характеристик системы $y_j \in Y$, $j=1 \dots n_y$.

В перечисленных множествах можно выделить управляемые и неуправляемые величины. В общем случае X , V , H , Y не пересекаемые множества, содержат как детерминированные так и стохастические составляющие. Входные воздействия E и внутренние параметры S являются независимыми (экзогенными) переменными, $\bar{X}(t); \bar{V}(t); \bar{H}(t)$. Выходные характеристики - зависимые переменные (эндогенные) $\bar{Y}(t)$. Процесс функционирования S описывается оператором F_s :

$$\bar{Y}(t) = F_s(\bar{X}, \bar{V}, \bar{H}, t) \quad (1)$$

$\bar{Y}(t)$ - выходная траектория. F_s - закон функционирования S . F_s может быть функция, функционал, логические условия, алгоритм, таблица или словесное описание правил.

Алгоритм функционирования A_s – метод получения выходных характеристик $\bar{Y}(t)$ с учётом входных воздействий $\bar{X}(t); \bar{V}(t); \bar{H}(t)$. Очевидно один и тот же F_s

может быть реализован различными способами, т.е. с помощью множества различных A_S .

Соотношение (1) является математическим описанием поведения объекта S моделирования во времени t , т.е. отражает его динамические свойства. (1) - это динамическая модель системы S . Для статических условий ММ есть отображения X, V, H в Y , т.е. $\bar{Y} = f(\bar{X}, \bar{V}, \bar{H})$ (2)

Соотношения (1), (2) могут быть заданы формулами, таблицами и т.д.

Также соотношения в ряде случаев могут быть получены через свойства системы в конкретные моменты времени, называемые состояниями.

Состояния системы S характеризуются векторами:

$\bar{Z}^I(\bar{z}_1^I, \dots, \bar{z}_k^I)$ и $\bar{Z}^{II}(\bar{z}_1^{II}, \dots, \bar{z}_k^{II})$, где $\bar{z}_1^I = z_1(t^I) \dots \bar{z}_k^I = z_k(t^I)$ в момент $t^I \in (t_0, T)$
 $\bar{z}_1^{II} = z_1(t^{II}) \dots \bar{z}_k^{II} = z_k(t^{II})$ в момент $t^{II} \in (t_0, T)$ и т.д. $k=1 \dots n_Z$.

$Z_1(t), Z_2(t) \dots Z_k(t)$ - это координаты точки в k -мерном фазовом пространстве. Каждой реализации процесса будет соответствовать некоторая фазовая траектория.

Совокупность всех возможных значений состояний $\{\bar{Z}\}$ называется пространством состояний объекта моделирования Z , причём $z_k \in Z$.

Состояние системы S в интервале времени $t_0 < t \leq T^I$ полностью определяется начальными условиями $\bar{Z}^0 = (\bar{z}_1^0, \dots, \bar{z}_k^0)$, где $\bar{z}_1^0 = z_1(t_0) \dots$ входными $\bar{X}(t)$, внутренними параметрами $\bar{H}(t)$ и воздействиями внешней среды $\bar{V}(t)$, которые имели место за промежуток времени $t^* - t_0$ с помощью 2-х векторных уравнений:

$$\bar{Z}(t) = \Phi(\bar{Z}^0, \bar{X}, \bar{V}, \bar{h}, t); \quad (3)$$

$$\bar{Y}(t) = F(\bar{Z}, t). \quad (4)$$

$$\text{иначе:} \quad Y(t) = F(\Phi(\bar{Z}^0, \bar{X}, \bar{V}, \bar{h}, t)). \quad (5)$$

Время в мод. S может рассматриваться на интервале моделирования (t_0, T) как непрер., так и дискретное, т.е. квантованное на отрезке длин. Δt .

Таким образом под ММ объекта понимаем конечное множество переменных $\{\bar{X}, \bar{Z}, \bar{h}\}$ вместе с математическими связями между ними и характеристиками $\bar{Y}$.

Моделирование называется детерминированным, если операторы F, Φ детерминированные, т.е. для конкретного входа выход детерминированный. Детерминированное моделирование – частный случай стохастического моделирования. В практике моделирование объектов в области системного анализа на первичных этапах исследования рациональнее использовать типовые математические схемы: диф. уравнения, конечные и вероятностные автоматы, СМО и т.д.

Не облад. такой степенью общности, как модели (3), (4), типовые математические схемы имеют преимущество простоты и наглядности, но при существенном сужении возможности применения.

В качестве детерминированных моделей, когда при исследовании случайный факт не учитывается, для представления систем, функционирующих в непрерывном времени, используются дифференциальные, интегральные и др.

уравнения, а для представления систем, функционирующих в дискретном времени – конечные автоматы и конечно разностные схемы.

В начале стохастических моделей (при учёте случайного фактора) для представления систем с дискретным временем используются вероятностные автоматы, а для представления систем с непрерывным временем – системы массового обслуживания (СМО). Большое практическое значение при исследовании сложных индивидуальных управленческих систем, к которым относятся АСУ, имеют так называемые агрегативные модели.

Агрегативные модели (системы) позволяют описать широкий круг объектов исследования с отображением системного характера этих объектов. Именно при агрегативном описании сложный объект расчленяется на конечное число частей (подсистем), сохраняя при этом связи, обеспечивая взаимодействие частей.

Непрерывно детерминированные модели (Д - схемы).

Рассмотрим особенности непрерывно детерминированного подхода на примере, используя в качестве ММ дифференциальные уравнения.

Дифференциальными уравнениями называются такие уравнения, в которых неизвестными будут функции одной переменной или нескольких переменных, причём в уравнение входят не только их функции но их производные различных порядков.

Если неизвестные - функции многих переменных, то уравнения называются — уравнения в частных производных. Если неизвестные функции одной независимой переменной, то имеют место обыкновенные дифференциальные уравнения.

Математическое соотношение для детерминированных систем в общем виде:

$$\vec{y}' = \vec{f}(\vec{y}, t), \vec{y}(t_0) = \vec{y}_0 \quad (7).$$

Например, процесс малых колебаний маятника описан обыкновенными

дифференциальным уравнением $m_1 l_1^2 \frac{\partial^2 \theta(t)}{\partial t^2} + m_2 g l \theta(t) = 0$ где m_1, l_1 - масса,

длина подвески маятника, θ - угол отклонения маятника от положения равновесия. Из этого уравнения можно найти оценки интересующих характеристик, например период колебаний $T = 2\pi \sqrt{l/g}$

Диф. уравнения, Д - схемы являются математическим аппаратом теории систем автоматического регулирования, управления.

При проектировании и эксплуатации систем САУ необходимо выбрать такие параметры системы, которые бы обеспечивали требуемую точность управления.

Следует отметить, что часто используемые в САУ системы диф. уравнений определяются путём линеаризацией управления объекта (системы), более сложного вида, имеющего нелинейности: $F(y^n, y^{n-1}, \dots, y, x^m, x^{m-1}, \dots, x_n) = 0$:

$$\frac{dF}{dy^n} \Delta y^n + \frac{dF}{dy^{n-1}} \Delta y^{n-1} \dots \frac{dF}{dy} \Delta y + \Delta y = \frac{dF}{dx^m} \Delta x^m + \dots \frac{dF}{dx_0} \Delta x + \Delta x$$

Дискретно-детерминированные модели (F-схемы)

ДДМ являются предметом рассмотрения теории автоматов (ТА). ТА – раздел теоретической кибернетики, изучающей устройства, перерабатывающие дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени.

Конечный автомат имеет множество внутренних состояний и входных сигналов, являющихся конечными множествами. Автомат задаётся F-схемой: $F = \langle Z, X, Y, \varphi, \psi, z_0 \rangle$, (1)

где Z, X, Y – соответственно конечные множества входных, выходных сигналов (алфавитов) и конечное множество внутренних состояний (алфавита). $z_0 \in Z$ – начальное состояние; $\varphi(z, x)$ – функция переходов; $\psi(z, x)$ – функция выхода. Автомат функционирует в дискретном автоматном времени, моментами которого являются такты, т.е. примыкающие друг к другу равные интервалы времени, каждому из которых соответствуют постоянные значения входного, выходного сигнала и внутреннего состояния. Абстрактный автомат имеет один входной и один выходной каналы.

В момент t , будучи в состоянии $z(t)$, автомат способен воспринять сигнал $x(t)$ и выдать сигнал $y(t) = \psi[z(t), x(t)]$, переходя в состояние $z(t+1) = \varphi[z(t), x(t)]$, $z(t) \in Z$; $y(t) \in Y$; $x(t) \in X$. Абстрактный КА в начальном состоянии z_0 принимая сигналы $x(0), x(1), x(2) \dots$ выдаёт сигналы $y(0), y(1), y(2) \dots$ (выходное слово). Существуют F-автомат 1-ого рода (Миля), функционирующий по схеме:

$$z(t+1) = \varphi[z(t), x(t)], t=0, 1, 2, \dots \quad (1)$$

$$y(t) = \psi[z(t), x(t)], t=0, 1, 2, \dots \quad (2)$$

F-автомат 2-ого рода:

$$z(t+1) = \varphi[z(t), x(t)], t=0, 1, 2, \dots \quad (3)$$

$$y(t) = \psi[z(t), x(t-1)], t=1, 2, 3, \dots \quad (4)$$

$$\text{Автомат 2-ого рода, для которого } y(t) = \psi[z(t)], t=0, 1, 2, \dots \quad (5)$$

т.е. функция выходов не зависит от входной переменной $x(t)$, называется автоматом Мура.

Т.о. уравнения 1-5 полностью задающие F-автомат, являются частным случаем уравнения

$$\vec{z}(t) = \Phi(\vec{z}_0, \vec{x}, \vec{v}, \vec{h}, t) \quad (6)$$

где $\vec{z}$ – вектор состояния, $\vec{x}$ – вектор независимых входных переменных, $\vec{v}$ – вектор воздействий внешней среды, $\vec{h}$ – вектор собственных внутренних параметров системы, $\vec{z}_0$ – вектор начального состояния, t – время; и уравнение $\vec{y}(t) = F(\vec{z}, t)$, (7)

когда система S – денормированная и на её вход поступает дискретный сигнал x .

По числу состояний конечные автоматы бывают с памятью и без памяти. Автоматы с памятью имеют более одного состояния, а автоматы без памяти (комбинационные или логические схемы) обладают лишь одним состоянием. При этом согласно (2), работа комбинационной схемы заключается в том, что она ставит в соответствие каждому входному сигналу $x(t)$ определённый выходной сигнал $y(t)$, т.е. реализует логическую функцию вида:

$$y(t)=\psi[x(t)], t=0,1,2,\dots$$

Эта функция называется булевой, если алфавиты X и Y , которым принадлежат значения сигналов x и y состоят из 2-х букв.

По характеру отсчёта времени (дискретному) F - автоматы делятся на синхронные и асинхронные. В синхронных автоматах моменты времени, в которые автомат "считывает" входные сигналы, определяются принудительно синхронизирующими сигналами. Реакция автомата на каждое значение входного сигнала заканчивается за один такт синхронизации.

Асинхронный F - автомат считывает входной сигнал непрерывно и поэтому, реагируя на достаточно длинный входной сигнал постоянной величины x , он может, как это следует из 1-5, несколько раз изменить своё состояние, выдавая соответствующее число выходных сигналов, пока не перейдёт в устойчивое.

Для задания F - автомата необходимо описать все элементы множества $F=\langle z,x,y,\varphi,\psi,z_0\rangle$, т.е. входной, внутренний и выходной алфавиты, а также функции переходов и выходов. Для задания работы F - автоматов наиболее часто используются табличный, графический и матричный способ.

В табличном способе задания используется таблицы переходов и выходов, строки которых соответствуют входным сигналам автомата, а столбцы - его состояниям. При этом обычно 1-ый столбец слева соответствует начальному состоянию z_0 . На пересечении i -ой строки и j -ого столбца таблицы переходов помещается соответствующее значение $\varphi(z_k, x_i)$ функции переходов, а в таблице выходов - $\psi(z_k, x_i)$ функции выходов. Для F - автомата Мура обе таблицы можно совместить, получив т.н. отмеченную таблицу переходов, в которой над каждым состоянием z_k автомата, обозначающим столбец таблицы, стоит соответствующий этому состоянию, согласно (5), выходной сигнал $\psi(z_i)$.

Описание работы F - автомата Мили таблицами переходов φ и выходов ψ иллюстрируется таблицей (1), а описание F - автомата Мура - таблицей переходов (2).

Таблица 1

X_j	Z_k			
	Z_0	Z_1	...	Z_k
Переходы				
X_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$	...	$\varphi(z_k, x_1)$
X_2	$\varphi(z_0, x_2)$	$\varphi(z_1, x_2)$	...	$\varphi(z_k, x_2)$
.....				
X_l	...	...	...	...
Выходы				
X_1	$\psi(z_0, x_1)$	$\psi(z_1, x_1)$	...	$\psi(z_k, x_1)$
.....				

x_1	$\psi(z_0, x_1)$	$\psi(z_1, x_1)$	...	$\psi(z_k, x_1)$
-------	------------------	------------------	-----	------------------

Таблица 2

x_i	$\psi(z_k)$			
	$\psi(z_0)$	$\psi(z_1)$	...	$\psi(z_k)$
	z_0	z_1	...	z_k
x_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$	...	$\varphi(z_k, x_1)$
x_2	$\varphi(z_0, x_2)$	$\varphi(z_1, x_2)$	...	$\varphi(z_k, x_2)$
.....				
x_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$	...	$\varphi(z_k, x_1)$

Примеры табличного способа задания F- автомата Мили F1 с тремя состояниями, двумя входными и двумя выходными сигналами приведены в таблице 3, а для F- автомата Мура F2 - в таблице 4.

Таблица 3

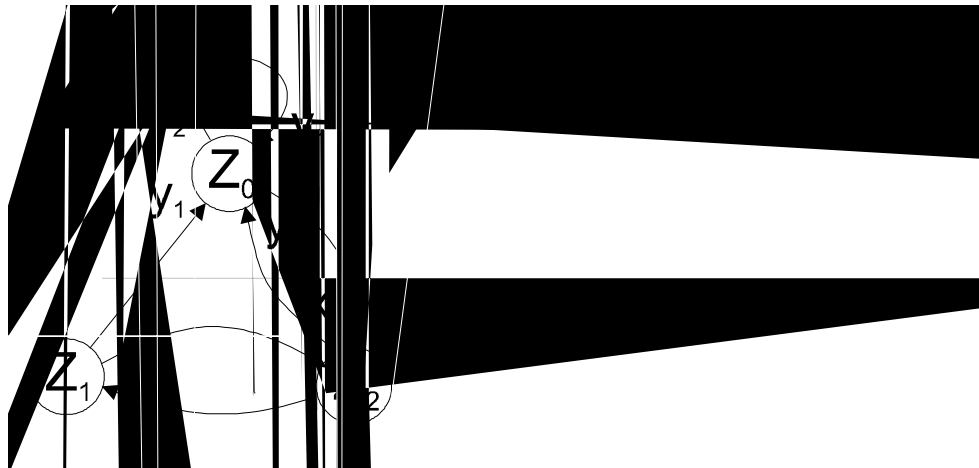
x_j	z_0		
	z_0	z_1	z_2
Переходы			
x_1	z_2	z_0	z_0
x_2	z_0	z_2	z_1
Выходы			
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1

Таблица 4

x_i	y				
	y_1	y_1	y_3	y_2	y_3
	z_0	z_1	z_2	z_3	z_4
x_1	z_1	z_4	z_4	z_2	z_2
x_2	z_3	z_1	z_1	z_0	z_0

При другом способе задания конечного автомата используется понятие направленного графа. Граф автомата представляет собой набор вершин, соответствующих различным состояниям автомата и соединяющих вершин дуг графа, соответствующих тем или иным переходам автомата. Если входной сигнал x_k вызывает переход из состояния z_i в состояние z_j , то на графе автомата дуга, соединяющая вершину z_i с вершиной z_j обозначается x_k . Для того, чтобы задать функцию переходов, дуги графа необходимо отметить соответствующими выходными сигналами. Для автоматов Мили эта разметка производится так: если входной сигнал x_k действует на состояние z_i , то согласно сказанному получается дуга, исходящая из z_i и помеченная x_k ; эту дугу дополнительно отмечают выходным сигналом $y=\psi(z_i, x_k)$. Для автомата

Мура аналогичная разметка графа такова: если входной сигнал x_k , действуя на некоторое состояние автомата, вызывает переход в состояние z_j , то дугу, направленную в z_j и помеченную x_k , дополнительно отмечают выходным сигналом $y = \psi(z_j, x_k)$. На рис. 1 приведены заданные ранее таблицами F-автоматы Мили F1 и Мура F2 соответственно.



$$C_2 = \begin{pmatrix} - & x_1 & - & x_2 & - \\ - & x_2 & - & - & x_1 \\ - & x_2 & - & - & x_1 \\ x_2 & - & x_1 & - & - \\ x_2 & - & x_1 & - & - \end{pmatrix} ; \quad \bar{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_2 \\ y_3 \end{bmatrix}$$

Для детерминированных автоматов переходы однозначны. Применительно к графическому способу задания F- автомата это означает, что в графе F- автомата из любой вершины не могут выходить 2 и более ребра, отмеченные одним и тем же входным сигналом. Аналогично этому в матрице соединений автомата С в каждой строке любой входной сигнал не должен встречаться более одного раза.

Рассмотрим вид таблицы переходов и графа асинхронного конечного автомата. Для F- автомата состояние z_k называется устойчивым, если для любого входа $x_i \in X$, для которого $\varphi(z_k, x_i) = z_k$ имеет место $\psi(z_k, x_i) = y_k$. Т.о. F- автомат называется асинхронным, если каждое его состояние $z_k \in Z$ устойчиво.

На практике всегда автоматы являются асинхронными, а устойчивость их состояний обеспечивается тем или иным способом, например, введением сигналов синхронизации. На уровне абстрактной теории удобно часто оперировать с синхронными конечными автоматами.

Пример. Рассмотрим асинхронный F- автомат Мура, который описан в табл. 5 и приведён на рис. 2.

Таблица 5

x_i	y		
	y_1	y_2	y_3
	z_0	z_1	z_2
x_1	z_1	z_1	z_1
x_2	z_2	z_1	z_2
x_3	z_0	z_0	z_2

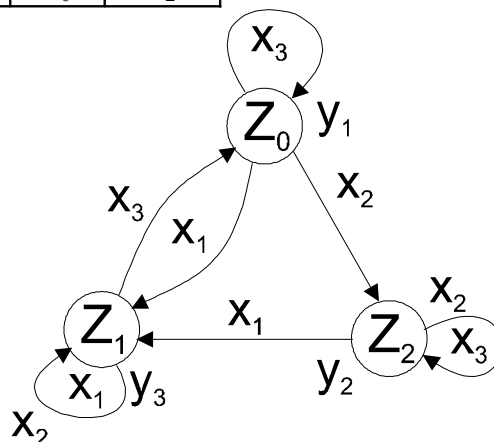
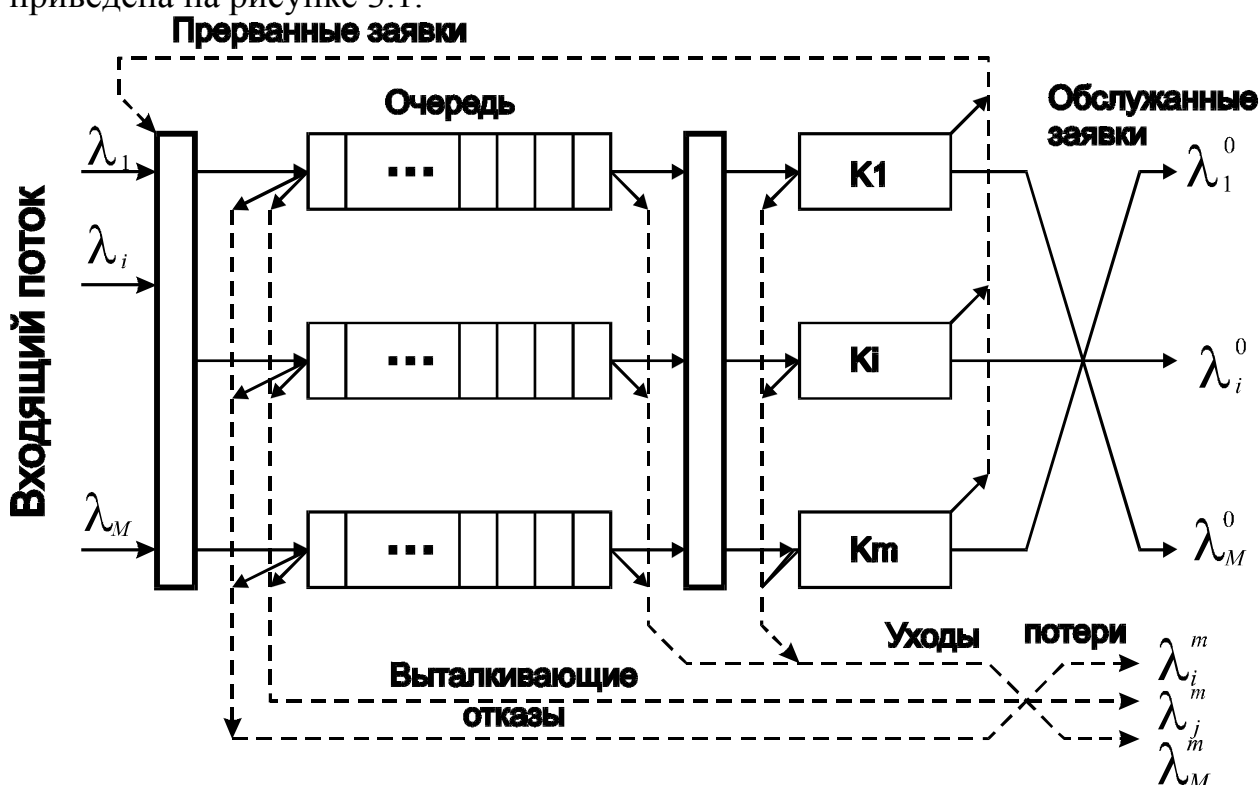


Рис. 2. Граф асинхронного автомата Мура.

Непрерывно-стохастические модели (Q - схемы).
К ним относятся системы массового обслуживания (англ. queuing system), которые называют Q- схемами.

Предмет ТМО – системы массового обслуживания (СМО) и сети массового обслуживания. Под СМО понимают динамическую систему, предназначенную для эффективного обслуживания случайного потока заявок при ограниченных ресурсах системы. Обобщённая структура СМО приведена на рисунке 3.1.



Поступающие на вход СМО однородные заявки в зависимости от порождающей причины делятся на типы, интенсивность потока заявок типа i ($i=1 \dots M$) обозначено λ_i . Совокупность заявок всех типов - входящий поток СМО.

74

универсального канала типа j считается известными функции распределения $F_{ji}(\tau)$ длительности обслуживания заявок произвольного типа. Для специализированных каналов функции распределения длительности обслуживания каналов заявок некоторых типов являются неопределёнными, назначение этих заявок на данный канал.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например, потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удалённых терминалов и т.д. При этом характерным для работы таких объектов является случайное поведение заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени.

Q - схемы можно исследовать аналитически и имитационными моделями. Последнее обеспечивает большую универсальность.

Рассмотрим понятие массового обслуживания.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно отобразить в виде некоторого i -ого прибора обслуживания Π_i , состоящего из накопителя заявок, в котором может находиться одновременно $l_i = 0 \dots L_i^H$ заявок, где L_i^H - ёмкость i -ого накопителя, и канала обслуживания заявок, k_i .

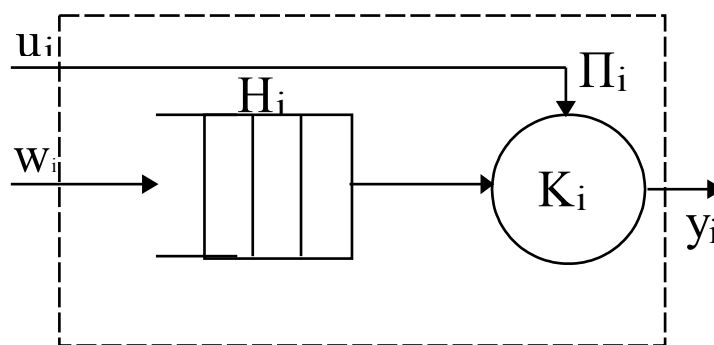


Рис. 3.2. Схема прибора СМО

На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i поток заявок w_i , на канал k_i - поток обслуживания u_i .

Потоком событий (ПС) называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Однородный ПС характеризуется только моментами поступления этих событий (вызывающими моментами) и задаётся последовательностью $\{t_n\} = \{0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \dots\}$, где t_n - момент поступления n -ого события - неотрицательное вещественное число. ОПС может быть также задан в виде

последовательности промежутков времени между n -ым и $n-1$ -ым событиями $\{\tau_n\}$.

Неоднородным ПС называется последовательность $\{t_n, f_n\}$, где t_n - вызывающие моменты; f_n - набор признаков события. Например, может быть задана принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т.п. Рассмотрим ОПС, для которого $\tau_i \in \{\tau_n\}$ - случайные величины, независимые между собой. Тогда ПС называется потоком с ограниченным последствием.

ПС называется ординарным, если вероятность того, что на малый интервал времени Δt , примыкающий к моменту времени t попадает больше одного события $P_{\geq 1}(t, \Delta t)$ пренебрежительно мала.

Если для любого интервала Δt событие $P_0(t, \Delta t) + P_1(t, \Delta t) + P_{\geq 1}(t, \Delta t) = 1$, $P_1(t, \Delta t)$ - вероятность попадания на интервал Δt ровно одного события. Как сумма вероятностей событий, образующих полную группу и несовместных, то для ординарного потока событий $P_0(t, \Delta t) + P_1(t, \Delta t) \approx 1$, $P_{\geq 1}(t, \Delta t) = \Theta(\Delta t)$, где $\Theta(\Delta t)$ - величина, порядок малости который выше, чем Δt , т.е. $\lim(\Theta(\Delta t)) = 0$ при $\Delta t \rightarrow 0$.

Стационарным ПС называется поток, для которого вероятность появления того или иного числа событий на интервале времени τ зависит от длины этого участка и не зависит от того, где на оси времени $0 - t$ взят этот участок. Для ОПС справедливо $0 * P_0(t, \Delta t) + 1 * P_1(t, \Delta t) = P_1(t, \Delta t)$ - среднее число событий на интервале Δt . Среднее число событий, наступающих на участке Δt в единицу времени составляет $P_1(t, \Delta t) / \Delta t$. Рассмотрим предел этого выражения при $\Delta t \rightarrow 0$

$$\lim P_1(t, \Delta t) / \Delta t = \lambda(t) * (1/\text{един.вр.}).$$

Если этот предел существует, то он называется интенсивностью (плотностью) ОПС. Для стандартного ПС $\lambda(t) = \lambda = \text{const}$.

Применительно к элементарному каналу обслуживания k_i можно считать что поток заявок $w_i \in W$, т.е. интервалы времени между моментами появления заявок на входе k_i образуют подмножество неуправляемых переменных, а поток обслуживания $u_i \in U$, т.е. интервалы времени между началом и окончанием обслуживания заявки образуют подмножество управляемых переменных.

Заявки, обслуженные каналом k_i и заявки, покинувшие прибор Π_i по различным причинам не обслуженными, образуют выходной поток $y_i \in Y$. Процесс функционирования прибора обслуживания Π_i можно представить как процесс изменения состояний его элементов во времени $Z_i(t)$. Переход в новое состояние для Π_i означает изменение кол-ва заявок, которые в нём находятся (в канале k_i и накопителе H_i). Т.о. вектор состояний для Π_i имеет вид: $\vec{Z}_i = (z_i^H, z_i^K)$, где z_i^H - состояния накопителя, ($z_i^H = 0$ - накопитель пуст, $z_i^H = 1$ - в накопителе одна заявка..., $z_i^H = L_i^H$ - накопитель занят полностью; z_i^K - состояние канала k_i ($z_i^K = 0$ - канал свободен, $z_i^K = 1$ канал занят).

Q-схемы реальных объектов образуются композицией многих элементарных приборов обслуживания Π_i . Если k_i различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная Q-схема), а если приборы Π_i и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная Q-схема).

Т.о. для задания Q-схемы необходимо оператор сопряжения R , отражающий взаимосвязь элементов структуры.

Связи в Q-схеме изображают в виде стрелок (линий потока, отражающих направление движения заявок). Различают разомкнутые и замкнутые Q-схемы. В разомкнутой выходной поток не может снова поступить на какой-либо элемент, т.е. обратная связь отсутствует.

Собственными (внутренними) параметрами Q-схемы будут являться кол-во фаз L_Φ , количество каналов в каждой фазе, L_{kj} , $j=1 \dots L_\Phi$, количество накопителей каждой фазы L_{kj} , $k=1 \dots L_\Phi$, ёмкость i -ого накопителя L_i^H .

Следует отметить, что в теории массового обслуживания в зависимости от ёмкости накопителя применяют следующую терминологию:

- системы с потерями ($L_i^H=0$, накопитель отсутствует);
- системы с ожиданием ($L_i^H \rightarrow \infty$);
- системы с ограниченной ёмкостью накопителя N_i (смешанные).

Обозначим всю совокупность собственных параметров Q-схемы как подмножество H .

Для задания Q-схемы также необходимо описать алгоритмы её функционирования, которые определяют правила поведения заявок в различных неоднозначных ситуациях.

В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе N_i и обслуживания заявок каналом k_i . Неоднородность потока заявок учитывается с помощью введения класса приоритетов.

В зависимости от динамики приоритетов Q-схемы различают статические и динамические. Статические приоритеты назначаются заранее и не зависят от состояний Q-схемы, т.е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании. Исходя из правил выбора заявок из накопитель N_i на обслуживание каналом k_i можно выделить относительные и абсолютные приоритеты. Относительный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель N , ожидает окончания обслуживания представляющей заявки каналом k_i и только после этого занимает канал. Абсолютный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель, прерывает обслуживание каналом k_i заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из k_i заявка может либо покинуть систему, либо может быть снова записана на какое-то место в N_i).

Необходимо также знать набор правил, по которым заявки покидают N_i и k_i : для N_i – либо правила переполнения, либо правила ухода, связанные с

истечением времени ожидания заявки в H_i ; для k_i – правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале k_i , т.е. правила блокировок канала. При этом различают блокировки k_i по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q-схеме, регулирующих поток заявок в зависимости от состояний Q-схемы. Набор возможных алгоритмов поведения заявок в Q-схеме можно представить в виде некоторого оператора алгоритмов поведения заявок **A**. Т.о. Q-схема, описывающая процесс функционирования СМО любой сложности однозначно задаётся в виде набора множеств: $Q = \langle W, U, H, Z, R, A \rangle$.

Имитационное моделирование систем.

Процедура имитационного моделирования.

Определение метода имитационного моделирования. Метод ИМ заключается в создании логико-аналитической (математической модели системы и внешних воздействий), имитации функционирования системы, т.е. в определении временных изменений состояния системы под влиянием внешних воздействий и в получении выборок значений выходных параметров, по которым определяются их основные вероятностные характеристики. Данное определение справедливо для стохастических систем.

При исследовании детерминированных систем отпадает необходимость изучения выборок значений выходных параметров.

Модель системы со структурным принципом управления представляет собой совокупность моделей элементов и их функциональные взаимосвязи. Модель элемента (агрегата, обслуживающего прибора) - это, в первую очередь, набор правил (алгоритмов) поведения устройства по отношению к выходным воздействиям (заявкам) и правил изменений состояний элемента. Элемент отображает функциональное устройство на том или ином уровне детализации. В простейшем случае устройство может находиться в работоспособном состоянии или в состоянии отказа. В работоспособном состоянии устройство может быть занято, например, выполнение операции по обслуживанию заявки или быть свободным. К правилам поведения устройства относятся правила выборки заявок из очереди; реакция устройства на поступление заявки, когда устройство занято или к нему имеется очередь заявок; реакция устройства на возникновение отказа в процессе обслуживания заявки и некоторые другие.

Имитационное моделирование (ИМ) — это метод исследования, который основан на том, что анализируемая динамическая система заменяется имитатором и с ним производятся эксперименты для получения об изучаемой системе. Роль имитатора зачастую выполняет программа ЭВМ. Основная идея метода ИМ состоит в следующем. Пусть необходимо определить функцию распределения случайной величины y . Допустим, что искомая величина y может быть представлена в виде зависимости: $y=f(\alpha, \beta, \dots$

, ω) где $\alpha, \beta, \dots, \omega$ случайные величины с известными функциями распределения.

Для решения задач такого вида применяется следующий алгоритм:

- 1) по каждой из величин $\alpha, \beta, \dots, \omega$ производится случайное испытание, в результате каждого определяется некоторое конкретное значение случайной величины $\alpha_i, \beta_i, \dots, \omega_i$;
- 2) используя найденные величины, определяется одно частное значение y_i по выше приведённой зависимости;
- 3) предыдущие операции повторяются N раз, в результате чего определяется N значений случайной величины y ;
- 4) на основании N значений величины находится её эмпирическая функция распределения.

Имитация функционирования системы.

Предположим, исследуется вычислительная система (ВС), состоящая из процессора 1 с основной памятью, устройство ввода перфокарт 4, АЦПУ 2 и дисплея 3 (рис. 4.1.).

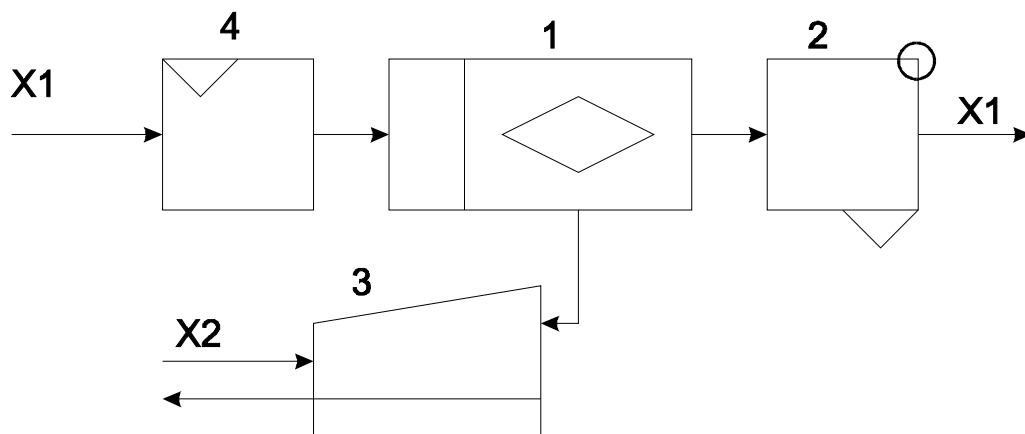


Рис. 4.1. Упрощённая схема моделируемой системы.

Через устройство 4 поступает поток заданий $X1$. Процессор обрабатывает задания и результаты выдаёт на АЦПУ 2. Одновременно с этим ВС используется, например, как информационно-справочная система. Оператор-пользователь, работающий за дисплеем, посылает в систему запросы $X2$, которые обрабатываются процессором и ответы выводятся на экран дисплея. Процессор работает в 2-х программном режиме: в одном разделе обрабатываются задания $X1$, в другом, с более высоким относительным приоритетом запросы $X2$. Представим данную ВС в упрощённом варианте в виде стохастической сети из 4-х СМО. Поток заданий и запросы будем называть потоками заявок. Считаем потоки $X1$ и $X2$ независимыми.

Известны ф.р. периодов следования заявок τ_1 и τ_2 и длительность обслуживания T_{1K} , T_{2K} заявок в k -ом устройстве. Требуется определить времена загрузки каждого устройства и времена реакции по каждому из потоков.

Вначале определяется момент поступления в систему 1-ой заявки потока X1 по результатам случайного испытания в соответствии с ф.р. периода следования заявок.

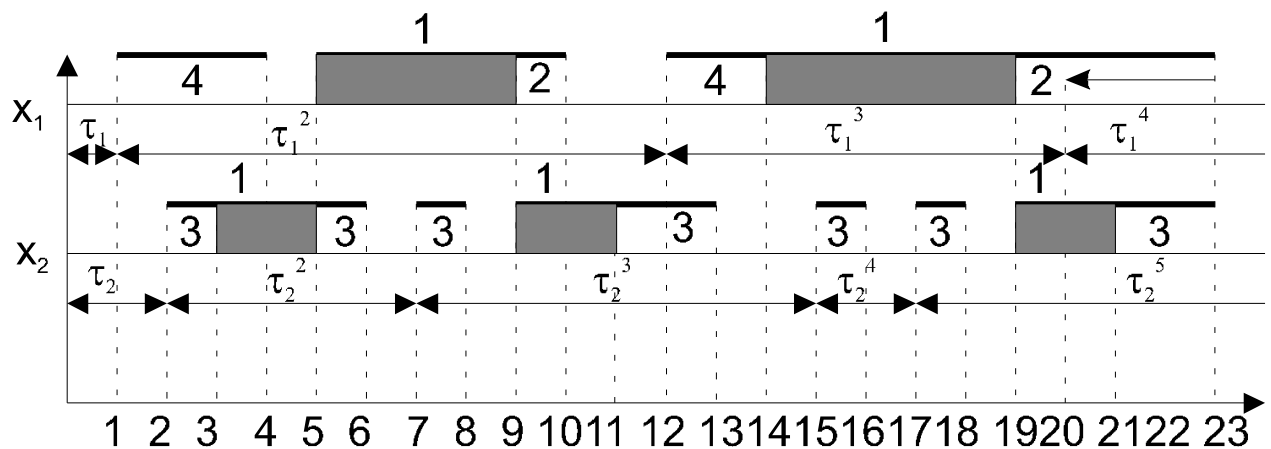


Рис. 4.2. Временная диаграмма функционирования ВС.

На рис. 2 это момент времени $t_1=0+\tau_1^1$ (здесь и далее верхний индекс обозначает порядковый номер заявки данного потока). То же самое делается для потока X2. На рис.2 момент поступления 1-ой заявки потока X2 $t_2=0+\tau_2^1$. Затем находится минимальное время, т.е. наиболее раннее событие. В примере это время t_1 . Для 1-ой заявки потока X1 определяется время обслуживания устройством ввода перфокарт T_{14}^1 методом случайного испытания и отмечается момент окончания обслуживания $t_4=t_1+T_{14}^1$. На рис. показан переход устройства 4 в состояние "занято". Одновременно определяется момент поступления следующей заявки потока X1: $t_{12}=t_1+\tau_1^2$. Следующее минимальное время это момент поступления заявки потока X2 - t_2 . Для этой заявки находится время обслуживания на дисплее T_{23}^1 и отслеживается время окончания обслуживания $t_3=t_2+T_{23}^1$. Определяется момент поступления второй заявки потока X2: $t_7=t_2+\tau_2^2$. Снова выбирается минимальное время – это t_3 . В этот момент заявка потока X2 начинает обрабатываться процессором. По результату случайного испытания определяется время её обслуживания T_{21}^1 и отмечается момент $t_5=t_3+T_{21}^1$ окончания обслуживания. Следующее минимальное время t_4 - момент завершения обслуживания заявки потока X1 устройством 4. С этого момента заявка может начать обрабатываться процессором, но он занят обслуживанием потока X2. Тогда заявка потока X1 переходит в состояние ожидания, становится в очередь. В следующий момент времени t_5 освобождается процессор. С этого момента процессор начинает обрабатывать заявку потока X1, а заявка потока X2 переходит на обслуживание дисплеем, т.е. ответ на запрос пользователя передаётся из основной памяти в буферный накопитель дисплея. Далее определяются соответствующие времена обслуживания: T_{11}^1 и T_{23}^1 и отмечаются моменты времени $t_9=t_5+T_{11}^1$ и $t_6=t_5+T_{23}^1$. В момент t_6 полностью завершается обработка первой заявки потока X2. По разности времени t_6 и t_2 вычисляется время реакции по этой заявке $u_2^1=t_6-t_2$. Следующий минимальный момент t_7 - это

наступление 2-ой заявки потока X2. Определяет время поступления очередной заявки этого потока $t_{15} = t_7 + \tau_2^3$. Затем вычисляется время обслуживания 2-ой заявки на дисплее T_{23}^2 и отмечается момент $t_8 = t_7 + T_{23}^2$, после чего заявка становится в очередь, т.к. процессор занят. Эта заявка поступит на обслуживание в процессор только после его освобождения в момент t_9 . В этот момент заявка потока X1 начинает обслуживаться в АЦПУ. Определяются времена обслуживания T_{21}^2 и T_{12}^1 по результатам случайных испытаний и отмечаются моменты окончания обслуживания $t_{11} = t_9 + T_{23}^2$ и $t_{10} = t_9 + T_{12}^1$. В момент времени t_{10} завершается полное обслуживание 1-ой заявки потока X1. Разность между этим моментом и моментом времени t_1 даёт 1-ое значение времени реакции по потоку X1 $u_1^1 = t_{10} - t_1$.

Указанные процедуры выполняются до истечения времени моделирования. В результате получается некоторое количество (выборка) случайных значений времени реакции (u_1) и (u_2) по 1-ому и 2-ому потокам. По этим значениям могут быть определены эмпирические функции распределения и вычислены количественные вероятностные характеристики времени реакции. В процессе моделирования можно суммировать продолжительности занятости каждого устройства обслуживанием всех потоков. Например, на рис. 2 занятость процессора 1 выделена заштрихованными ступеньками. Если результаты суммирования разделить на время моделирования, то получатся коэффициенты загрузки устройств.

Можно определить время ожидания заявок в очереди, обслуженных системой, среднюю и максимальную длину очереди заявок к каждому устройству, требуемая ёмкость памяти и др.

Имитация даёт возможность учесть надёжностные характеристики ВС. В частности, если известны времена наработки на отказ и восстановления всех входящих в систему устройств, то определяются моменты возникновения отказов устройств в период моделирования и моменты восстановления. Если устройство отказало, то возможны решения:

- снятие заявки без возврата;
- помещение заявки в очередь и дообслуживание после восстановления;
- поступление на повторное обслуживание из очереди;

Обобщённые алгоритмы имитационного моделирования.

Алгоритм моделирования по принципу особых состояний.

Оно использовалось в приведённом выше примере. В качестве событий выделены:

- поступление заявки в систему;
- освобождение элемента после обслуживания заявки;
- завершения моделирования;

* возникновение отказа	}	устройств	другие	типы
* завершение		восстановления устройств	событий	

Процесс имитации развивался с использованием управляющих последовательностей, определяемых по функциям распределения вероятностей исходных данных путём проведения случайных испытаний. В качестве управляющих последовательностей использовались в примере последовательности значений периодов следования заявок по каждому i -ому потоку $\{\tau_i\}$ и длительности обслуживания заявок i -ого потока устройством $\{T_{ik}\}$. Моменты наступления будущих событий определялись по простым рекуррентным соотношениям. Эта особенность даёт возможность построить простой циклический алгоритм моделирования, который сводится к следующим действиям:

- 1) определяется событие с минимальным временем — наиболее раннее событие;
- 2) модельному времени присваивается значение времени наступления наиболее раннего события;
- 3) определяется тип события;
- 4) в зависимости от типа события предпринимаются действия, направленные на загрузку устройств и продвижение заявок в соответствии с алгоритмом их обработки, и вычисляются моменты наступления будущих событий; эти действия называют реакцией модели на события;
- 5) перечисленные действия повторяются до истечения времени моделирования.

В процессе моделирования производится измерение и статистическая обработка значений выходных характеристик. Обобщённая схема алгоритма моделирования по принципу особых состояний приведена на рисунке 5.1.

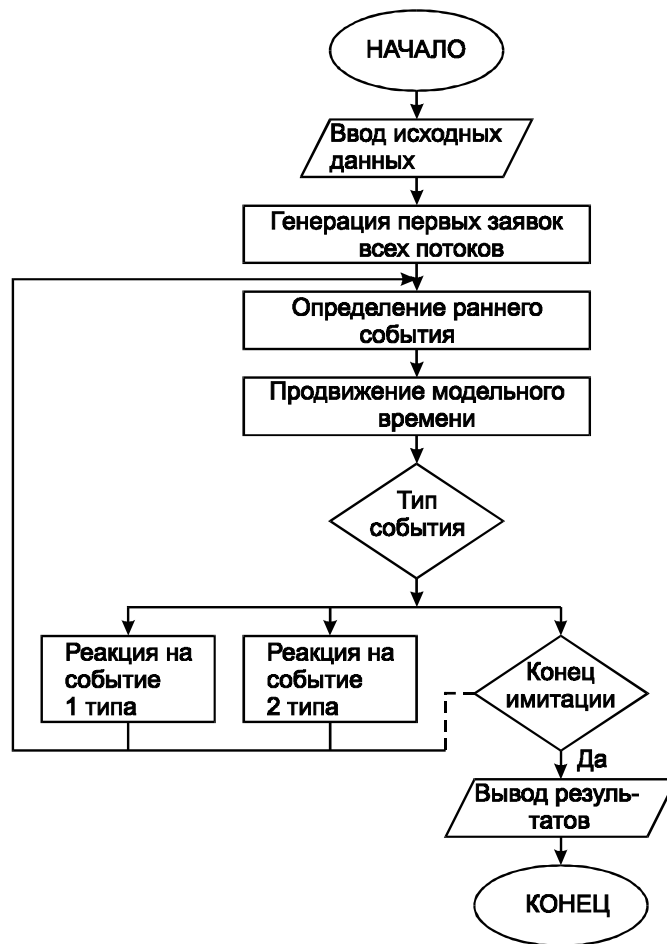


Рис. 5.1. Обобщённый алгоритм моделирования систем по принципу особых состояний

Алгоритм моделирования по принципу Δt .

Укрупнённая схема моделирующего алгоритма, который реализует принцип постоянного приращения модельного времени (принципа Δt), представлен на следующем рисунке:

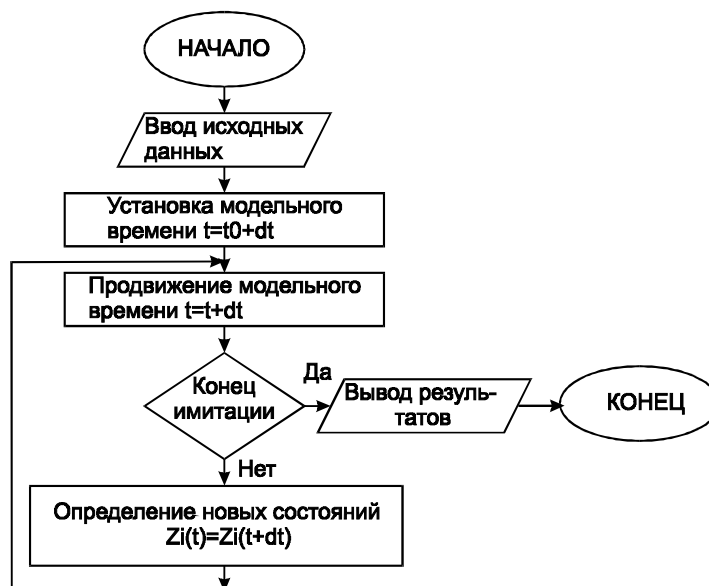


Рис. 5.2. Обобщённый алгоритм моделирования систем по принципу приращений " Δt "

В начале инициализируется программа, в частности вводятся значения $Z_i(t_0)$, $i=1,2,\dots,k$. Которые характеризуют состояние системы в k -мерном фазовом пространстве состояний в начальный момент времени t_0 . Модельное время устанавливается $t = t_0 = 0$. Основные операции по имитации системы осуществляется в цикле. Функционирование системы отслеживается по последовательной схеме состояний $Z_i(t)$. Для этого модельному даётся некоторое приращение dt . Затем по вектору текущих состояний определяются новые состояния $Z_i(t + dt)$, которые становятся текущими. Для определения новых состояний по текущим в формализованном описании системы должны существовать необходимые математические зависимости. По ходу имитации измеряются, вычисляются, фиксируются необходимые выходные характеристики. При моделировании стохастических систем вместо новых состояний вычисляются распределения вероятностей для возможных состояний. Конкретные значения вектора текущих состояний определяются по результатам случайных испытаний. В результате проведения имитационного эксперимента получается одна из возможных реализаций случайного многомерного процесса в заданном интервале времени (t_0, T_k) .

Моделирующий алгоритм, основанный на применении dt применим для более широкого круга систем, чем алгоритм, построенный по принципу особых состояний. Однако при его реализации возникают проблемы определения величины dt . Для моделирования ВС на системном уровне в основном используются принцип особых состояний.

Методы определения характеристик моделируемых систем.

Измеряемые характеристики моделируемых систем.

При имитационном моделировании можно измерять значения любых характеристик, интересующих исследователя. Обычно по результатам вычислений определяются характеристики всей системы, каждого потока и устройства.

Для всей системы производится подсчёт поступивших в систему заявок, полностью обслуженных и покинувших систему заявок без обслуживания по тем или иным причинам. Соотношения этих величин характеризует производительность системы при определённой рабочей нагрузке.

По каждому потоку заявок могут вычисляться времена реакций и ожидания, количества обслуженных и потерянных заявок. По каждому устройству определяется время загрузки при обслуживании одной заявки и число обслуженным устройством заявок, время простоя устройства в результате отказов и количество отказов, возникших в процессе моделирования, длины очередей и занимаемые ёмкости памяти.

При статистическом моделировании большая часть характеристик — это случайные величины. По каждой такой характеристике y определяется N значений, по которым строится гистограмма относительных частот, вычисляется математическое ожидание, дисперсия и моменты более высокого порядка, определяются средние по времени и максимальные значения. Коэффициенты загрузки устройств вычисляются по формуле:

$$\rho_k = V_k * N_{ok} / T_m \quad (1)$$

V_k - среднее время обслуживания одной заявки k -ым устройством;

N_{ok} - количество обслуженных заявок устройством за время моделирования T_m .

Определение условий удовлетворения стохастических ограничений при имитационном моделировании производится путём простого подсчёта количества измерений, вышедших и не вышедших за допустимые пределы.

Расчёт математического ожидания и дисперсии выходной характеристики.

В случае стационарного эргодического процесса функционирования системы вычисление $M(y)$ и $D(y)$ выходной характеристики y производится усреднением не по времени, а по множеству $N_{\text{знач.}}$ измеренных по одной реализации достаточной длительности. В целях экономия ОЗУ ЭВМ $M(y)$ и $D(y)$ вычисляются по рекуррентным формулам:

$$m_n = m_{n-1} * (n-1) / n + y / n; \quad (2)$$

где m_{n-1} - математическое ожидание, вычисленное на предыдущем шаге.

$$d_n = d_{n-1} * (n-2) / (n-1) + 1/n * (y_n - m_{n-1})^2 \quad (3)$$

здесь d_{n-1} - дисперсия, вычисленная на предыдущем шаге.

При большом количестве измерений эти оценки являются состоятельными и несмещёнными.

Расчёт среднего по времени значения выходной характеристики.

Например, средняя длина очереди к каждому устройству вычисляется по формуле:

$$l_N = \sum_{i=1}^N l_i \tau_i / T_m \quad (4)$$

где i - номер очередного изменения состояния очереди (занесение заявки в очередь или исключение из очереди); N - количество изменений состояния очереди; τ_i - интервал времени между двумя последними изменениями очереди.

$$\text{Ёмкость накопитель:} \quad q_n = \sum_{i=1}^N q_i \tau_i / T_m \quad (5)$$

где q_i - ёмкость накопителя, занятая в интервале между двумя последними обращениями к накопителю для ввода-вывода заявки.

Построение гистограммы для стационарной системы.

Γ – эмпирическая плотность распределения вероятностей. Задаются границы изменения интересующей характеристики. $y_i \rightarrow [y_n; y_b]$, числом интервалов N_g . Определяется ширина интервала $\Delta = (y_n - y_b)/N_g$. Затем в процессе моделирования по мере появления значений y_i определяется число попаданий этой случайной величины в каждый из интервалов R_i гистограммы. По этим данным вычисляется относительная частота по каждому интервалу: $G_i = R_i/(N \cdot \Delta)$, где N – общее число измерений y . Площадь гистограммы равна единице, равна сумме площадей:

$$S_i = \sum_1^{N_g} G_i \cdot \Delta = \sum \frac{R_i}{N \cdot \Delta} \cdot \Delta = \sum \frac{R_i}{N} = 1, \quad \text{т.к. } N = \sum_1^{N_g} R_i$$

При необходимости выдвигается гипотеза о том, что эмпирическое распределение согласуется с некоторым теоретическим распределением. Эта гипотеза проверяется по тому или иному критерию. Например, при использовании критерия χ^2 в качестве меры

расхождения используется выражение

$$\chi^2 = \frac{\sum_1^{N_g} (R_i - N \cdot P_i)^2}{N \cdot P_i}$$

(6);

где - P_i определяется из выбранного теоретического распределения вероятность попадания случайной величины в i -ый интервал.

$$P_i = \int_{y_i}^{y_{i+1}} \varphi(x) dx = F(y_{i+1}) - F(y_i) \quad (7).$$

Из теоремы Пирсона следует, что для любой функции распределения $F(y)$ случайной величины y при $N \rightarrow \infty$ распределения величины χ^2 имеет вид:

$$M_k(z) = p(\chi^2 < z) = \frac{1}{2^{k/2} \cdot \Gamma(k/2)} \int_0^z t^{k/2-1} \cdot e^{-t/2} dt, \quad \text{где } z - \text{значение случайной}$$

величины χ^2 ,

$k = N_g - (r + 1)$ - число степеней свободы распределения χ^2 . r - количество параметров теоретического распределения, $\Gamma(k/2)$ - гамма функция.

Функция распределения χ^2 табулирована. По вычисленному значению χ^2 и числу степеней свободы с помощью таблиц определяется вероятность $P(\chi^2 < Z)$. Если она превышает заданный уровень значимости C , то выдвинутая гипотеза принимается.

Моделирование случайных воздействий.

Важной задачей в практике имитационного моделирования систем на ЭВМ является расчёт случайных величин. В языках программирования существуют датчики равномерно распределённых псевдослучайных величин в интервале $\{0,1\}$. Остановимся на вопросах преобразования последовательности псевдослучайных величин $\{X_i\}$ в последовательности

$\{Y_i\}$ с заданным законом распределения и моделировании различных случайных событий.

Рассмотрим особенности моделирования случайных событий.

Пусть имеются случайные числа x_i , т.е. возможные значения случайной величины ξ , равномерно распределённой в интервале $\{0,1\}$. Необходимо реализовать случайное событие A , наступающее с заданной вероятностью P . Определим A как событие, состоящее в том, что выбранное значение x_i удовлетворяет неравенству:

$$x_i \leq P \quad (1)$$

Тогда вероятность события A будет : $P(A) = \int_0^P dx = P$. Противоположное

событию A состоит в том, что $x_i > P$. Тогда $P(\bar{A}) = 1 - P$. Процедура моделирования состоит в этом случае в выборе значений x_i и сравнение их с P . При этом, если условие (1) удовлетворяется, то исходом испытания будет событие A .

Таким же образом можно рассмотреть группу событий. Пусть $A_1, A_2, \dots, A_n$ – полная группа событий, наступающая с вероятностями $P_1, P_2, \dots, P_n$ соответственно. Определим A_m как событие, состоящее, в том, что выбранное значение x_i случайной величины ξ удовлетворяет неравенству:

$$l_{m-1} < x_i \leq l_m, \text{ где } l_k = \sum_{i=1}^k P_i. \quad (2)$$

Тогда $P(A_m) = \int_{l_{m-1}}^{l_m} dx = P_m$. Процедура моделирования испытаний в этом случае

состоит в последовательности сравнений случайных чисел x_i со значениями l_k . Исходом испытания оказывается событие A_m , если выполняется условие (2). Эту процедуру называют определением исхода по жребию в соответствии с вероятностями $P_1, P_2, \dots, P_n$.

При моделировании систем часто необходимо осуществить такие испытания, при которых искомый результат является сложным событием, зависящим от 2-х и более простых.

Пусть например, независимые события A и B имеют вероятности наступления P_A и P_B . Возможными исходами совместных испытаний в этом случае будут события $AA, \bar{A}B, A\bar{B}, \bar{A}\bar{B}$ с вероятностями $P_AP_B, (1-P_A)P_B, P_A(1-P_B), (1-P_A)(1-P_B)$. Для моделирования совместных испытаний можно использовать последовательную проверку условия (1). Он требует двух чисел x_i .

Рассмотрим случай, когда события A и B являются зависимыми и наступают с вероятностями P_A и P_B . Обозначим через $P(B/A)$ условную вероятность события B при условии, что событие A произошло. Считаем, что $P(B/A)$ задана. Из последовательности случайных чисел $\{X_i\}$ извлекается определённое число x_m и проверяется справедливость неравенства $x_m < P_A$. Если это неравенство справедливо, то наступило событие A . Для испытания, связанного с событием B используется вероятность $P(B/A)$. Из совокупности

чисел $\{X_i\}$ берётся очередное число x_{m+1} и проверяется условие $x_{m+1} \leq P(B/A)$. В зависимости от того выполняется или нет это неравенство, исходом испытания является AB или $\overline{A}\overline{B}$. Если неравенство $x_m < P_A$ не выполняется, то наступило событие $\overline{A}$. Поэтому для испытания, связанного с событием B необходимо определить вероятность:

$$P(B / \overline{A}) = [P(B) - P(A) / P(B / A)] / (1 - P(A))$$

Выберем из совокупности $\{X_i\}$ число x_{m+1} и проверим справедливость неравенства $x_m \leq P(B / \overline{A})$. В зависимости от того, выполняется оно или нет, получаем исходы испытания $\overline{A}B$. Алгоритм вычислений можно представить в виде схемы, которая изображена на рисунке 7.1.

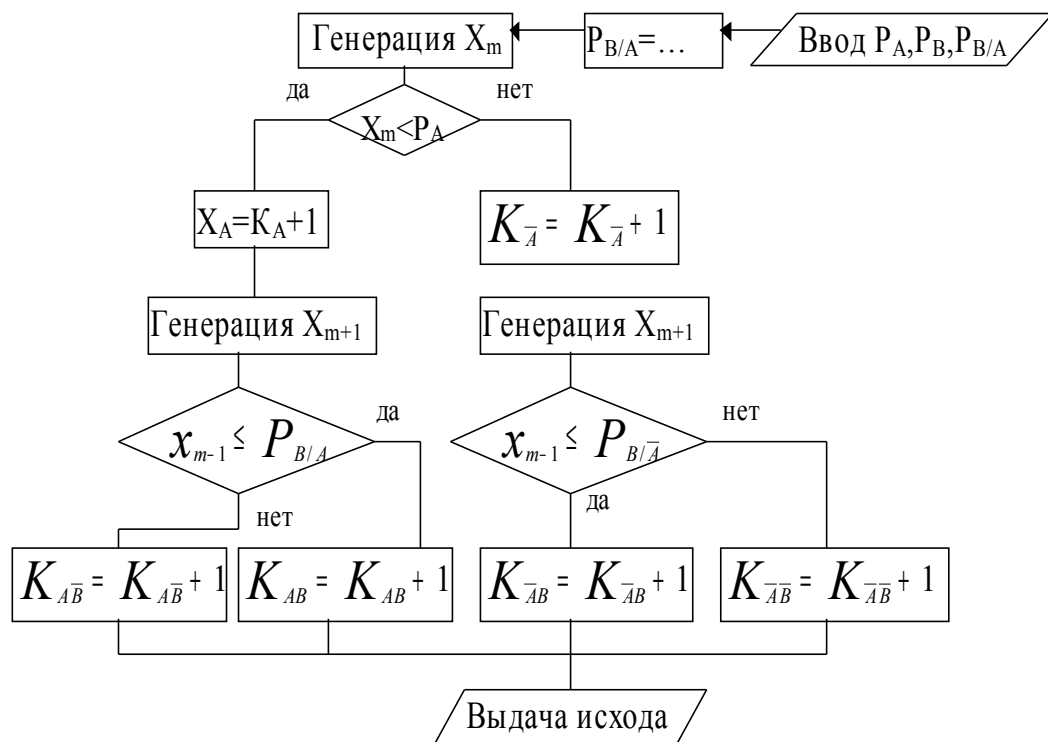


Рис.7.1. Схема моделирования группы случайных событий

Преобразование случайных величин.

Дискретная случайная величина η принимает значения $y_1 \leq y_2 \leq y_3 \dots y_l$ с вероятностями $P_1, P_2, \dots, P_l$ составляющими дифференциальное распределение вероятностей:

$$\begin{array}{ccc} y & y_1 & y_2 & \dots & y_j & \dots \\ P(\eta=y) & P_1 & P_2 & \dots & P_j & \dots \end{array} \quad (3)$$

При этом интегральная функция распределения $F_\eta(y) = P(\eta \leq y) = \sum_{j=1}^m P_j$; $y_m \leq y_{m+1}$; $m=1, 2, \dots$

$$F_\eta(y)=0, y < y_1. \quad (4)$$

Для получения дискретных случайных величин можно использовать метод обратной функции. Если ξ - равномерно распределённая на интервале $(0, 1)$, случайная величина η получается с помощью преобразования

$$\eta = F_{\eta}^{-1}(\xi), \text{ где } F_{\eta}^{-1} - \text{функция, обратная } F_{\eta}. \quad (5)$$

Алгоритм вычисления по (4) и (5) сводится к выполнению следующих действий:

$$\begin{aligned} &\text{если } x_1 < P_1 \text{ то } \eta = y_1 \text{ иначе,} \\ &\text{если } x_2 < P_1 + P_2 \text{ то } \eta = y_2 \text{ иначе,} \\ &\dots\dots\dots \\ &\text{если } x_j < \sum_{j=1}^m P_j \text{ то } \eta = y_m \text{ иначе} \\ &\dots\dots\dots \end{aligned} \quad (6)$$

При счёте по (6) среднее число циклов сравнения равняется $\sum_{j=1}^{\infty} j P_j$

Пример 1. Необходимо методом обратной функции на основании базовой последовательности случайных чисел $\{x_i\}$, равномерно распределённых в интервале $(0,1)$, получить последовательность чисел $\{y_i\}$, имеющих биномиальное распределение, задающее вероятность y удачных исходов в N реализациях некоторого эксперимента:

$$P(\tau_i=y) = P_N(y) = C_N^y P^y (1-P)^{N-y}, \text{ где } P=0.5 \text{ и } N=6; C_N^y = N!/y!(N-y)!$$

Математическое ожидание и дисперсия биномиального распределения соответственно будут $M[y] = np(1-P)$. Используя для P_j обозначения, принятые в (6), вычислим:

j	...	1	2	3	4	5	6	7
y_j	...	0	1	2	3	4	5	6
P_j	...	0.01562	0.09375	0.23438	0.3125	0.23438	0.09375	0.01562
$\sum_{j=1}^m P_j$	...	0.01562	0.10937	0.34375	0.65625	0.89063	0.98438	1.0000

Например, получив из равномерного распределения число $X_i = 0.89063$ и проведя сравнения по алгоритму (6), найдём, что $0.85393 < 0.89063$, т.е. $y_i = 4$.

При этом среднее число циклов сравнения $\bar{\mu} = 1*0.01562 + 2*0.09375 + 3*0.23438 + 4*0.31250 + 5*0.23438 + 6*(0.09375 + 0.01562) \approx 3.98$.

Вычисление непрерывных случайных величин.

Непрерывная случайная величина η задана интегральной функцией распределения:

$$F_{\eta}(y) = P(\eta \leq y) = \int_{-\infty}^y f_{\eta}(y) dy, \text{ где } f_{\eta}(y) - \text{плотность}$$

вероятностей.

Для получения непрерывных случайных величин с заданным законом распределения, как и для дискретных величин можно использовать метод обратной функции. Взаимно однозначная монотонная функция $\eta = F_{\eta}^{-1}(\xi)$, полученная решением относительно η уравнения $F_{\eta}(y) = \xi$ преобразует равномерно распределённую на интервале $(0,1)$ величину ξ в η с требуемой плотностью $f_{\eta}(y)$.

Действительно, если случайная величина η имеет плотность распределения $f_\eta(y)$, то распределение случайной величины $\xi = \int_0^\eta f_\eta(y) dy$ является равномерным.

Т.о. чтобы получить число, принадлежащее последовательность случайных чисел $\{y\}$, имеющих функцию плотности $f_\eta(y)$, необходимо разделить относительно y_i управление

$$\int_{-\infty}^{y_i} f_\eta(y) dy = x_j \quad (7)$$

Пример 2. Необходимо получить случайные числа y_i с показательным законом распределения.

$$f_\eta(y) = \lambda e^{-\lambda y}, y > 0.$$

В силу соотношения (7) получим $\lambda \int_0^{y_i} e^{-\lambda y} dy = x_i$, где x_i - случайное число, имеющее равномерное распределение в интервале (0,1), тогда

$$1 - e^{-\lambda y_i} = x_j, y_j = -\ln(1 - x_j) / \lambda.$$

Рассмотрим универсальный метод моделирования непрерывных случайных величин (**метод исключений**).

При моделировании случайной величины y с плотностью распределения вероятностей $f_\eta(y)$ в интервале $a \leq y \leq b$ независимые значения x_m и x_{m+1} преобразуются в значения

$$y_{1m} = a + (b-a) * x_m \quad (8)$$

$$z_{1m+1} = f_{1\eta}(y) * x_{m+1} \quad (9)$$

где $f_{1\eta}(y) = \max |f_\eta(y)|$. При этом y_{1m} и z_{1m+1} - значения случайных величин, равномерно распределенных на интервале (a, b) и $(0, f_{1\eta})$. Эти значения можно рассматривать как абсциссы и ординаты случайных точек, равномерно распределяющихся внутри прямоугольника со сторонами $b-a$ и $f_{1\eta}$, охватывающего кривую распределения $f_\eta(y)$ (см. рисунок 7.2.).

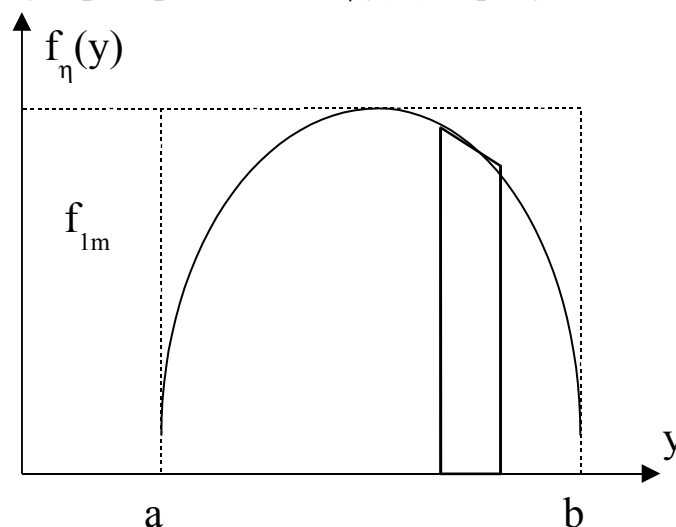


Рис. 7.2. Иллюстрация метода исключений

$$\text{Если } z_{1m+1} \leq f_{\eta}(y_{1m}), \quad (10)$$

тогда пара y_{1m}, z_{1m+1} определяет случайную точку под кривой f_{η} . Вероятность попадания случайной точки, удовлетворяющей условию (10) под кривую f_{η} равна единице, а вероятность попадания в заштрихованную элементарную площадку равна $f_{\eta}(y_{1m}) \cdot \Delta y_{1m}$. Это обозначает, что абсциссы y_{1m} случайных точек, попадающих под кривую f_{η} - значения случайной величины y с заданной плотностью вероятности $f_{\eta}(y)$. Моделируемый алгоритм состоит из функций: 1) получения x_{m1} и x_{m+1} от датчика; 2) расчёта y_{1v} и z_{1m+1} согласно (8) и (9); 3) вычисления $f_{\eta}(y_{1m})$; 4) сравнения z_{1m+1} с $f_{\eta}(y_{1m})$. Если условия (10) выполняются, то $y=y_{1m}$; если нет, то значения y_{1m} и z_{1m+1} исключаются и процесс повторяется, начиная с пункта 1. При моделировании системы 2-х случайных величин (y_1, y_2) с плотностью вероятности $f(y_1, y_2)$, $a_1 \leq y_1 \leq b_1$; $a_2 \leq y_2 \leq b_2$, аналогично моделированию одной случайной величины, три значения: x_m, x_{m+1}, x_{m+2} , выданные датчиком E , преобразуются в значения:

$$y_{1m} = a_1 + (b_1 - a_1)x_m \quad (11)$$

$$y_{2m} = a_2 + (b_2 - a_2)x_{m+1} \quad (12)$$

$$z_{3m} = f_{\eta m} x_{m+2}, \text{ где } f_{\eta m} = \max[f(x_1, x_2)] \quad (13)$$

$$\text{Если } z_{3m} \leq f(y_{1m}, y_{2m}) \quad (14)$$

$$\text{то } y_1 = y_{1m}; y_2 = y_{2m}.$$

В этом случае случайные точки с координатами y_{1m}, y_{2m}, z_{3m} - равномерно распределены в пределах параллелепипеда со сторонами, равными $(b_1 - a_1)(b_2 - a_2), f_{\eta m}$ и условие (14) означает попадание точки под поверхность f_{η} . Аналогично моделируется система n случайных величин $(y_1, y_2, \dots, y_n)$.

Моделирование нормально распределённой случайной величины y .

Оно может быть осуществлено на основании центральной предельной теоремы, согласно которой закон распределения суммы независимых случайных величин стремится к нормальному с увеличением числа

слагаемых. Для решения некоторых задач практически сумму $y = \sum_{i=1}^n y_i$

значений, выданных с генератором случайных чисел с характеристиками $f(x_i)=1, 0 \leq x_i \leq 1, m_x=0.5, \sigma_{x_i} = 0.5/\sqrt{3}$.

Можно считать значениями распределённой случайной величины $y = \sum_{i=1}^n x_i$

при $n \geq 8$. Так как все слагаемые x_i имеют одинаковые математические ожидания m_x и дисперсии D_x , то $m_y = nm_x, D_y = nD_x$. В таблице 1 приведены формулы для расчёта случайных величин для различных видов распределений на базе случайной величины с равномерным распределением.

Получение случайной величины с различными распределениями.

Таблица 1.

Нужное распределение	Плотность распределения	Способ получения
----------------------	-------------------------	------------------

		случайной величины
Экспоненциальное	$f(y) = \lambda e^{-\lambda(y-\mu)}, \mu \leq y < \infty$	$y' = -\frac{1}{\lambda} \ln(1 - x_i) + \mu$
Гамма распределение (целочисленные значения η)	$f(y) = \frac{\lambda^\eta}{\Gamma(\eta)} e^{-\lambda y} y^{\eta-1}, 0 < y < \infty$	$y' = -\frac{1}{\lambda} \sum_{i=1}^{\eta} \ln(1 - x_i)$
Распределение χ^2	$f(y) = \frac{1}{2^{\eta/2} \Gamma(\eta/2)} y^{(\eta/2)-1} e^{-y/2}, 0 \leq y < \infty$	$y' = \sum_{i=1}^{\eta} R_N; R_N -$ нормированная случ. величина с нормальным законом распределения
Логарифмические норм. Распределение	$f(y) = \frac{1}{\sqrt{2\pi} \sigma y} \exp\left(-\frac{1}{2} \left[\frac{\ln y - \mu}{\sigma}\right]^2\right), 0 \leq y < \infty$	$y' = e^{\sigma R_N + \mu}$
Вейбулла	$f(y) = \frac{\eta}{\sigma^\eta} y^{\eta-1} \exp\left[-\left(\frac{y}{\sigma}\right)^\eta\right], 0 \leq y < \infty$	$y' = -\sigma \left[\ln(1 - x_i)\right]^{1/\eta}$

Моделирование систем с использованием типовых математических схем

Блочные иерархические модели процессов функционирования систем

Рассмотрим машинную модель M_m , системы S как совокупность блоков $\{m_i\}$, $i=1, 2, \dots, n$. Каждый блок модели можно охарактеризовать конечным набором возможных состояний $\{Z_0\}$, в которых он может находиться. Пусть в течение рассматриваемого интервала времени $(0, T)$ блок i изменяет состояние в моменты времени $t_i^j \leq T$, где j - номер момента времени. Момент времени можно разделить на три группы:

- случайные, связанные с внутренними свойствами блока;
- случайные, связанные с изменением состоянием других блоков, имитирующая воздействие среды E ;
- детерминированные моменты, связанные с заданным расписанием функционирования блока.

Моментами смены состояний модели M_m в целом $t^{(k)} \leq T$ будем считать все моменты изменения блоков $\{m_i\}$, рис. 8.1. см. ниже.

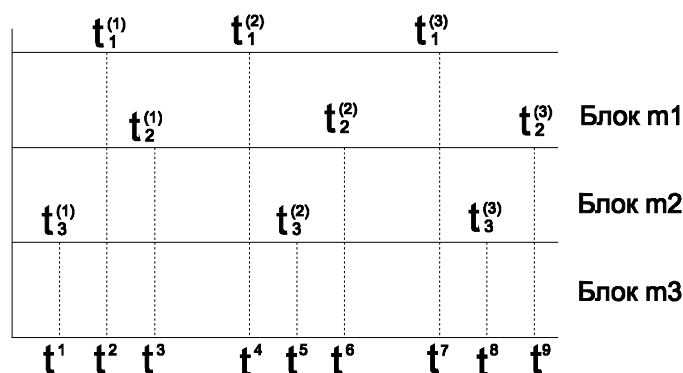


Рис. 8.1. Смена состояний модели для случаев 3-х блоков

При этом моменты $t_i^{(j)}$ и t^k являются моментами системного времени, т.е. времени, в котором функционирует система S . При машинной реализации модели M_m её блки представляются соответствующими программными модулями.

Особенности реализации процессов с использованием Q-схем

При моделировании Q-схем следует адекватно учитывать как связи, отражающие движения заявок (сплошные линии) так и управляющие связи (пунктирные линии).

Рассмотрим фрагмент Q-схемы (Рис. 8.2.):

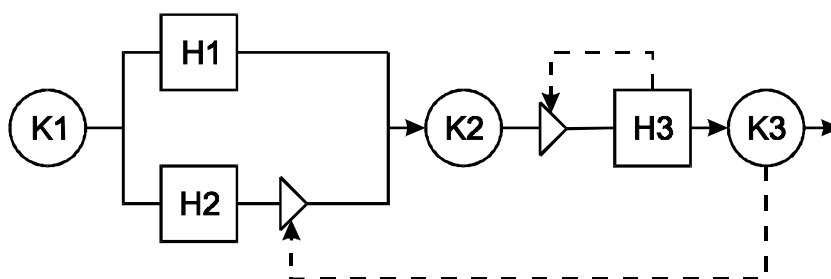


Рис. 8.2. Фрагмент Q-схемы.

Примерами управляющих связей являются различные блокировки обслуживающих каналов (по входу и по выходу): "клапаны" изображены в виде треугольников, а управляющие связи пунктирными линиями.

Блокировка канала по входу означает, что этот канал отключается от входящего потока заявок, а блокировка канала по выходу указывает, что заявка обслуженная заблокированным каналом, остаётся в этом канале до момента снятия блокировки. В этом случае, если перед накопителем нет "клапана", то при его переполнении будут иметь место потери заявок.

Моделирующий алгоритм должен отвечать следующим требованиям:

- ◇ обладать универсальностью относительно структуры, алгоритмов функционирования и параметров системы S ;
- ◇ обеспечивать одновременную и независимую работу системы S ;
- ◇ укладываться в приемлемые затраты ресурсов ЭВМ. (памяти, времени расчёта для реализации машинного эксперимента);

- ◇ проводить разбиение на достаточно автономные логические части (блоки);
- ◇ гарантировать выполнение рекуррентного правила расчётов;

При этом необходимо иметь виду, что появление одной заявки входящего потока в некоторый момент времени t_i может вызвать изменение состояния не более чем одного из элементов Q-схемы, а окончание обслуживания заявки в момент t_i в некотором канале K может привести в этот момент времени к последовательному изменению состояний нескольких элементов (H, K) , т.е. будет иметь место процесс распространения смены состояний в направлении противоположном движению заявки в системе S . Поэтому просмотр элементов Q-схемы должен быть противоположным движению заявок.

Все виды моделирующих алгоритмов Q-схемы можно классифицировать следующим образом (см. Рис. 8.3.):

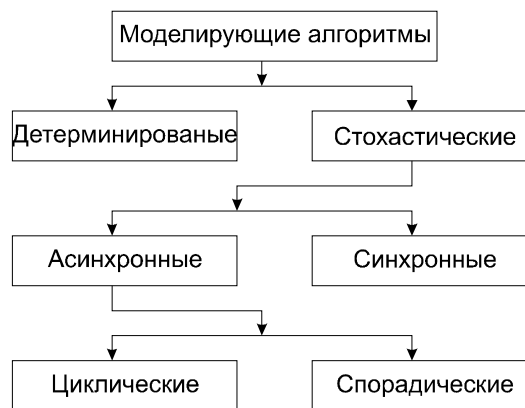


Рис. 8.3. Виды моделирующих алгоритмов Q-схемы.

Алгоритмы моделирующие Q-схему по принципу " Δt " являются детерминированными (по шагу), а по принципу особых состояний – стохастические. Последние могут быть реализованы синхронным и асинхронным способами.

При синхронном способе один из элементов Q-схемы (И, Н или К) выбирается в качестве ведущего и по нему "синхронизируется" весь процесс моделирования.

При асинхронном способе – ведущий (синхронизирующий) элемент не используется, а очередному шагу моделирования (просмотру элементов Q-схемы) может соответствовать любое особое состояние всего множества элементов И, Н и К. При этом просмотр элементов Q-схемы организован так, что при каждом особом состоянии либо циклически просматриваются все элементы, спорадически - только те элементы, которые в этом случае могут изменить своё состояние. (просмотр с прогнозированием)

Построение и реализация моделирующих алгоритмов Q-схем

Прежде чем использовать какой либо язык для моделирования Q-схемы, необходимо глубже вникнуть в суть процесса построения и реализации М.А.

Пример. Рассмотрим Q-схему (Рис. 8.4.):

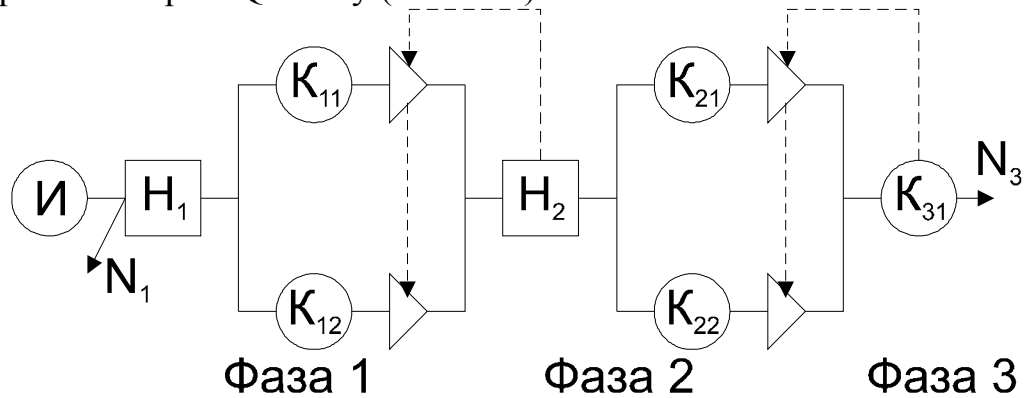


Рис. 8.4. Трёхфазная Q-схема.

Примем обозначения:

P - вероятность потери заявки ($P=N_1/(N_1+N_3)$);

t_m - время появления очередной заявки из источника;

$t_{k,j}$ - время окончания обслуживания заявки каналом $K_{k,j}$, $k=1,2,3\dots$; $j=1,2\dots$;

$z_i, z_{k,j}$ - состояния накопителей и каналов обслуживания;

t_n - текущее время моделирования;

L_i - ёмкость i -ого накопителя;

L_k^m - число каналов в k -ой фазе;

N_1, N_2 - число выходных заявок;

T - интервал моделирования;

При имитации Q-схемы на ЭВМ требуется организовать массив состояний:

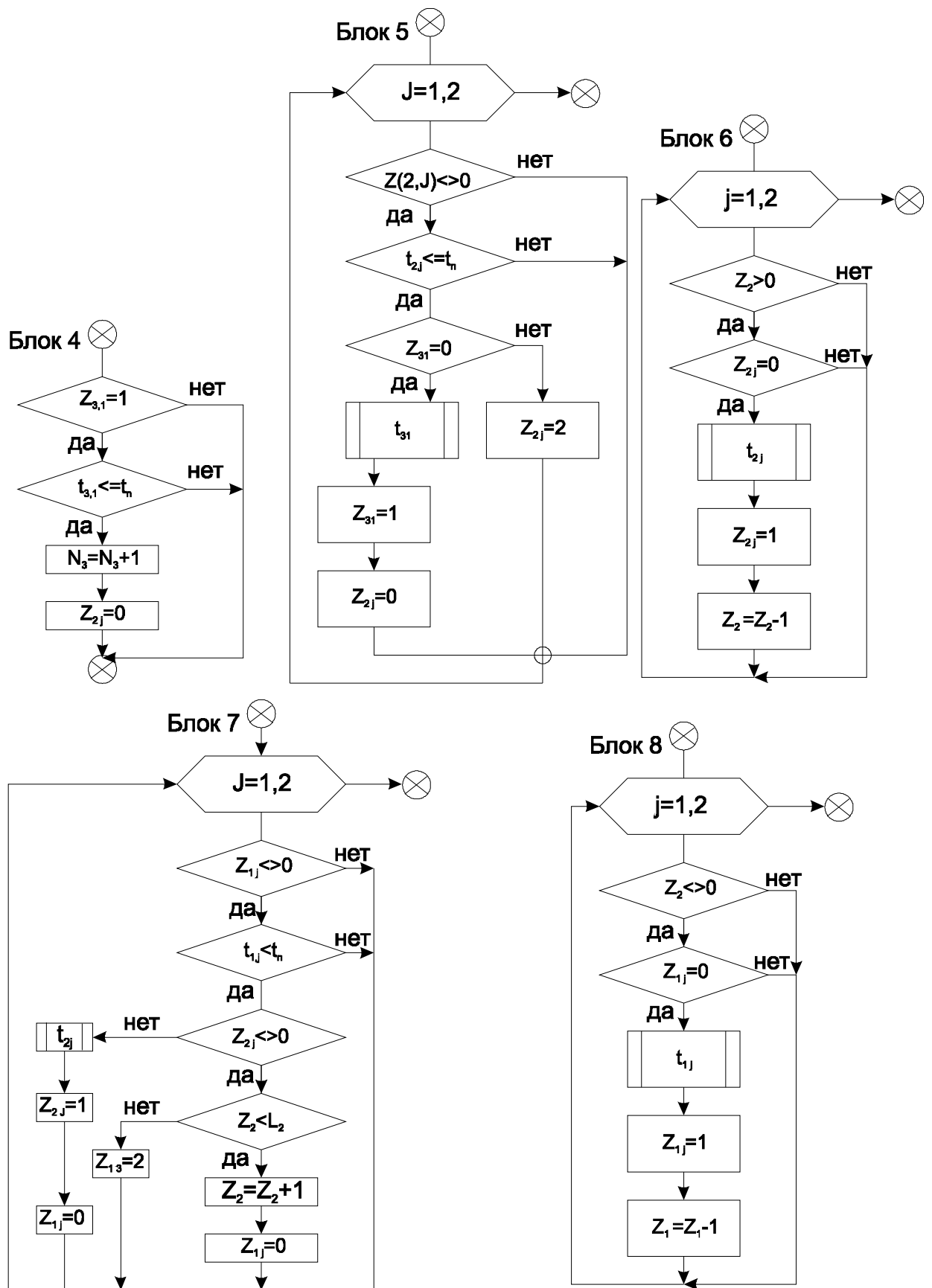
$z_{k,j}, t_{k,j}, j=1, L_k^m$; z_i - число заявок в накопителе H_i ; $i=1,2$; t_i - i -ая заявка из источника.

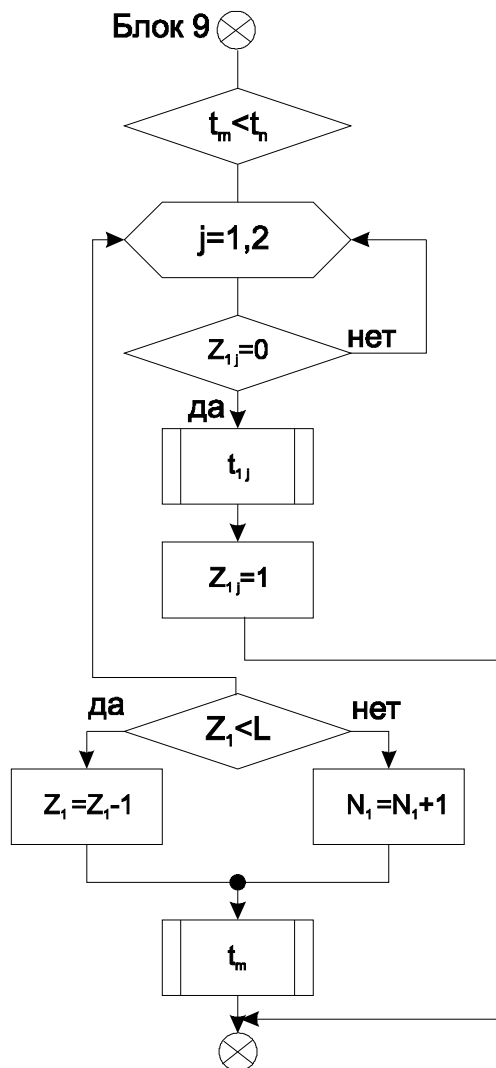
$z_{k,j} = \{1$ - канал занят; 0 - канал свободен; 2 - заблокирован};

Укрупнённая схема детерминированного МА Q-схемы, построенного по "принципу Δt " представлена на рисунке 8.5.



Рис. 8.5. Блок схема моделирования Q-схемы по принципу " Δt ".
А далее более подробно рассмотрены алгоритмы блоков 4-9.





Программные и технические средства моделирования систем.

Моделирование систем и языки программирования.

Большое значение при реализации модели на ЭВМ имеет вопрос правильного выбора языка программирования.

Язык программирования должен отражать внутреннюю структуру понятий при описании широкого круга понятий. Высокий уровень языка моделирования значительно упрощает программирование моделей.

Основными моментами при выборе ЯМ является:

- проблемная ориентация;
- возможности сбора, обработки, вывода результатов;
- быстродействие;
- простота отладки;
- доступность восприятия.

Этими свойствами обладают процедурные языки высокого уровня. Для моделирования могут быть использованы языки Имитационного моделирования (ЯИМ) и общего назначения (ЯОМ).

Более удобными являются ЯИМ. Они обеспечивают:

- удобство программирования модели системы;

— проблемная ориентация.

Недостатки ЯИМ:

- неэффективность рабочих программ;
- сложность отладки;
- недостаток документации.

Основные функции языка программирования:

- управление процессами (согласование системного и машинного времени);
- управление ресурсами (выбор и распределение ограниченных средств описываемой системы).

Как специализированные языки, ЯИМ обладают некоторыми программными свойствами и понятиями, которые не встречаются в ЯОН. К ним относятся:

Совмещение. Параллельно протекающие в реальных системах S процессы представляются с помощью последовательно работающей ЭВМ. ЯИМ позволяют обойти эту трудность путём введения понятий системного времени.

Размер. ЯИМ используют динамическое распределение памяти (компоненты модели системы M появляются в ОЗУ и исчезают в зависимости от текущего состояния). Эффективность моделирования достигается так же использованием блочных конструкций: блоков, подблоков и т.д.

Изменения. ЯИМ предусматривают обработку списков, отражающих изменения состояний процесса функционирования моделируемой системы на системном уровне.

Взаимосвязь. Для отражения большого количества между компонентами модели в статике и динамике ЯИМ включаем системно организованные логические возможности и реализации теории множеств.

Стохастичность. ЯИМ используют специальные программные генерации последовательностей случайных чисел, программы преобразования в соответствующие законы распределения.

Анализ. ЯИМ предусматривают системные способы статистической обработки и анализа результатов моделирования.

Наиболее известными языками моделирования являются SIMULA, SIMSCRIPT, GPSS, SOL, CSL.

Для языков, используемых в задачах моделирования, можно составить классификацию следующего вида. (см. рис. 9.1.)

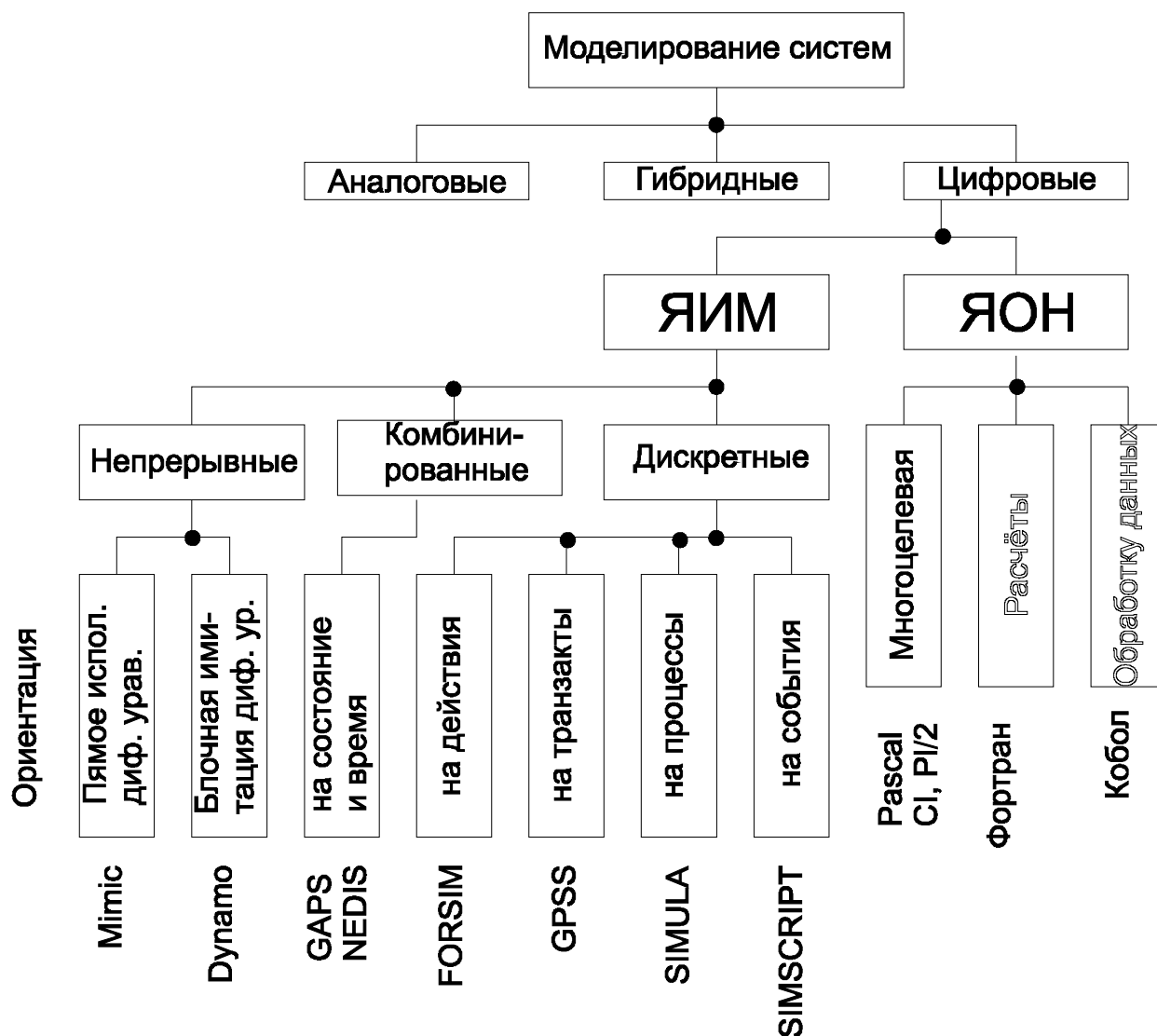


Рис. 9.1. Классификация языков моделирования.

Язык DYNAMO используется для решения разностных уравнений.

Представление системы S в виде типовой схемы, в которой участвуют как дискретные, так и непрерывные величины, называются комбинированными. Предполагается, что в системе могут наступать события двух видов: 1) события, от состояния Z_i ; 2) события, зависящие от времени t . При использовании языка GAPS на пользователя возлагается работа по составлению на яз. FORTRAN подпрограмм, в которых описываются условия наступления событий, законы изменения непрерывной величины, правил перехода из одного состояния в другое. SIMSCRIPT - язык событий, созданный на базе языка FORNRAN. Каждая модель M_j состоит из элементов, с которыми происходят события, представляющие собой последовательность формул, изменяющих состояние моделируемой системы с течением времени. Работа со списками, определяемые пользователем, последовательность событий в системном времени, работа с множествами. FORSIT - пакет ПП на языке FORNRAN позволяет оперировать только фиксированными массивами данных, описывающих объекты моделируемой системы. Удобен для описания систем с большим числом разнообразных

ресурсов. Полное описание динамики модели можно получить с помощью ПП.

SIMULA – расширение языка ALGOL. Блочное представление моделируемой системы. Функционирование процесса разбивается на этапы, происходящие в системном времени. Главная роль в языке SIMULA отводится понятию параллельного оперирования с процессами в системном времени, универсальной обработки списков с процессами в роли компонент.

GPSS- интегрирующая языковая система, применяющаяся для описания пространственного движения объектов. Такие динамические объекты в языке GPSS называются транзактами и представляют собой элементы потока. Транзакты "создаются" и "уничтожаются". Функцию каждого из них можно представить как движение через модель М с поочерёдным воздействием на её блоки. Функциональный аппарат языка образуют блоки, описывающие логику модели, сообщая транзактам, куда двигаться и что делать дальше. Данные для ЭВМ подготавливаются в виде пакета управляющих и определяющих карт, которым составляется по схеме модели, набранной из стандартных символов. Созданная программа GPSS, работая в режиме интерпретации, генерирует и передаёт транзакты из блока в блок. Каждый переход транзакта приписывается к определенному моменту системного времени.

При моделировании предпочтение отдают языку, который более знаком, универсален. Вместе с увеличением числа команд возрастают трудности использования ЯИМ. Получены экспертные оценки ЯИМ по степени их эффективности.

Баллы	Возможности	Простота применения	Предпочтение пользователя
5	SIMULA	GPSS	SIMSCRIPT
4	SIMSCRIPT	SIMSCRIPT	GPSS
3	GPSS	SIMULA	SIMULA

Суммарный балл:

SIMULA -11

SIMSCRIPT -13

GPSS -12

Если предпочтение отдаётся блочной конструкции модели при наличии минимального опыта в моделировании, то следует выбрать язык GPSS, но при этом следует помнить, что он негибок, требует большого объёма памяти и затрат машинного времени для счёта.

Язык программирования GPSS

Этот язык с 1968 года входит в математическое обеспечение машин фирмы IBM, один из наиболее популярных языков ИМ.

Общие сведения.

GPSS составлен из объектов и операций (логических правил). Объекты делятся на семь классов:

- динамические (ДО);
- аппаратно-ориентированные (АО);
- статические (СО);
- операционные (ОО);
- вычислительные (ВО);
- запоминающие (ЗО);
- группирующие (ГО).

ДО – элементы потока обслуживания заявки или "транзакты". Они создаются и уничтожаются, с каждым транзактом может быть связано некоторое число "параметров"

АО – соответствуют элементам оборудования, которые управляются ДО.

К ним относятся:

- накопители;
- устройства;
- логические переключатели.

СО:

- очереди;
- таблицы.

ЗО:

ячейки;
матрицы ячеек.

ГО:

- группы;
- списки.

ВО:

- арифметические и булевы переменные;
- функции.

Каждой очереди соответствует перечень транзактов, задержанных в какой-либо точке системы и запись длительности этих задержек: $T_3 = \{\tau_{i3}\}$.

Таблицы могут использоваться для построения распределений выбранных величин.

ОО - блоки – формируют логику системы, давая транзактам указания, куда идти дальше.

Для того чтобы смоделировать систему, необходимо составить её описание в терминах GPSS, затем симулятор генерирует транзакты, продвигает через заданные блоки и выполняет действия соответствующие блокам.

Продвижение создаёт блок GENERATE. Каждое продвижение транзакта является событием, которое должно произойти в определённый момент времени. Симулятор регистрирует время наступления каждого события, затем производит обработку событий в правильной хронологической последовательности.

Если транзакты заблокированы, то симулятор продвинет их тогда, когда изменятся блокирующие правила. Симулятор моделирует часы, их показания в любой момент времени называют абсолютным временем. Относительное

время показывает текущее время в модели. При помощи специальной операции относительное время может устанавливаться в нуль и последующий счёт времени будет производиться от этой точки. Другой операцией в нуль могут устанавливаться оба значения времени. Все времена в модели изображаются целыми числами. Симулятор рассчитывает схему по принципу ближайшего события. Центральной задачей симулятора является просмотр и проверка всех возможных событий. Транзакты входят в цепи. Существует пять видов цепей:

- 1) Цепь текущих событий включает в себя те транзакты, планируемое время наступления которых равно или меньше текущего часового.
- 2) Цепь будущих событий включает в себя транзакты, время которых не дошло для обслуживания.
- 3) Цепь прерванных событий.
- 4) Цепь парных транзактов – в текущий момент времени имеют статус парности (ожидают прибытия синхронизирующих транзактов).
- 5) Цепь пользователя включает транзакты, которые пользователь удалили из цепи текущих транзактов.

Цепь текущих событий организуется в порядке убывания приоритетов транзактов и в порядке очередности поступления.

В зависимости от различных условий и требований пользователь система помещает транзакты в те или иные цепи.

Программа на GPSS создаётся в текстовом редакторе в определённом формате. Формат ввода содержит 3 различные поля: поле метки (позиции 5-9), поле операции (позиции 13-23) и поле переменных (позиции 26-80). Поле переменных содержит подполя, которые обозначены A, B, C, D, ..., H.

Последующие отделяются от предыдущих запятыми. Пропущенное значение в поле переменных выделяется запятыми (кроме конца поля).

Каждый из объектов требует определённого числа ячеек ОЗУ, в которых во время моделирования хранятся атрибуты объекта (АТО). АТО, к которым может обращаться программист, называются стандартными числовыми атрибутами (СЧА). Все СЧА имеют одно- или 2-х буквенные мнемонические обозначения. Мнемонические обозначения указывают на тип СЧА, а целочисленное значение – на конкретный СЧА.

Номера блоков можно определять символическими обозначениями. При этом обозначение должно включать от 3-х до 5-ти знаков, отличных от пробела, первые три из которых должны быть буквами. Эти ограничения необходимы для того, чтобы избежать смешивания атрибутов системы и символов.

Дополнительным ограничением является недопустимость таких специальных знаков, как "-", "+", " " и т.д.

Если в полях A, B, C блока представлены стандартные числовые атрибуты N_j или W_j , то необходимо, чтобы номер блока был представлен в качестве аргумента. Если этот номер блока определяется символически, то такое представление должно быть отличным от мнемонических обозначений, указанных СЧА (N или W). В префикса символического имени используется знак доллара \$. Пользователь может относительную адресацию. В

символической записи CROSS±n символ CROSS указывает на нужный блок, а число n—на номер блока, отсчитываемого от номера блока CROSS. При косвенной адресации предполагается, что нужный аргумент представлен некоторым параметром. Последний обозначается *, за которой следует целое число. Например, S*10 соответствует текущему значению накопителя, номер которого задан параметром 10 (буква S - означает накопитель). Косвенная адресация неприменима только для СЧА C1, M1, RN_n.

Аппаратно - ориентированные блоки.

К группе АО - блоков относятся:

SEIZE - блок занятия прибора;
RELEASE - освобождение прибора;
PREEMT - захват устройства;
RETURN - возврат захваченного прибора старому транзакту;
ENTER - вход в устройство (накопитель);
LEAVE - выход из накопитель;
LOGIG - изменение логических переключателей.

Введение в моделирующую программу устройств и накопителей позволяет автоматически регистрировать статическую информацию.

Для управления **ключами** используется оператор LOGIG. Предусмотрено три режима изменения состояния ключа: сброс в "0", установка в "1", инвертированное изменение состояния ключа на противоположное.

Динамически - ориентированные блоки.

В процесс моделирования транзакты создаются, порождают другие транзакты, собираются и уничтожаются. Каждому сообщению соответствует набор параметров, количество которых может быть установлено до 100. По умолчанию количество параметров принимается равным 12. Сообщениям можно присваивать приоритет от 0 до 127. По умолчанию приоритет равен 0. С динамической категорией объектов связаны блоки, которые можно разделить на пять групп: задержки, создания и уничтожения, изменения маршрутов, синхронизация, изменения атрибутов сообщений.

Группы: задержки: ADVANCE ;
 создания: GENERATE;
 уничтожения: TERMINATE, SPLIT, ASSEMBLE;
 изменения маршрутов: TRANSFER, LOOP, GATE, TEST;
 синхронизации: MATCH, GATHER;
 изменения атрибутов сообщений: ASSIGN, INDEX, MARK,
 PRIORITY.

Функции блоков:

ADVANCE - задержка транзактов;
GENERATE - генерации;
TERMINATE - уничтожения;
SPLIT - расщепления;
ASSEMBLE - соединения;

TRANSFER	- передачи;
LOOP	-организации цикла;
GATE	- проверка состояния;
TEST	- сравнения атрибутов;
MATCH	- синхронизации;
GATHER	- сбора;
ASSIGN	- изменений значений параметров;
INDEX	- увеличение индекса;
MARK	- ;
PRIORITY	- изменение приоритета;

Вычислительная категория

В вычислительной категории используются объекты 3-х видов: арифметические, логические, и функции. Арифметические объекты описываются блоком `variable` в режиме целых чисел и `FVARIABLE` в режиме с плавающей точкой. Название карты описывают арифметические действия над СЧА. Аргументы и результаты рассматриваются как целые числа. При вычислении используются операции: $+$, $-$, $*$, $/$ (с отбрасыванием остатка, d - деление по модулю (остаток считается положительным)). Допускается использование не более 5-ти скобок.

`FVARIABLE` - описывает арифметические сочетания различных СЧА, вычисляемых в режиме с плавающей запятой. От аргументов и промежуточных результатов не берётся целая часть, а только от конечного результата.

Блок - `BVARIABLE` - логическое высказывание, состоящее из некоторой совокупности СЧА и логических атрибутов. При вычислении используется 3 типа операторов: логические, условные и булевы, например, 2 `BVARIABLE` `M1 'LE' P6`.

Функции описываются с помощью блока `FUNCTION` в виде совокупности диапазонов, например:

```
3 FUNCTION RN1,C5
0,0/.35,11/.42,1.7/.75,2.2/1.0,3.8
```

Статическая категория

К ней относятся блоки:

QUEUE	- для занятия очереди;
DEPART	- для освобождения из очереди;
TABULATE	- для регистрации частоты попадания заданного СЧА;
TABLE	- для вывода характеристик таблицы;
SAVEVALUE	- для сохранения информации в специальных ячейках ОЗУ;
MSAVEVALUE	- для сохранения информации в ячейках ОЗУ;
MATRIX	- для описания матрицы;
INITIAL	- для присвоения ячейкам и матрицам начальных значений.

Группирующая категория

К ней относятся блоки LINK и UNLINK, позволяющие переводить сообщения из списка текущих активных событий в списки пользователя неактивных событий и обратно. Использование списков позволяет моделировать различные дисциплины обслуживания событий.

Специальные типы блоков

Они используются для составления сложных моделей. К ним относятся блоки вывода статистики (PRINT, TRACE, UNTRACE), изменения модели (EXECUTE, CHANGE) блоки BUFFER и HELP, а так же блоки управления группами транзактов (JOIN, REMOVE, EXAMINE, SCAN, ALTER).

Блок BUFFER даёт возможность для немедленного прекращения обработки транзакта и возврата к началу списка текущих событий.

HELP - для составления пользователем независимых программ, которые могут взаимодействовать с GPSS.

JOIN - для включения транзакта или числового значения в группу.

REMOVE - обратная функция JOIN.

EXAMINE - предоставляет возможность пользователю выбора пути, по которому пойдёт транзакт в зависимости от того, принадлежит он группе или нет.

SCAN - для анализа получения значений атрибутов транзактов и изменения их пути.

ALTER - для изменения значений атрибутов транзактов из группы.

Примеры решения задач моделирования на GPSS

Пример 1. Моделирование непрерывной случайной величины распределённой по экспоненциальному закону с $\lambda = 0.1$.

```
10  SIMULATE
20  EXPON FUNCTION RN1, C24
30  0.0/.1,.104/.2,.222/.3,.365/.4,.509/.5,.69
      .6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12
      .9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5
      .98,3.9/.99,4.6/.995,5.3/.995,6.2/.999,7.01.9997,8.0
40  GENERATE 10,FN$EXPON
50  MARK1
60  TABULATE XTIME
70  TERMINATE 1
80  XTIME TABLE P1,0,2,100
90  START 200
100 END
```

Модель включает 4 блока, выполняющие следующие функции:

40 - генерирование транзакта;

50 - присвоение параметру 1 транзакта значения, равного текущему значению часового времени;

60 - уничтожение транзакта;

10 - признак, необходимый для прогона модели;
 20 - описание функции (EXPON - метка, RN1 - генератор случайной функции, число пар координат-24);
 30 - задание значений пар координат функции;
 80 - определение таблицы; XTIME - метка, табулируемой величиной является P1 значение параметров последовательных транзактов, верхний предел первого интервала равен 0, ширина интервала - 2, общее число интервалов - 100;
 90 - признак ввода данных, необходимый для выполнения моделирования; прогон модели должен завершиться после прохождения через неё 200 транзактов;
 100 - признак конца программы.

Пример 2. Составить модель композиции двух случайных величин X_1 и X_2 имеющих экспоненциальные распределения с параметрами λ_1 и λ_2 ($X = X_1 + X_2$), удовлетворяющих обобщённому закону Эрланга 1-ого порядка: $g(t) = \lambda_1 \lambda_2 (e^{-\lambda_1} - e^{-\lambda_2}) / (\lambda_1 - \lambda_2)$.

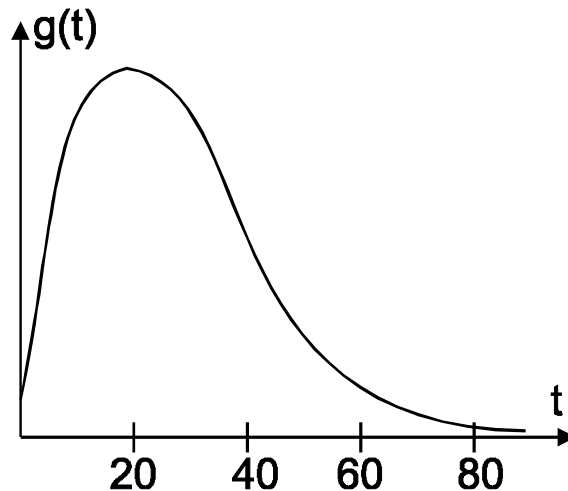


Рис. 10.1. К задаче моделирования композиции 2-х случайных величин.

```

10  SIMULAT
20  EXPON FUNCTION RN1, C24
30  0.0/.1,.104/.2,.222/.3,.365/.4,.509/.5,.69
    .6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12
    .9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5
    .98,3.9/.99,4.6/.995,5.3/.995,6.2/.999,7.01.9997,8.0

40  GENERATE 0,0,,1
50  ASSIGN 1,K500
60  INPUT ADVANCE 10,FN$EXPON
70  ADVANCE 20, FN$EXPON
80  TABULATE XTIME
90  LOOP 1,INPUT
100 TERMINATE 1
110 XTIME TABLE M1,0,5,100
  
```

120 START 1

130 END.

Функции блоков:

40 - генерирование 1-ого транзакта в момент времени $t=0$;

50 - присвоение параметру 1 значения, равного 500;

60 - моделирование экспоненциального распределённых временных интервалов с параметром λ_1 ;

70 - моделирование экспоненциального распределённых временных интервалов с параметром λ_2 ;

80 - формирование таблиц частот XTIME для суммарных интервалов;

90 - контроль числа проходов транзактов через сегмент блоков, начинающийся с блока INPUT;

100 - уничтожение транзакта.

Пример 3. Моделирование однолинейной системы с пуассоновским входящим потоком с параметром $\lambda = 0.1$ 1/сек. И экспоненциальным временем обслуживания с параметром $\mu = 0.2$ 1/сек.

10 SIMULATE

20 LINE EQU 1

30 EXPON FUNCTION RN1,C24

40 0.0/.1,.104/.2,.222/.3,.365/.4,.509/.5,.69

.6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12

.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5

.98,3.9/.99,4.6/.995,5.3/.995,6.2/.999,7.01.9997,8.0

50 GENERATE 10, FN\$EXPON

60 ASSIGN 1,LINE

70 QUEUE O1

80 SEIZE LINE

90 DEPART O1

100 ADVANCE 5, FN\$EXPON

110 RELEASE LINE

120 TABULATE XTIME

130 TERMINATE 1

140 XTIME TABLE M1,0,10,100

150 START 500

160 END.

9 блоков: 50 - генерирование транзактов;

60 - назначение параметру 1 транзакта номера, соответствующего прибору LINE;

70 - вхождение транзакта в очередь на прибор;

80 - занятие прибора;

90 - выход из очереди;

100 - моделирование обслуживания;

110 - освобождения прибора;

120 - формирование таблицы частот XTIME для времени прохождения транзакта;

130 - уничтожение транзакта;

20 - назначение величины 1 переменной LINE.

Пример 4. Моделирование работы однолинейной системы, имеющей 3 Пуассоновских потока требований с относительными приоритетами и параметрами $\lambda_1=0.01$ 1/сек., $\lambda_2=0.04$ 1/сек., $\lambda_3=0.05$ 1/сек. Экспоненциальный закон обслуживания $\mu_1=0.2$ 1/сек.

```
10  SIMULATE
20  LINE EQU 1
30  EX FUNCTION
40  0.0/.1,.104/.2,.222/.3,.365/.4,.509/.5,.69
    .6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12
    .9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5
    .98,3.9/.99,4.6/.995,5.3/.995,6.2/.999,7.01.9997,8.0
50  GENERATE 100, FN$EX,,,3
60  TRANSFER ,INPUT
70  GENERATE 25, FN$EX,,,2
80  TRANSFER ,INPUT
90  GENERATE 20, FN$EX,,,1
100 INPUT ASSIGN 1,LINE
110 QUEUE LINE
120 SEIZE LINE
130 DEPART LINE
140 ADVANCE 5, FN$EX
150 RELEASE LINE
160 TERMINATE 1
170 START 1000
```

60, 80 - безусловная передача транзактов;

50, 70, 90 - генерирование транзактов с приоритетами 3, 2, 1.

Планирование машинных экспериментов с моделями систем.

Методы планирования эксперимента на модели.

Основная задача планирования машинных экспериментов заключается в получении необходимой информации об исследуемой системе при ограниченных ресурсах (затраты машинного времени, памяти и т.п.). К числу частных задач, решаемых при планировании машинных экспериментов, относятся задачи уменьшения затрат машинного времени на моделирование, уменьшения погрешности результатов моделирования, проверки адекватности модели и т.п.

Эффективность машинных экспериментов существенно зависит от выбора плана эксперимента, т.к. именно план определяет объём и порядок проведения вычислений на ЭВМ, приёмы накопления и статистической обработки результатов моделирования системы. Поэтому основная задача

планирования машинных экспериментов с моделью формируется следующим образом: необходимо получить об объёме моделирования, заданном в виде моделирующего алгоритма (программы) при минимальных или ограниченных затратах машинных ресурсов на реализацию процесса моделирования.

Таким образом, при машинном моделировании необходимо не только рационально планировать и проектировать саму модель системы, но и процесс её использования, т.е. проведения с ней эксперимента.

При планировании машинных экспериментов возникает целый ряд проблем, взаимно связанных как с особенностью функционирования моделируемого объекта, так и с особенностью машинной реализации модели и обработки результатов эксперимента. В первую очередь к таким относятся проблемы построения плана машинного эксперимента, стохастической сходимости результатов, ограниченности машинных ресурсов, уменьшения дисперсии оценок, полученных на машинной модели и т.д.

Рассмотрим основные понятия теории планирования эксперимента. В планировании эксперимента различают входные (изогенные) и выходные (эндогенные) переменные: $x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_e$. Входные переменные в ТПЭ называют факторами а выходные — реакциями. Каждый фактор $x_i, i=1, 2, \dots, k$ может принимать в эксперименте одно или несколько значений, называемых уровнями. Фиксированный набор уровней факторов определяет одно из возможных состояний рассматриваемой системы. Одновременно этот набор представляет собой условия проведения одного из возможных экспериментов.

Каждому фиксированному набору уровню факторов соответствует определённая точка в многомерном пространстве, называемая факторным пространством. Эксперименты не могут быть реализованы во всех точках факторного пространства, а лишь в принадлежащих допустимой области, как это например оказано для случая двух факторов X_1 и X_2 на рисунке (см. ниже рис. 11.1.).

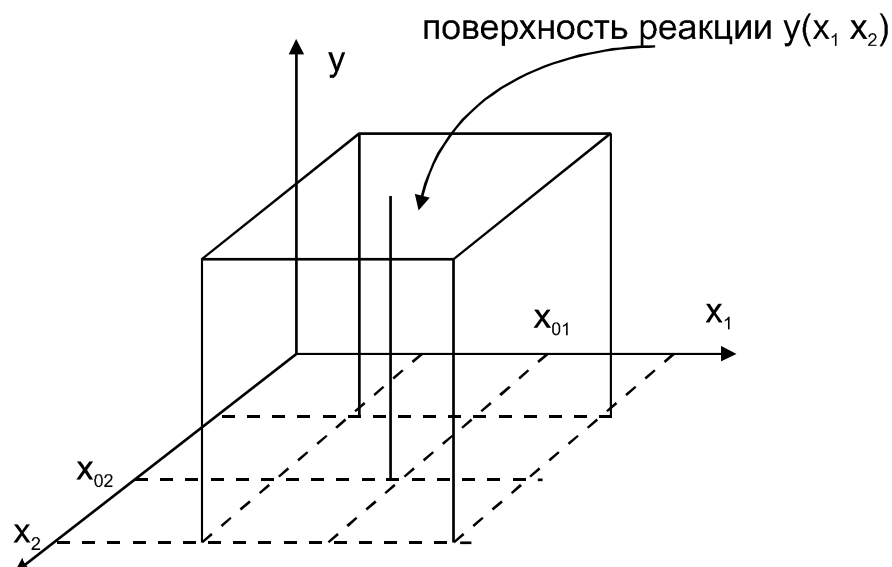


Рис. 11.1. Геометрическое представление поверхности реакции.

Реакцию (отклик) системы можно представить в виде зависимости: $y_l = \Psi_l(x_1, x_2, \dots, x_k)$; $l = 1 \dots m$. Функцию Ψ_l , связанную с факторами, называют функцией отклика, а её геометрический образ – поверхностью отклика. Исследователь заранее не известен вид зависимостей Ψ_l , $l = 1 \dots m$, поэтому используют приближение соотношения: $y_l = \Psi_l(x_1, x_2, \dots, x_k)$, $l = \overline{1, m}$.

Зависимость и Ψ_l находятся по данным эксперимента. Последний необходимо поставить так, чтобы при минимальных затратах ресурсов (числе испытаний), варьируя выходные значения по специально сформулированным правилам, построить математическую модель системы и оценить её характеристики. Факторы при проведении эксперимента могут быть управляемыми и неуправляемыми, количественными или качественными, фиксированными и случайными. Фактор относится к изучаемым, если он включён в модель для изучения свойств системы. Количественными факторами являются интенсивности входящих потоков заявок, интенсивности потоков обслуживания, ёмкости накопителей, количество обслуживающих каналов и другие. Качественным факторам не соответствует числовая шкала (дисциплины постановки на очередь, обслуживание каналов и другие).

Фактор является управляемым, если его уровни целенаправленно выбираются экспериментатором.

При планировании эксперимента обычно изменяются несколько факторов. Основными требованиями, предъявляемыми к факторам - независимость и совместимость. Совместимость означает, что все комбинации факторов осуществимы.

Для выбора конкретной модели планирования эксперимента необходимо сформулировать такие её особенности, как адекватность, содержательность, простота.

План эксперимента обычно используется для определения экстремальной характеристики объекта. Поэтому планирование эксперимента называется экстремальным. В планировании эксперимента наибольшее значение нашли модели в виде алгебраических полиномов.

Предполагаем, что изучается влияние K количественных факторов x_i на некоторую η в отведённый для экспериментирования локальной области факторного пространства ограниченного $x_{i \min} - x_{i \max}$, $i = 1 \dots k$.

Функцию отклика обычно выбирают линейной или квадратичной.

$$\eta = b_0 + \sum b_i x_i + \sum b_{ij} x_i x_j = f(\vec{x}) \vec{B} \quad (1)$$

где $\vec{f}(\vec{x})$ - вектор с элементами $f_\alpha(\vec{x})$, $\alpha = \overline{0, d}$, входящих в исходный

полином; $\vec{B}$ - вектор коэффициентов. Для двух факторов имеем: $f_0=1$, $f_1=x_1$, $f_2=x_2$, $f_{12}=x_1 x_2$, $f_{11}=x_1^2$, $f_{22}=x_2^2$. $\vec{B} = (b_0, b_1, b_2, b_{12}, b_{11}, b_{22})$.

Так как полином (1) содержит d коэффициентов, то план эксперимента должен содержать $N \geq d$ различных экспериментальных точек:

$$D = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{12} & x_{22} & \dots & x_{k2} \\ \dots & \dots & \dots & \dots \\ x_{1N} & x_{2N} & \dots & x_{kN} \end{bmatrix}$$

где x_{in} - значение, которое принимает i -ая переменная в u -ом испытании. $i=1 \dots k$, $u=1 \dots N$. Матрица D называется планом эксперимента.

Реализовав испытания в N очках области факторного пространства, определённом планом эксперимента, получим вектор наблюдений имеющий следующий вид:

$$\bar{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

где y_u - реакция соответствующей u -ой точке плана.

Плану эксперимента поставим в соответствие матрицу планирования:

$$x = \begin{bmatrix} f_{01} & f_{11} & f_{21} & \dots & f_{111} & f_{221} \\ f_{02} & f_{12} & f_{22} & \dots & f_{112} & f_{222} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ f_{0N} & f_{1N} & f_{2N} & \dots & f_{11N} & f_{22N} \end{bmatrix}$$

где f_{il} , f_{ijl} - координатные функции при соответствующих коэффициентах модели, в l -ом эксперименте.

Построению плана эксперимента предшествует проведение ряда неформализованных действий (принятия решения) направленных на выбор локальной области факторного пространства G .

Необходимо учитывать, что как только модель сформирована включение дополнительных факторов для уточнения модели невозможно. Вначале следует выбрать границы $x_{i \min}$ и $x_{i \max}$ области определения факторов исходя из свойств объекта. Например, температура при термобарических экспериментах не может быть ниже абсолютного нуля и выше температуры плавления материала из которого изготовлена термобарокамера.

После определения области G необходимо найти нулевые (основные) уровни факторов и интервалы варьирования Δx_i , $i=1 \dots k$.

Эксперимент, в котором реализуются все возможные сочетания уровней факторов, называется полным факторным экспериментом (ПЭФ). Если выбранная модель включает только линейные члены полинома и их произведения, то для оценки коэффициентов модели используется ПЭ с варьированием всех k факторов на двух уровнях, т.е. $q=2$. Такие планы называются планы типа 2^k , где $n=2^k$ - число всех возможных испытаний.

Начальным этапом ПЭ для получения коэффициентов линейной модели основан на варьировании факторов на двух уровнях: нижнем x_{in} и верхнем x_{iv} ,

симметрично расположенных относительно основного уровня x_{i0} , $i=1 \dots k$. Геометрическая интерпретация показана ниже на рис. 11.2.:

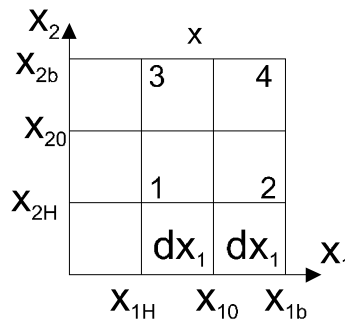


Рис. 11.2. ПЭФ типа 2^2 .

Для упрощения записи условий каждого эксперимента факторы кодируют в виде безразмерных величин $\tilde{x}_i = (x_i - x_{i0}) / \Delta x_i$, $i = 1, 2, \dots, k$. Средний уровень кодированного фактора является нулём 0, граничные значения соответственно +1 и -1.

Стратегическое планирование машинных экспериментов с моделями систем

Можно выделить стратегическое и тактическое ПЭ на моделях систем.

Стратегическое планирование – ставит своей целью получение необходимой информации о системе S с помощью модели M_M , реализованной на ЭВМ.

Оно аналогично внешнему проектированию при создании системы S .

Тактическое планирование – определяет способы проведения каждой серии испытаний машинной модели M_M . Оно аналогично внутреннему проектированию системы S .

Рассмотрим элементы стратегического планирования ПЭ. Его целью может быть:

1. Получение функции реакции системы от независимых фактов:
 $y = f(b_0, b_1, \dots, x_1, x_2, \dots, x_k)$
2. Нахождение экстремума: $f(b_0, b_1, \dots, x_1, x_2, \dots, x_k)$.

Во 2-ом случае для определения наилучшей комбинации фактов могут быть использованы методы систематической или случайной выборки.

К систематическим относятся методы:

- одного фактора;
- предельного анализа;
- наискорейшего спуска;
- равномерной сетки.

Проблемой является большое количество факторов. Для $k=10$ ПЭФ должен состоять из 1024 точек. Используют неполные планы, метод "поверхности реакции".

Следующей проблемой является многокомпонентность функции реакции. Здесь можно использовать последовательное однокомпонентное ПЭ. Этот подход не всегда возможен из-за связанности компонентов. Используются интегральные оценки с применением весовых функций, функций полезности и т.д.

Другой проблемой является стохастическая сходимость результатов ПЭ. В качестве результатов ПЭ используются средние некоторых распределений, для оценки которых применяют выборочные средние, найденные путём многократны прогонов модели на ЭВМ. Сходимость выборочных средних с ростом объема выборки называется стохастической. Эта сходимость, как правило, медленная. Если σ - стандартное отклонение среднего N наблюдений будет равно $\sigma/\sqrt{N}$, т.е. для уменьшения случайной выборки в k раз требуется увеличить объем выборки в k^2 раз.

Планирование машинного эксперимента представляет собой итерационный процесс, когда выбранная модель плана эксперимента проверяется на реализуемость, а затем, если это необходимо, вносят соответствующие коррективы в модель.

Планирование эксперимента с моделью проводится в несколько этапов:

- 1) построение структурной модели;
- 2) построение функциональной модели.

Структурная модель ПЭ характеризуется числом факторов и числом уровней для каждого фактора. Из опыта известно, что 20% факторов определяют 80% свойств системы.

Ортогональное распределение плана упрощает определение коэффициентов аппроксимации. Упрощение дает принятие числа уровней всех факторов одинаковыми (не больше 3). Функциональная модель ПЭ определяет количество элементов структурной модели N_ϕ , т.е. необходимое число различных информационных точек N_ϕ . Причём $N_\phi < N_c$, где $N_c = q_1 q_2 \dots q_k$ – число экспериментов ПФЭ.

Тактическое планирование машинных экспериментов с моделями систем

Здесь решают проблемы:

- определения начальных условий и их влияния на достижения установившегося результата при моделировании;
- обеспечения точности и достоверности результатов моделирования;
- уменьшения дисперсии оценок характеристик процесса функционирования моделируемых систем;
- выбора правил автоматической остановки имитационного эксперимента с моделями.

Рассмотрим ПФЭ типа 2^3 :

номер испытания	1	2	3	4	5	6	7	8
x_1	-1	-1	-1	-1	+1	+1	+1	+1
x_2	-1	-1	+1	+1	-1	-1	+1	+1
x_3	-1	+1	-1	-1	-1	+1	-1	+1

ПЭФ даёт возможность определить не только коэффициенты регрессии, соответствующие линейным эффектам, но и коэффициенты регрессии соответствующие всем эффектам взаимодействия. Эффект взаимодействия двух или более факторов появляется при одновременном варьировании этих

факторов, когда действие каждого из них на выход зависит от уровня, на которых находятся другие факторы.

Для оценки свободного члена b_0 и определения эффектов взаимодействия b_{12} , b_{13} , ..., b_{123} ... план эксперимента D расширяют до матрицы планирования X путём добавления соответствующей фиктивной переменной: единичного столбца $\bar{x}_0$ и столбцов произведений $\bar{x}_1\bar{x}_2, \bar{x}_1\bar{x}_3, \dots, \bar{x}_1\bar{x}_2\bar{x}_3$ как показано, например, для ПЭФ типа 2^3 в таблице (см. ниже):

Номер испытания	$\bar{x}_0$	План ПЭФ			$\bar{x}_1\bar{x}_2$	$\bar{x}_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$	Реакция y
		$\bar{x}_0$	$\bar{x}_2$	$\bar{x}_3$					
1	+1	+1	+1	+1	+	+	+	+	y_1
2	+1	-1	+1	+1	-	-	+	-	y_2
3	+1	+1	-1	+1	-	+	-	-	y_3
4	+1	-1	-1	+1	+	-	-	+	y_4
5	+1	+1	+1	-1	+	-	-	-	y_5
6	+1	-1	+1	-1	-	+	-	+	y_6
7	+1	+1	-1	-1	-	-	+	+	y_7
8	+1	-1	-1	-1	+	+	+	-	y_8

Как видно из рассмотренных ПЭ типа 2^2 и 2^3 количество испытаний ПЭФ значительно превосходит число определяемых коэффициентов линейной модели плана эксперимента, что увеличивает расход ресурсов ЭВМ по времени. Возникает проблема сокращения количества экспериментов. С этой целью рассмотрим построение планов так называемого дробного факторного эксперимента (ДФЭ). Пусть имеется ПЭФ типа 2^2 . Используя матрицу планирования X, например приведённую в предыдущей таблице, можно вычислить коэффициенты и предусмотреть результаты в виде уравнения:

$$y = b_0 + b_1\bar{x}_1 + b_2\bar{x}_2 + b_{12}\bar{x}_1\bar{x}_2$$

N/n	$\bar{x}_0$	План ПЭФ		$\bar{x}_1\bar{x}_2(\bar{x}_0)$	Отклик y
		$\bar{x}_1$	$\bar{x}_2$		
1	1	+1	+1	+1	y_1
2	1	-1	+1	-1	y_2
3	1	+1	-1	-1	y_3
4	1	-1	-1	+1	y_4

Если в выбранных интервалах варьирования уровня процесс можно описать линейной моделью, то достаточно определить три коэффициента b_0, b_1, b_2 . Т.о. остаётся одна степень свободы, которую можно использовать для построения плана эксперимента D для 3-х переменных, в которых уровни 3-его фактора изменяются как в таблице рассмотренной немного раньше для столбца $\bar{x}_1\bar{x}_2$ (эффектов взаимодействия).

Мы получим так называемый дробный факторный эксперимент. В нём уже не будет отдельных оценок для коэффициентов регрессии, как в ПФ, они будут рассчитываться по формулам:

$$\tilde{b}_1 = \beta_1 + \beta_{23}; b_2 = \beta_2 + \beta_{13}; b_3 = \beta_3 + \beta_{12}.$$

При постулировании линейной модели все парные взаимодействия не учитываются. Т.е. вместо γ испытаний в ПФ для 3-х факторов получим 4 испытания в ДФЭ. Правило проведения ДФЭ формулируется так: для сокращения числа испытаний новому фактору присваивается значение вектор-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь.

При проведении эксперимента из 4-х испытаний для оценки влияния 3-х факторов пользуются половинный ПФ типа 2^3 , так называемой "полу репликой". Если приравнять x_3 и x_1x_2 , что можно получить 2-ую "полу реплику".

Для обозначения дробных реплик, в которых λ линейных эффектов приравнены к эффектам взаимодействия, пользуются условным обозначением $2^{k-\lambda}$. Например, "полу реплика" от 2^6 записывается в виде 2^{6-1} , а "четверть реплика" 2^{6-2} .

Смешивание надо производить так, чтобы основные коэффициенты были смешаны с коэффициентами при взаимодействиях самого высокого порядка. Не рекомендуется использовать реплики для $n \geq 15$.

Следует иметь в виду, что малый шаг варьирования Δx_j ($j=1 \dots n$) может повлечь статистическую незначимость оценки коэффициента уравнения регрессии. В случае, если полученная мат. модель окажется неадекватной, проводятся эксперименты с меньшим шагом варьирования.

Если линейные модели, построенные с помощью ПФ и ДФЭ, неадекватны, то переходят к построению квадратичных моделей.

Оптимальный план для квадратичной модели целесообразно строить таким образом, чтобы он включал точки плана для линейной модели. Это позволяет сократить число опытов.

V. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВЫХ ПРОЕКТОВ (РАБОТ)

Предусмотрена курсовая работа, в которой раскрывается некоторая тема (темы) из списка, приведенного в рабочей программе. Студент подбирает учебную и научную литературу по теме курсовой работы, знакомится с этой литературой и составляет план работы. Далее раскрывает этот план. Выбор темы курсовой работы и практического задания оговариваются с каждым студентом отдельно. Работа оформляется в соответствии со стандартом университета.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ (ПРАКТИКУМОВ)

Для выполнения лабораторных заданий студенту необходимо иметь конспект лекций. Студент знакомится с заданием и выполняет его, опираясь на конспект лекций. При выполнении задания связанного с программированием, студент разрабатывает алгоритм решения предложенной задачи, набирает текст программы, отлаживает и тестирует программу. При написании программы необходимо использовать конструкции языка программирования, используемого математического пакета.

VII. ЛАБОРАТОРНЫЙ ПРАКТИКУМ (ПЕРЕЧЕНЬ ОСНОВНЫХ ТЕМ)

Перечень основных тем лабораторных занятий содержится в рабочей программе, для выполнения лабораторной работы студенту необходимо иметь конспект лекций. Сдача лабораторных работ проводится в форме собеседования с каждым студентом отдельно.

МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ

Введение

1. Линейные интерполяционные полиномы
2. Линейные интерполяционные полиномы для дискретизованной области
3. Одномерная задача стационарной теплопроводности
4. Двумерная задача стационарной теплопроводности
5. Сравнение с аналитическим решением
6. Автоматическая генерация расчетной сетки и пример расчета температурного поля в области сложной формы
7. Элементы высокого порядка

ВВЕДЕНИЕ

Метод конечных элементов основан на идее аппроксимации непрерывной функции (в физической интерпретации - температуры, давления, перемещения и т.д.) дискретной моделью, которая строится на множестве кусочно-непрерывных функций, определенных на конечном числе подобластей, называемых конечными элементами. Исследуемая геометрическая область разбивается на элементы таким образом, чтобы на каждом из них неизвестная функция аппроксимировалась пробной функцией (как правило, полиномом). Причем эти пробные функции должны удовлетворять граничным условиям непрерывности, совпадающим с граничными условиями, налагаемыми самой задачей. Выбор для каждого элемента аппроксимирующей функции будет определять соответствующий тип элемента.

В данной работе рассматривается вычислительный алгоритм метода конечных элементов в формулировке, основанной на процедуре минимизации функционала, соответствующего решаемой непрерывной задаче. В результате выполнения указанной процедуры происходит замещение уравнения или системы уравнений в частных производных системой нелинейных уравнений, имеющих в качестве коэффициентов аппроксимирующие функции, которые фактически являются значениями искомой функции в вершинах разбиения.

При построении алгоритма используется математический аппарат, предоставляемый системой Maple.

1. ЛИНЕЙНЫЕ ИНТЕРПОЛЯЦИОННЫЕ ПОЛИНОМЫ

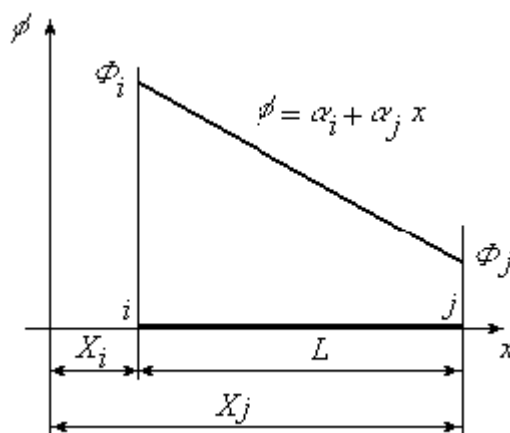
Рассмотрим особенности построения линейных интерполяционных соотношений, соответствующих симплекс-элементам.

> **restart:with(linalg):**

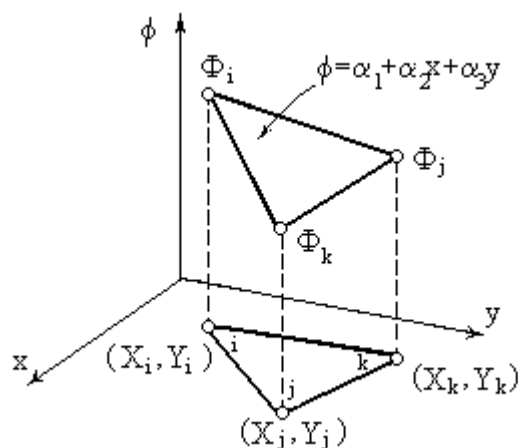
Warning, the protected names norm and trace have been redefined and unprotected

Ниже на рисунках изображены:

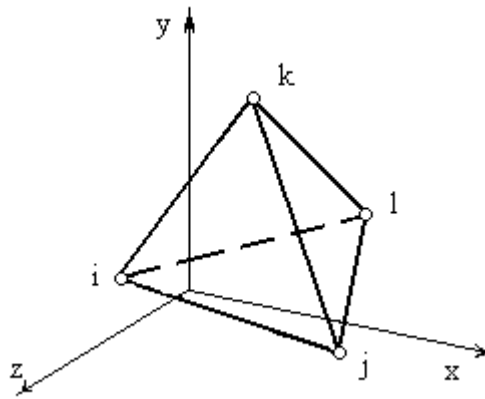
- одномерный симплекс-элемент, представляющий собой прямолинейный отрезок длины L с двумя узлами, по одному на каждом конце;



- двумерный симплекс-элемент. Это треугольник с прямолинейными сторонами и тремя узлами, по одному в каждой вершине;



- трехмерный симплекс-элемент. Он представляет собой тетраэдр. Четыре его узла обозначены индексами i, j, k, l , причем обход узлов i, j, k осуществляется в том порядке, как они записаны, против часовой стрелки. Узел l расположен в вершине, находящейся вне плоскости узлов i, j, k .



В общем (трехмерном) случае линейный интерполяционный полином для симплекс-элемента имеет вид:

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z$$

(В двумерном случае $z = 0$, а в одномерном - $z = 0$ и $y = 0$.)

Определим коэффициенты

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4$$

с помощью известных значений искомой функции в узловых точках.

$$P := [\Phi_1, \Phi_2, \Phi_3, \Phi_4]$$

из решения матричного уравнения

$$\alpha = C^{(-1)} \Phi$$

где C - следующая матрица

$$C := \begin{bmatrix} 1 & X_1 & Y_1 & Z_1 \\ 1 & X_2 & Y_2 & Z_2 \\ 1 & X_3 & Y_3 & Z_3 \\ 1 & X_4 & Y_4 & Z_4 \end{bmatrix}$$

Объединив в матрицу B множители при α в интерполяционной формуле


```
> B := [1, x, y, z]
```

получим матрицу так называемых функций формы (базисных функций), вычисляемую как

```
> N := multiply(B, inverse(C))
```

Тогда интерполяционный полином для искомой функции может быть получен следующей операцией

```
> phi := multiply(N, P)
```

Рассмотрим трехмерный симплекс-элемент, используемый для аппроксимации распределения непрерывной функции внутри тетраэдра.

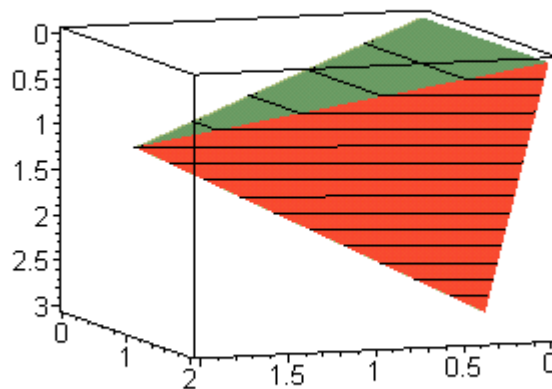
Вначале определяем массивы численных данных с известными координатами четырех узлов (см. рис):

```
> X:=[1,0,2,1]:Y:=[2,0,0,0]:Z:=[1,0,0,3]:
```

Отобразим этот тетраэдр на экране дисплея

```
> with(plots):  
polygonplot3d([[X[1],Y[1],Z[1]],  
[X[2],Y[2],Z[2]],  
[X[3],Y[3],Z[3]],  
[X[4],Y[4],Z[4]],  
  
[X[2],Y[2],Z[2]],  
[X[1],Y[1],Z[1]]],  
axes=boxed,orientation=[-160,-100],  
shading=XY,style=PATCHCONTOUR,light=[20,10,5,3,1],lightmodel='light3')  
;
```

```
Warning, the name changecoords has been redefined
```



Пусть значениями функции в соответствующих узловых точках будут

```
>  $\Phi := [40, 34., 26, 15]$ 
```

Тогда искомый интерполяционный полином примет вид

```
> phi;
```

$$7.5000000003y + 34. - 4.000000000x - 5.000000000z$$

Во внутренней точке с координатами (1, 0.5, 1) имеем значение функции

```
> subs(x=1,y=0.5,z=1,phi);
```

28.75000000

Аналогично получим полином, соответствующий двумерному симплекс-элементу:

```
> P := [Phi[1], Phi[2], Phi[3]]:
```

```
> C:=matrix([[1,X[1],Y[1]],  
[1,X[2],Y[2]],  
[1,X[3],Y[3]] ]):
```

```
> B:=(1,x,y):
```

```
> N:=multiply(B,inverse(C)):
```

```
> phi:=multiply(N,P):
```

Определяем массивы численных данных:

координаты узлов -

```
> X:=[1,0,2]:Y:=[2,0,0]:
```

Значения функции в соответствующих узловых точках

```
> Phi:=[40,34.,26]:
```

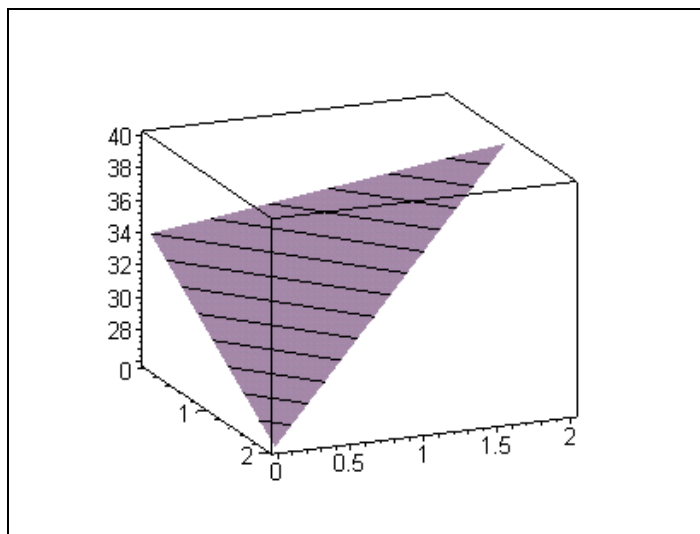
Искомый интерполяционный полином будет иметь вид

```
> phi;
```

$$5.000000000 y + 34. - 4.00000000 x$$

В отличие от трехмерного случая, полученную аппроксимирующую функцию можно представить наглядно:

```
> with(plots):  
polygonplot3d([  
[X[1],Y[1],Phi[1]],  
[X[2],Y[2],Phi[2]],  
[X[3],Y[3],Phi[3]]],axes=boxed,style=PATCHCONTOUR);
```



В точке с координатами (1, 0.5) имеем значение

```
> subs(x=1,y=0.5,phi);
```

$$32.50000000$$

Теперь для одномерного симплекс-элемента:

```
> restart:with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> P := [Phi[1], Phi[2]]:  
> C:=matrix([[1,X[1]],  
[1,X[2]] ]):  
> B:=(1,x):  
> N:=multiply(B,inverse(C)):  
> phi:=multiply(N,P):
```

Массивы численных данных:

координаты узлов -

```
> X:=[1,0]:
```

значения функции в соответствующих узловых точках

```
> Phi:=[40,34.]:
```

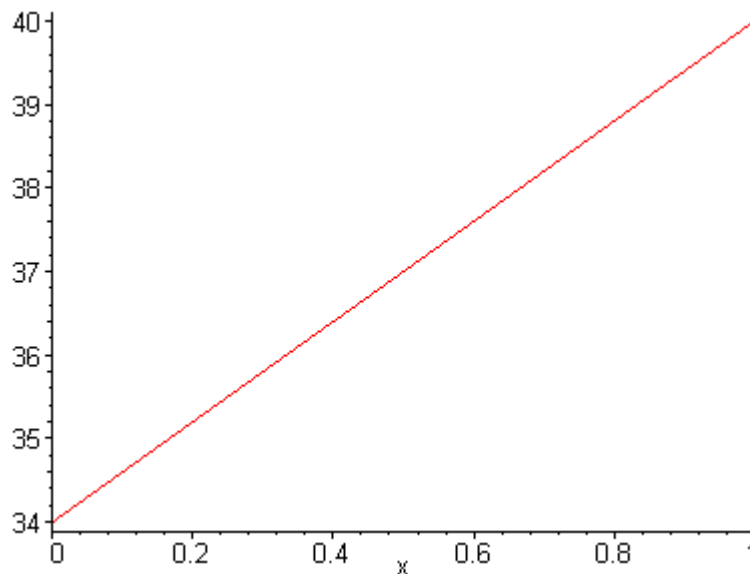
Искомый интерполяционный полином:

```
> phi;
```

$6. x + 34.$

Изображаем его на графике:

```
> plot(phi,x=0..1);
```



В точке с координатой $x = 0.5$ имеем значение искомой функции

> **subs(x=0.5,phi);**

37.0

Замечание

В данной работе интерполяционные соотношения использованы при рассмотрении скалярной величины. При интерполировании векторных величин, например перемещений в каждом узле необходимо определить более одной неизвестной (степени свободы). В этом случае векторную величину можно представить ее компонентами, которые рассматриваются как неизвестные скалярные величины. Каждый узел будет содержать одну, две или три неизвестные в зависимости от того, какая задача рассматривается - одномерная, двумерная или трехмерная. Причем в одномерной задаче представления векторной и скалярной величин внутри элемента совпадают, так как в обоих случаях в каждом узле отыскивается только одна неизвестная:

$\phi := \text{multiply}(N, P)$

В двумерной, трехмерной, или в общем случае n -мерной задаче необходимо аппроксимировать каждую компоненту неизвестной, например:

for i to n do $\phi_i := \text{multiply}(N, P_i)$ end do

Упражнение

Узловые значения температуры для треугольного симплекс-элемента равны $\Phi=130$, $\Phi=100$, $\Phi=125$. Выясните, где изотерма 125 пересекает границы элемента.

2. ИНТЕРПОЛЯЦИОННЫЕ ПОЛИНОМЫ ДЛЯ ДИСКРЕТИЗИРОВАННОЙ ОБЛАСТИ

> **restart:with(linalg):**

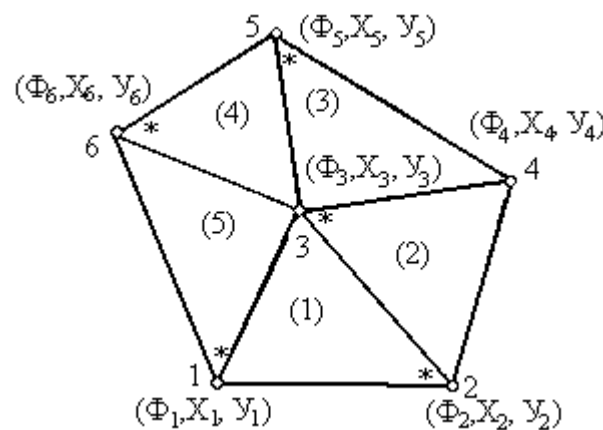
Warning, the protected names norm and trace have been redefined and unprotected

В общем виде интерполяционный полином, полученный в п.1 будет иметь вид

$$\phi_e = N_e \Phi_e$$

где e - индекс, указывающий на отдельный элемент.

Технику включения двумерного элемента в область проиллюстрируем на примере простой пятиэлементной конфигурации, показанной на следующем рисунке



Определим числовые массивы координат пяти узлов и значений искомой функции в указанных точках:

> **X:=[1,3,2,4,2,0]: Y:=[0,0,1,1,2,1]: Phi:=[2.5,3,3,2,3.5,3]:**

и составим массив, характеризующий всю пятиугольную область:

> **G1:=array(
[
[X[1],Y[1],X[2],Y[2],X[3],Y[3],Phi[1],Phi[2],Phi[3]],**

```
[X[3],Y[3],X[2],Y[2],X[4],Y[4],Phi[3],Phi[2],Phi[4]],
[X[5],Y[5],X[3],Y[3],X[4],Y[4],Phi[5],Phi[3],Phi[4]],
[X[6],Y[6],X[3],Y[3],X[5],Y[5],Phi[6],Phi[3],Phi[5]],
[X[1],Y[1],X[3],Y[3],X[6],Y[6],Phi[1],Phi[3],Phi[6]]
]);
```

$$G1 := \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 1 & 2.5 & 3 & 3 \\ 2 & 1 & 3 & 0 & 4 & 1 & 3 & 3 & 2 \\ 2 & 2 & 2 & 1 & 4 & 1 & 3.5 & 3 & 2 \\ 0 & 1 & 2 & 1 & 2 & 2 & 3 & 3 & 3.5 \\ 1 & 0 & 2 & 1 & 0 & 1 & 2.5 & 3 & 3 \end{bmatrix}$$

Зададим вектор коэффициентов **B** (для вычисления базисных функций) и вектор **phi**, в котором будут накапливаться полиномы, аппроксимирующие непрерывную функцию внутри каждого элемента

```
> B:=vector([1,x,y]):phi:=vector(5):
```

Оформим цикл матричных вычислений

```
> for e to 5 do
C:=matrix([[1,G1[e,1],G1[e,2]],
[1,G1[e,3],G1[e,4]],
[1,G1[e,5],G1[e,6]] ]):
N:=multiply(B,inverse(C)):
P:=(G1[e,7],G1[e,8],G1[e,9]):
phi[e]:=multiply(N,P):
od:
```

Вектор полученных полиномов имеет вид

```
> evalm(phi);
```

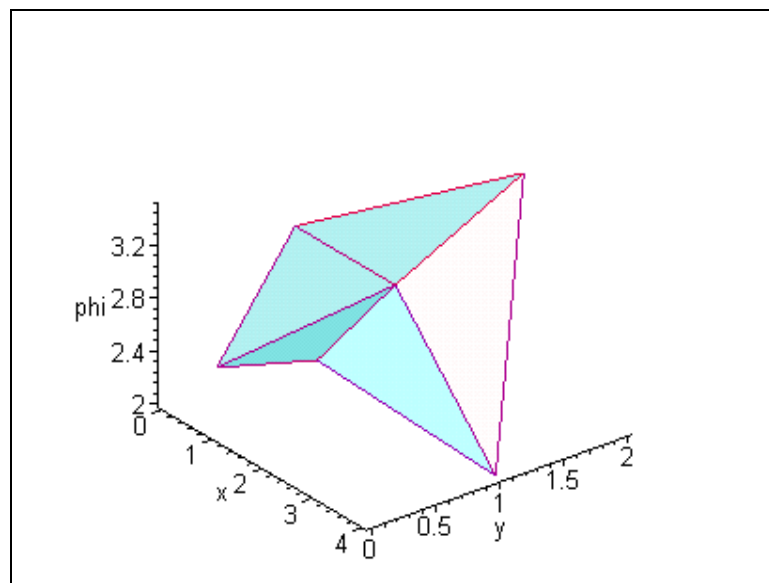
$$\begin{bmatrix} 2.250000000 + .250000000 x + .250000000 y, \frac{9}{2} - \frac{1}{2}x - \frac{1}{2}y, 3.5 + .5y - \frac{1}{2}x, 2.5 + .5y, \\ 2.500000000 + .500000000 y \end{bmatrix}$$

Изобразим картину распределения Φ на графике

```
> with(plots):
list_point :=
[
seq(
```

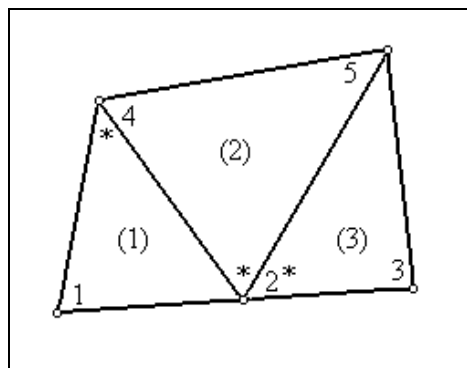
```
[
[G1[S,1],G1[S,2],G1[S,7]],
[G1[S,3],G1[S,4],G1[S,8]],
[G1[S,5],G1[S,6],G1[S,9]]
],
S=1..5]:
polygonplot3d(list_point, scaling=UNCONSTRAINED, axes=FRAMED,
titlefont=[TIMES,ROMAN,12], shading=Z,labels=[x,y,phi], style=hidden,
orientation=[-50,40],lightmodel=light3);
```

Warning, the name changecoords has been redefined



Упражнение

Получите интерполяционные соотношения для элементов, составляющих область, показанную на рисунке, если заданы координаты точек дискретизации и значения искомой функции в них:



> **X:=[0,2,4,0.25,3.75]: Y:=[0,0.25,0.51,2,2.75]: Phi:=[100,80,60,70,45]:**

Изобразите ситуацию на графике.

3. ОДНОМЕРНАЯ ЗАДАЧА ТЕПЛОПРОВОДНОСТИ

> **restart:**

Рассмотрим одномерный поток тепла в стержне с теплоизолированной боковой поверхностью (рис.1).

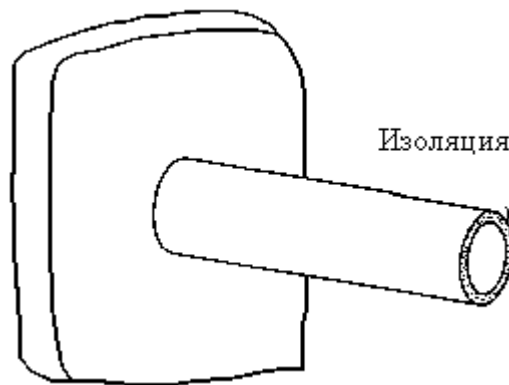


Рис.1. Модель теплопроводности в стержне

В вариационном исчислении устанавливается, что для минимизации функционала

$$\chi = \int \frac{K \left(\frac{\partial}{\partial x} T(x) \right)^2}{2} dV + \int q T + \frac{1}{2} \alpha (T - T_w)^2 dS \quad (1)$$

необходимо, чтобы удовлетворялось дифференциальное уравнение

$$K \left(\frac{\partial^2}{\partial x^2} T(x) \right) = 0 \quad (2)$$

с граничными условиями

$$K \left(\frac{\partial}{\partial x} T(x) \right) + q + \alpha (T - T_w) = 0 \quad (3)$$

В формулах (1)-(3): T - температура, K - коэффициент теплопроводности, q - тепловой поток заданной интенсивности, α - коэффициент конвективного теплообмена, T_w - температура окружающей среды,

V - объем стержня, S - площадь поперечного сечения стержня, x - координата вдоль оси стержня.

Таким образом, любое распределение $T(x)$, при котором функционал χ (1) становится минимальным, является решением исходной задачи теплопроводности (2)-(3).

> **with(linalg):**

Представим стержень дискретной моделью, содержащей шесть узловых точек:

> **m:=6:**

Phi:=vector(m):

с координатами

> **X:=[0,1,2,3,4,5]:**

Пусть площадь поперечного сечения стержня будет

> **A:=Pi:**

В граничных точках известны значения температуры:

> **Phi[1]:=100:Phi[6]:=0:**

Следующая часть программного кода формирует рабочий массив данных G1

> **for i to m do**

q[i]:=0: alpha[i]:=1: Tw[i]:=0:

od:

n:=m-1:

G1:=array(1..n,1..11):

for i to n do

K[i]:=1:

G1[i,1]:=X[i]: G1[i,2]:=X[i+1]:

G1[i,3]:=Phi[i]: G1[i,4]:=Phi[i+1]:

```

G1[i,5]:=q[i]: G1[i,6]:=q[i+1]:
G1[i,7]:=alpha[i]:G1[i,8]:=alpha[i+1]:
G1[i,9]:=Tw[i]: G1[i,10]:=Tw[i+1]:G1[i,11]:=K[i]:
od:
print (' число элементов - ` ,n);

```

число элементов -, 5

```
> evalm(G1);
```

$$\begin{bmatrix} 0 & 1 & 100 & \Phi_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 2 & \Phi_2 & \Phi_3 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 2 & 3 & \Phi_3 & \Phi_4 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 3 & 4 & \Phi_4 & \Phi_5 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 4 & 5 & \Phi_5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

В отдельных узлах можно задать тепловой поток (по умолчанию для остальных i узлов $q[i]=0$)

>

В узлах контактирующих с окружающей средой можно задать температуру последней Tw

(по умолчанию $Tw[i]=0$)

и известные коэффициенты теплоотдачи окружающей среде

(по умолчанию $alpha[i]=1$)

>

>

Также можно ввести коэффициенты теплопроводности в отдельных элементах

(по умолчанию $K[j]=1$)

>

В следующей ячейке автоматически вычисляются интерполяционные полиномы и

функционалы для каждого из n отдельных элементов, в результате получается суммарный функционал.

```

> B:=vector([1,x]):
eqns:=NULL:ch:=0:

for e to n do

C:=matrix([[1,G1[e,1]],
[1,G1[e,2]] ]):
N:=multiply(B,inverse(C)):
P:=(G1[e,3],G1[e,4]):
phi:=multiply(N,P):

chi:=int((1/2)*K[e]*(diff(phi,x))^2*A,x=G1[e,1]..G1[e,2]):

if q[e]<>0 or q[e+1]<>0 then
chi:=chi+sum( 'int(q[k]*Phi[k],s=0..A)', 'k'=e..e+1):
fi:

if Tw[e]<>0 or Tw[e+1]<>0 then
chi:=chi+sum('int((alpha[k]/2)*(Phi[k]-Tw[k])^2,s=0..A)',
'k'=e..e+1):
fi:

ch:=ch+chi:

od:
ch;

```

$$\frac{1}{2}(-100 + \Phi_2)^2 \pi + \frac{1}{2}(-\Phi_2 + \Phi_3)^2 \pi + \frac{1}{2}(-\Phi_3 + \Phi_4)^2 \pi + \frac{1}{2}(-\Phi_4 + \Phi_5)^2 \pi + \frac{1}{2} \Phi_5^2 \pi$$

Далее осуществляется минимизация функционала

```

> var1:=NULL:
for e to m do
if type(Phi[e],integer) then
eq1:=0=0
else
eq1:=diff(ch,Phi[e])=0:
var1:=var1,Phi[e]:
fi:
eqns:=eqns,eq1:
od:

```

Формируется система линейных алгебраических уравнений

```
> eqns:={eqns}:var:={var1}:
```

решение которой дает неизвестные значения температуры в узлах:

```
> sols := solve( eqns,var);
```

```
sols := {  $\Phi_3 = 60$ ,  $\Phi_2 = 80$ ,  $\Phi_4 = 40$ ,  $\Phi_5 = 20$  }
```

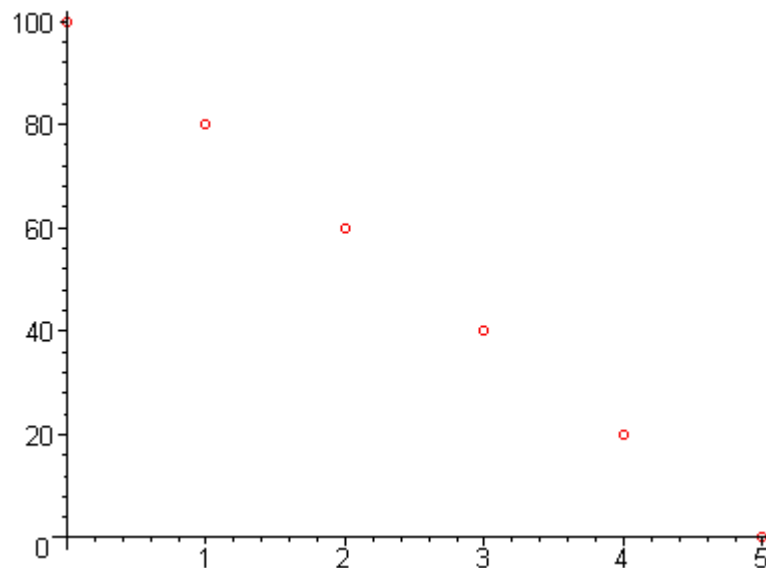
Для построения графика понадобится процедура сортировки вычисленных значений

```
> sorty:=proc(Phi,sols,m)
local i,j:
for i to nops(sols) do
for j to m do
if Phi[j]<>op(1,sols[i]) then next
else Phi[j]:=op(2,sols[i])
fi:
od:
od:
end:
sorty(Phi,sols,m):
evalm(Phi);
evalm(G1):
```

```
[ 100, 80, 60, 40, 20, 0 ]
```

А теперь непосредственно график:

```
> with(plots):
f_1:=plot([seq([X[i], Phi[i]], i = 1 .. nops(X))], style = point, symbol = circle,
color = red):display(f_1);
```



Рассмотренное решение простого примера может быть легко проверено аналитически с помощью средств Maple.

Задаем дифференциальное уравнение и граничные условия

```
> with(DEtools):
du1:=diff(T_0(x),x,x)=0;
in1:=T_0(0)=100,T_0(5)=0;
```

$$du1 := \frac{\partial^2}{\partial x^2} T_0(x) = 0$$

$$in1 := T_0(0) = 100, T_0(5) = 0$$

получаем решение

```
> qq:=dsolve({du1,in1},{T_0(x)}):
> T_a:=subs(qq,T_0(x));
```

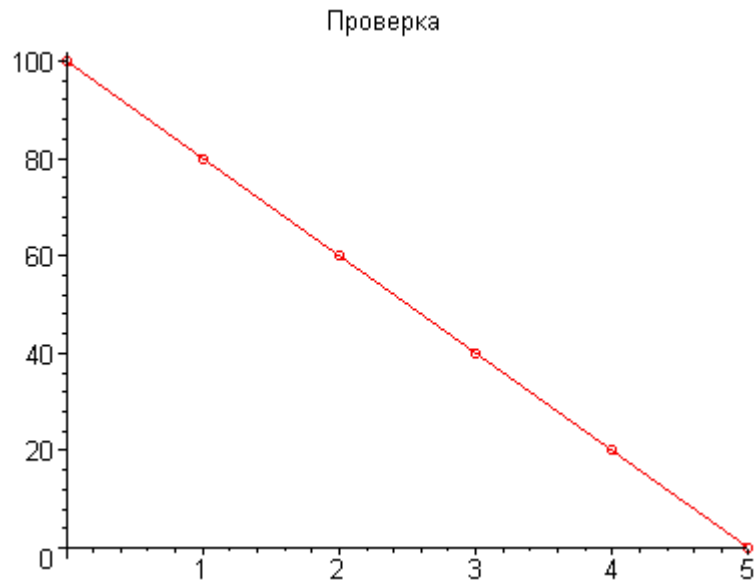
$$T_a := -20 x + 100$$

Теперь определяем параметры графика

```
> f_2:=plot(T_a,x=0..5,title='Проверка');
```

и изображаем полученные численное и аналитическое решение на одном чертеже

```
> display({f_1,f_2});
```



В данной задаче аналитическое и численное решения полностью совпадают.

Упражнение

Найдите численное решение рассмотренной задачи, если

> **Phi[1]:=100;q[6]:=0: K=[2,2,2,3,3]:**

4. ДВУМЕРНАЯ ЗАДАЧА ТЕПЛОПРОВОДНОСТИ

4.1. Постановка задачи

> *restart; with(linalg)*

Warning, the protected names norm and trace have been redefined and unprotected

Рассмотрим задачу теплопроводности в двумерной постановке.

С вариационной точки зрения отыскание минимума функционала

$$\chi = \int \frac{1 \left(K_x \left(\frac{\partial}{\partial x} T(x, y) \right)^2 + K_y \left(\frac{\partial}{\partial y} T(x, y) \right)^2 - 2 Q T(x, y) \right)}{2} dV + \int q T(x, y) + \frac{1 \alpha (T(x, y) - T_w)^2}{2} dS$$

(4.1)

эквивалентно решению дифференциального уравнения теплопроводности

$$\left(\frac{\partial}{\partial x} K_x \left(\frac{\partial}{\partial x} T(x, y) \right) \right) + \left(\frac{\partial}{\partial y} K_y \left(\frac{\partial}{\partial y} T(x, y) \right) \right) + Q = 0 \quad (4.2)$$

с граничными условиями

$$K_x \left(\frac{\partial}{\partial x} T(x, y) \right) l_x + K_y \left(\frac{\partial}{\partial y} T(x, y) \right) l_y + q + \alpha (T_{x, y} - T_w) = 0 \quad (4.3)$$

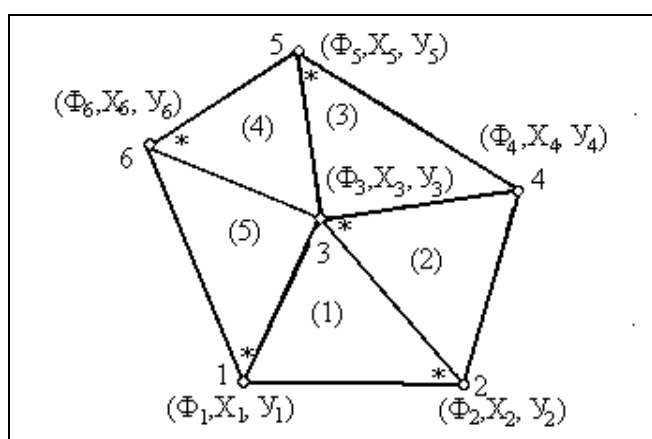
В формулах (4.1)-(4.3): T - температура, K - коэффициент теплопроводности, q - тепловой поток заданной интенсивности, Q - внутренний тепловой источник или сток, α - коэффициент конвективного теплообмена, T_w - температура окружающей среды, l - направляющие косинусы вектора нормали к поверхности,

V - объем, S - площадь поверхности, x, y - координаты.

Таким образом, любое распределение $T(x, y)$, при котором функционал χ (1) становится минимальным, является решением исходной задачи теплопроводности (2)-(3).

4.2. Описание геометрии области определения в симплекс-элементах

Пусть рассматривается область структуры, аналогичной геометрии из раздела 2 (рис.4.1).



Задаем число узловых точек

> **m:=6:**

и число элементов

> **n:=5:**

Вводим координат узловых точек

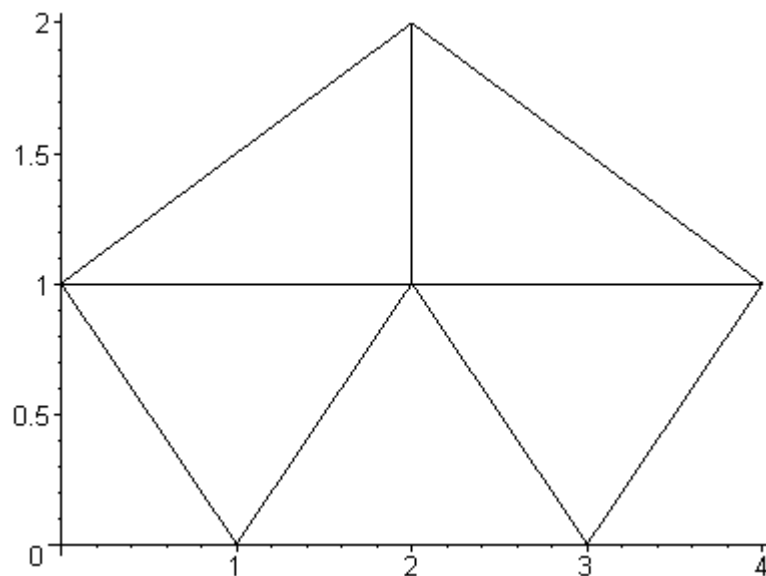
> **X:=[1,3,2,4,2,0]: Y:=[0,0,1,1,2,1]:**

Данные о структуре области могут быть представлены номерами узлов каждого из элементов, которые вводятся в обходе по часовой стрелки, начиная с узла, помеченного на рисунке звездочкой.

> **G1:=array(1..n,1..25,
[
[2,3,1],
[3,2,4],
[5,3,4],
[6,3,5],
[1,3,6]
]):**

Построим изображение расчетной сетки

> **PLOT(POLYGONS([[X_{G1_{e,1}}, Y_{G1_{e,1}}],[X_{G1_{e,2}}, Y_{G1_{e,2}}],[X_{G1_{e,3}}, Y_{G1_{e,3}}]]\$(e=1..n)))**



Вычислим площади каждого из элементов

>
for e to n do $A_e = X_{G1_{e,2}} Y_{G1_{e,3}} + X_{G1_{e,3}} Y_{G1_{e,1}} + X_{G1_{e,1}} Y_{G1_{e,2}} - X_{G1_{e,2}} Y_{G1_{e,1}} - X_{G1_{e,3}} Y_{G1_{e,2}} - X_{G1_{e,1}} Y_{G1_{e,3}}$
end do

4.3. Описание физических свойств расчетной области

Определим параметры по умолчанию

```
> Phi:=vector(m);  
for i to m do  
q[i]:=0: Q[i]:=0: alpha[i]:=1: Tw[i]:=0:  
od:
```

```
 $\Phi := \text{array}(1..6, [ \ ])$ 
```

Зададим известные узловые температуры:

```
> Phi[1]:=100.:Phi[4]:=0:
```

Зададим известные тепловые потоки в узлах расчетной сетки:

```
> Q[3]:=-10:
```

Можно задать известные температуры окружающей среды для элементов, с ней контактирующих (например, $Tw[2]=20$)

```
>
```

и соответствующие коэффициенты теплоотдачи окружающей среде (например $alpha[2]=5$)

```
>
```

Коэффициенты теплопроводности отдельных элементов могут быть отличными от принятых по умолчанию (например $K[3]=1.3$)

```
>
```

Сформируем рабочий массив данных

```
> for i to n do  
     $K_i = 1;$   
     $Gl_{i,4} = X_{Gl_{i,1}};$   
     $Gl_{i,5} = Y_{Gl_{i,1}};$   
     $Gl_{i,6} = X_{Gl_{i,2}};$   
     $Gl_{i,7} = Y_{Gl_{i,2}};$   
     $Gl_{i,8} = X_{Gl_{i,3}};$   
     $Gl_{i,9} = Y_{Gl_{i,3}};$   
od;
```

```

 $Gl_{i,10} = \Phi Gl_{i,1};$ 
 $Gl_{i,11} = \Phi Gl_{i,2};$ 
 $Gl_{i,12} = \Phi Gl_{i,3};$ 
 $Gl_{i,13} = q Gl_{i,1};$ 
 $Gl_{i,14} = q Gl_{i,2};$ 
 $Gl_{i,15} = q Gl_{i,3};$ 
 $Gl_{i,16} = \alpha Gl_{i,1};$ 
 $Gl_{i,17} = \alpha Gl_{i,2};$ 
 $Gl_{i,18} = \alpha Gl_{i,3};$ 
 $Gl_{i,19} = Tw Gl_{i,1};$ 
 $Gl_{i,20} = Tw Gl_{i,2};$ 
 $Gl_{i,21} = Tw Gl_{i,3};$ 
 $Gl_{i,22} = K_i;$ 
 $Gl_{i,23} = Q Gl_{i,1};$ 
 $Gl_{i,24} = Q Gl_{i,2};$ 
 $Gl_{i,25} = Q Gl_{i,3};$ 

```

```

end do;
evalm(Gl)

```

$$\begin{bmatrix} 2 & 3 & 1 & 3 & 0 & 2 & 1 & 1 & 0 & \Phi_2 & \Phi_3 & 100. & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & -10 & 0 \\ 3 & 2 & 4 & 2 & 1 & 3 & 0 & 4 & 1 & \Phi_3 & \Phi_2 & 0. & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & -10 & 0 & 0 \\ 5 & 3 & 4 & 2 & 2 & 2 & 1 & 4 & 1 & \Phi_5 & \Phi_3 & 0. & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & -10 & 0 \\ 6 & 3 & 5 & 0 & 1 & 2 & 1 & 2 & 2 & \Phi_6 & \Phi_3 & \Phi_5 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & -10 & 0 \\ 1 & 3 & 6 & 1 & 0 & 2 & 1 & 0 & 1 & 100. & \Phi_3 & \Phi_6 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & -10 & 0 \end{bmatrix}$$

Следующий фрагмент программного кода производит автоматическое вычисление интерполяционных полиномов и функционалов для каждого из n отдельных элементов. В итоге получается суммарный функционал, характеризующий задачу.

```

> B:=vector([1,x,y]):
eqns:=NULL:ch:=0:

```

```

for e to n do

```

```

C:=matrix([[1,G1[e,4],G1[e,5]],
[1,G1[e,6],G1[e,7]],
[1,G1[e,8],G1[e,9]] ]):
N:=multiply(B,inverse(C)):

```

```

P:=(G1[e,10],G1[e,11],G1[e,12]):
phi:=multiply(N,P):

```

```

chi:=int((1/2)*(K[e]*((diff(phi,x))^2+K[e]*(diff(phi,y))^2)*A[e]),z=0..1):

```

```

if G1[e,23]<>0 or G1[e,24]<>0 or G1[e,25]<>0 then
chi:=chi-sum( 'int((1/2)*G1[e,k]*G1[e,k-13],z=0..1)', 'k'=23..25):
fi:

```

```

if G1[e,13]<>0 or G1[e,14]<>0 or G1[e,15]<>0 then
chi:=chi+sum( 'int(G1[e,k]*G1[e,k-3],s=0..A[e])', 'k'=13..15):
fi:

```

```

if G1[e,19]<>0 or G1[e,20]<>0 or G1[e,21]<>0 then
chi:=chi+sum('int((G1[e,k-3]/2)*(G1[e,k-9]-G1[e,k])^2,s=0..A[e])',
'k'=19..21):
fi:

```

```

ch:=ch+chi:

```

```

od:
ch;

```

$$\begin{aligned}
& (.50000000000 \Phi_2 - 50.)^2 + (-.50000000000 \Phi_2 + \Phi_3 - 50.)^2 + 25 \Phi_3 + \frac{1}{2} \Phi_3^2 + \left(\frac{1}{2} \Phi_3 - \Phi_2 \right)^2 + 2 (\Phi_5 - \Phi_3)^2 \\
& + \left(-\frac{1}{2} \Phi_6 + \frac{1}{2} \Phi_3 \right)^2 + (-.50000000000 \Phi_6 + .50000000000 \Phi_3)^2 \\
& + (-100. + .50000000000 \Phi_3 + .50000000000 \Phi_6)^2
\end{aligned}$$

Осуществляем минимизацию функционала

```

> varl := NULL;
for i to m do
  if type(Φi, integer) or type(Φi, float) then eql := 0 = 0 else eql :=  $\frac{\partial}{\partial \Phi_i} ch = 0$ ; varl := varl, Φi end if;
  eqns := eqns, eql
end do

```

Формируем итоговую систему линейных алгебраических уравнений (СЛАУ)

```
> eqns := {eqns}; var := {varl}
```

```
eqns := {-2.000000000 Φ2 + 9.000000000 Φ3 - 175.0000000 - 4 Φ5 - .5000000000 Φ6 = 0, 4 Φ5 - 4 Φ3 = 0,
1.500000000 Φ6 - .5000000000 Φ3 - 100.0000000 = 0, 0 = 0, 3.000000000 Φ2 - 2.000000000 Φ3 = 0}
```

и решаем ее

```
> sols := solve(eqns, var)
```

```
sols := { Φ6 = 86.50793651, Φ5 = 59.52380952, Φ2 = 39.68253968, Φ3 = 59.52380952 }
```

Таким образом получены неизвестные значения температуры в соответствующих узлах:

Процедура сортировки и вывод итогового вектора узловых значений

```
> sorty := proc(Φ, sols, m)
    local i, j;
    for i to nops(sols) do
        for j to m do if Φ[j] ≠ op(1, sols[i]) then next else Φ[j] := op(2, sols[i]) end if end do
        end do
    end proc;
    sorty(Φ, sols, m);
    evalm(Φ);
    evalm(Gl)
```

```
[ 100., 39.68253968, 59.52380952, 0., 59.52380952, 86.50793651 ]
```

Проверим невязки

```
> op(eqns)
```

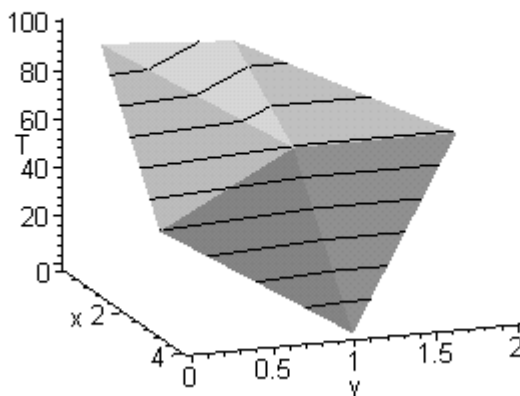
```
- .6 10-7 = 0, 0 = 0, 0. = 0
```

Построим график распределения температуры

```
> with(plots):
list_poy := [seq([
[G1[S,4],G1[S,5],G1[S,10]],
[G1[S,6],G1[S,7],G1[S,11]],
[G1[S,8],G1[S,9],G1[S,12]]
],S=1..n)];
polygonplot3d(list_poy, scaling=UNCONSTRAINED, axes=FRAMED,
```

**titlefont=[TIMES,ROMAN,12], shading=ZGREYSCALE,labels=[x,y,T],
style=PATCHCONTOUR, orientation=[-20,70]);**

Warning, the name changecoords has been redefined



Упражнение

Для некоторых достаточно тонких элементов конструкций изменением температуры по толщине можно пренебречь. Однако для таких элементов значительное влияние оказывает теплообмен с окружающей средой вдоль лицевых поверхностей.

Найдите распределение температуры для геометрической области рассматриваемой в данном разделе, если происходит подобная теплоотдача при температуре окружающей среды $T_w=20$, а коэффициент конвективного теплообмена $\alpha=20$. Коэффициент теплопроводности всей области в целом считать равным $K=15$.

5. СРАВНЕНИЕ С АНАЛИТИЧЕСКИМ РЕШЕНИЕМ

В добавление к обычным ошибкам округления и аппроксимации, свойственным почти любой вычислительной процедуре в методе конечных элементов имеют место:

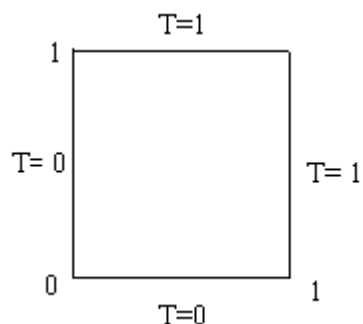
- а) ошибки, являющиеся результатом геометрических различий границы области определения и ее конечно-элементным приближением;
- б) ошибки, обусловленные разницей между точным решением задачи в частных производных и его представлением пробной функцией.

```
> restart:
with(linalg):
with(plots):
FIG:= array(1..2):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
Warning, the name changecoords has been redefined
```

Рассмотрим решение уравнения Лапласа (уравнение 4.2 с нулевой правой частью) на единичном квадрате с граничными условиями, указанными на рисунке.



При решении задачи методом конечных элементов используем процедуры, описанные в п.4.

Принятое число узловых точек - 25, число элементов -32.

```
> m:=25:
```

```
> n:=32:
```

Координаты узловых точек

```
> X:=[0,0.25,0.5,0.75,1,
0,0.25,0.5,0.75,1,
0,0.25,0.5,0.75,1,
0,0.25,0.5,0.75,1,
0,0.25,0.5,0.75,1]:
Y:=[0,0,0,0,0,
```

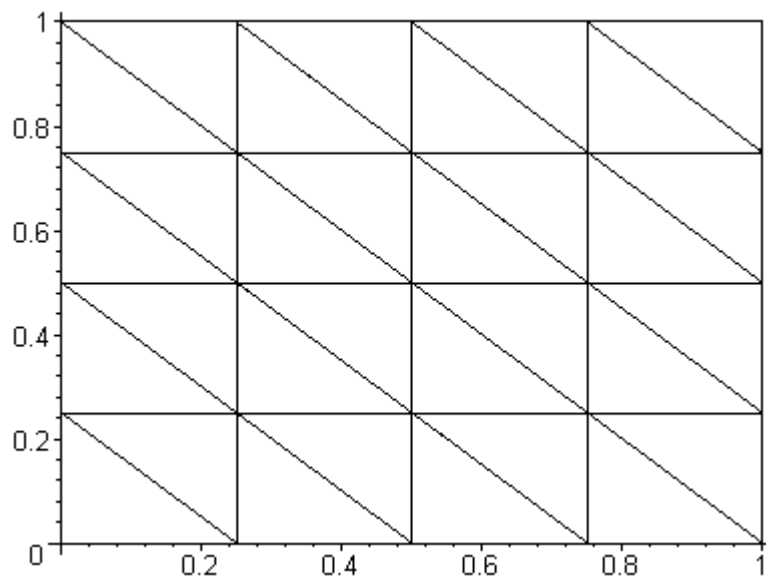
```
0.25,0.25,0.25,0.25,0.25,
0.5,0.5,0.5,0.5,0.5,
0.75,0.75,0.75,0.75,0.75,
1,1,1,1,1]:
```

Структура области.

```
> G1:=array(1..n,1..25,
[
[1,2,6], [2,7,6], [2,3,7],[3,8,7],[3,4,8],[4,9,8],[4,5,9],[5,10,9],
[6,7,11],[7,12,11],[7,8,12],[8,13,12],[8,9,13],[9,14,13],[9,10,14],
[10,15,14],[11,12,16],[12,17,16],[12,13,17],[13,18,17],[13,14,18],
[14,19,18],[14,15,19],[15,20,19],[16,17,21],[17,22,21],[17,18,22],
[18,23,22],[18,19,23],[19,24,23],[19,20,24],[20,25,24]
]):
```

Конечно-элементная расчетная сетка

```
> PLOT(POLYGONS([X[G1[e,1]],Y[G1[e,1]]], [X[G1[e,2]],Y[G1[e,2]]],
[X[G1[e,3]],Y[G1[e,3]]] ] $e=1..n));
```



Параметры по умолчанию

```
> Phi:=vector(m):
for i to m do
q[i]:=0: Q[i]:=0: alpha[i]:=1: Tw[i]:=0:
od:
```

Ввод известных значений функции в точках на границе

```
> Phi[1]:=0.:Phi[2]:=0.:Phi[3]:=0.:Phi[4]:=0.:Phi[5]:=0.5:
Phi[6]:=0.:Phi[15]:=1.:Phi[11]:=0.:Phi[20]:=1.:
```



```
Phi[16]:=0.:Phi[10]:=1.:
Phi[21]:=0.5:Phi[22]:=1:Phi[23]:=1:Phi[24]:=1:Phi[25]:=1.:
```

Расчетный блок конечно-элементного метода

```
> for e to n do
A[e]:=(X[G1[e,2]]*Y[G1[e,3]]+X[G1[e,3]]*Y[G1[e,1]]+X[G1[e,1]]*Y[G1[e,2]]-
X[G1[e,2]]*Y[G1[e,1]]-X[G1[e,3]]*Y[G1[e,2]]-X[G1[e,1]]*Y[G1[e,3]]):

od:

for i to n do
K[i]:=1:
G1[i,4]:=X[G1[i,1]]: G1[i,5]:=Y[G1[i,1]]:
G1[i,6]:=X[G1[i,2]]: G1[i,7]:=Y[G1[i,2]]:
G1[i,8]:=X[G1[i,3]]: G1[i,9]:=Y[G1[i,3]]:
G1[i,10]:=Phi[G1[i,1]]:G1[i,11]:=Phi[G1[i,2]]:G1[i,12]:=Phi[G1[i,3]]:
G1[i,13]:=q[G1[i,1]]:G1[i,14]:=q[G1[i,2]]:G1[i,15]:=q[G1[i,3]]:
G1[i,16]:=alpha[G1[i,1]]:G1[i,17]:=alpha[G1[i,2]]:
G1[i,18]:=alpha[G1[i,3]]:
G1[i,19]:=Tw[G1[i,1]]:G1[i,20]:=Tw[G1[i,2]]:G1[i,21]:=Tw[G1[i,3]]:
G1[i,22]:=K[i]:
G1[i,23]:=Q[G1[i,1]]:G1[i,24]:=Q[G1[i,2]]:G1[i,25]:=Q[G1[i,3]]:
od:
evalm(G1):

B:=vector([1,x,y]):
eqns:=NULL:ch:=0:

for e to n do

C:=matrix([[1,G1[e,4],G1[e,5]],
[1,G1[e,6],G1[e,7]],
[1,G1[e,8],G1[e,9]] ]):
N:=multiply(B,inverse(C)):

P:=(G1[e,10],G1[e,11],G1[e,12]):
phi:=multiply(N,P):

chi:=int((1/2)*(K[e]*((diff(phi,x))^2+K[e]*(diff(phi,y))^2)*A[e]),z=0..1):

if G1[e,23]<>0 or G1[e,24]<>0 or G1[e,25]<>0 then
chi:=chi-A[e]*sum( 'int((1/2)*G1[e,k]*G1[e,k-13],z=0..1)', 'k'=23..25):
fi:
```

```

if G1[e,13]<>0 or G1[e,14]<>0 or G1[e,15]<> 0 then
chi:=chi+sum( 'int(G1[e,k]*G1[e,k-3],s=0..A[e])', 'k'=13..15):
fi:

if G1[e,19]<>0 or G1[e,20]<>0 or G1[e,21]<>0 then
chi:=chi+sum('int(((G1[e,k-3]/2)*(G1[e,k-9]-G1[e,k])^2,s=0..A[e])',
'k'=19..21):
fi:

ch:=ch+chi:

od:
ch:

var1:=NULL:
for i to m do
if type(Phi[i],integer) or type(Phi[i],float) then
eq1:=0=0
else
eq1:=diff(ch,Phi[i])=0:
var1:=var1,Phi[i]:
fi:
eqns:=eqns,eq1:
od:

eqns:={eqns}:var:={var1}:
sols := solve( eqns,var):

sorty:=proc(Phi,sols,m)
local i,j:
for i to nops(sols) do
for j to m do
if Phi[j]<>op(1,sols[i]) then next
else Phi[j]:=op(2,sols[i])
fi:
od:
od:
end:
sorty(Phi,sols,m):
evalm(Phi):
evalm(G1):

```

Матрица найденных значений для неизвестных узлов

```
> T_num:=matrix([[Phi[7],Phi[8],Phi[9]],
[Phi[12],Phi[13],Phi[14]],
[Phi[17],Phi[18],Phi[19]]]);
```

$$T_{num} = \begin{bmatrix} .1428571429 & .2857142857 & .5000000000 \\ .2857142857 & .5000000000 & .7142857143 \\ .5000000000 & .7142857143 & .8571428571 \end{bmatrix}$$

Описание графического представления матрицы решений

```
>
FIG[1]:=matrixplot(T_num,axes=FRAMED,style=PATCHCONTOUR,title='
МКЭ',labels=[x,y,T], orientation=[-10,90]):
```

Аналитическое решение выполним по методу Фурье, который включает в себе разделение переменных и применение принципа суперпозиции. При этом разыскивается так называемое формальное решение, т.е. решение в виде бесконечного ряда, в котором каждый член есть решение уравнения Лапласа и удовлетворяет поставленным граничным условиям.

Число внутренних узлов по каждой из осей координат

```
> Nn:=3:
```

Число членов ряда, используемых для построения решения

```
> Ns:=40:
```

Блок вычислений по методу Фурье

```
> tt:=1/(Nn+1):p:=1: q:=1:
a:=array(1..Ns):b:=array(1..Ns):
for n from 1 to Ns do
a[n]:=sqrt(2/p)*int(0*sin(Pi*n*x),x=0..1):
b[n]:=sqrt(2/p)*int(1*sin(Pi*n*x),x=0..1):
od:

u:=array(1..Nn,1..Nn):
for i from 1 to Nn do
for j from 1 to Nn do
trew:=sum('sin(Pi*k*i*tt)/sinh(Pi*k*q)*(a[k]*sinh(Pi*k*(1-
j*tt))+b[k]*sinh(Pi*k*j*tt))', 'k=1..Ns'):
u[i,j]:=sqrt(2)*trew:
od:
od:
```

```

for n from 1 to Ns do
a[n]:=sqrt(2/p)*int(0*sin(Pi*n*y),y=0..1):
b[n]:=sqrt(2/p)*int(1*sin(Pi*n*y),y=0..1):
od:

u1:=array(1..Nn,1..Nn):
for i from 1 to Nn do
for j from 1 to Nn do
trew:=sum('sin(Pi*k*j*tt)/sinh(Pi*k*q)*(a[k]*sinh(Pi*k*(1-
i*tt))+b[k]*sinh(Pi*k*i*tt))', 'k=1..Ns'):
u1[i,j]:=sqrt(2)*trew:
od:od:

```

Матрица значений, найденных по методу Фурье, в точках области, соответствующих конечно-элементному разбиению

```
> T_anal:=evalf(evalm(u+u1));
```

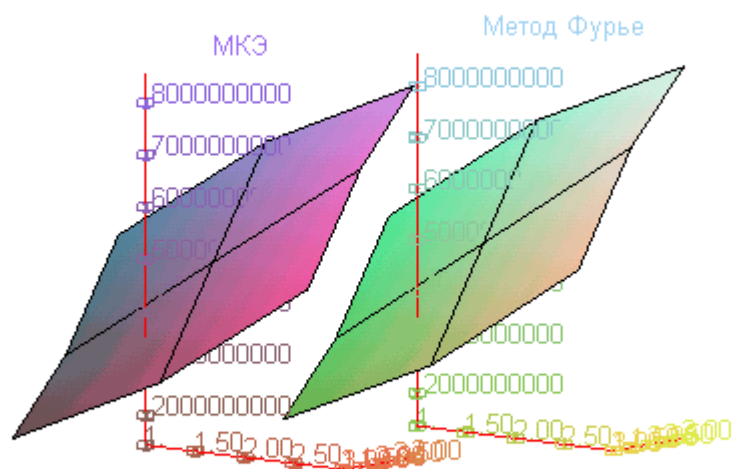
$$T_{anal} := \begin{bmatrix} .1359433362 & .2774424497 & .4999999995 \\ .2774424497 & .4999999998 & .7225575495 \\ .4999999995 & .7225575495 & .8640566628 \end{bmatrix}$$

Описание графического представления матрицы решений по методу Фурье

```
>
FIG[2]:=matrixplot(T_anal,axes=FRAMED,style=PATCHCONTOUR,title='
Метод Фурье',labels=[x,y,T], orientation=[-10,90]):
```

Общий характер решения по двум методам

```
> display(FIG);
```



Матрица относительных ошибок для найденных значений непрерывной функции в узлах разбиения (проценты)

```
> fd:=evalm(abs(T_num-T_anal)):mer:=matrix(3,3):
for i to 3 do
for j to 3 do
mer[i,j]:=100*fd[i,j]/T_anal[i,j]
od:
od:
evalm(mer);
```

$$\begin{bmatrix} 5.085800373 & 2.981460122 & .1000000001 \cdot 10^{-6} \\ 2.981460122 & .4000000002 \cdot 10^{-7} & 1.144799498 \\ .1000000001 \cdot 10^{-6} & 1.144799498 & .8001565172 \end{bmatrix}$$

>

Таким образом, максимальная относительная погрешность алгоритма метода конечных элементов в сравнении с аналитическим решением по методу Фурье не превысила 5.1 %.

При измельчении расчетной сетки и приближении треугольного конечного элемента по форме к равностороннему, численное решение, как правило, стремиться к уточнению.

Упражнение

Проверьте последнее утверждение

6. АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ КОНЕЧНО-ЭЛЕМЕНТНОЙ СЕТКИ И

ПРИМЕР РАСЧЕТА ПОЛЯ ТЕМПЕРАТУР В ОБЛАСТИ СО СЛОЖНОЙ ГЕОМЕТРИЕЙ

> **restart:**

Процедуры автоматического получения расчетной сетки могут быть основаны на различных геометрических соображениях и принципах, как правило, не имея при этом глубокого теоретического обоснования. Главная их цель - выработать структуру исходных данных для последующего расчета.

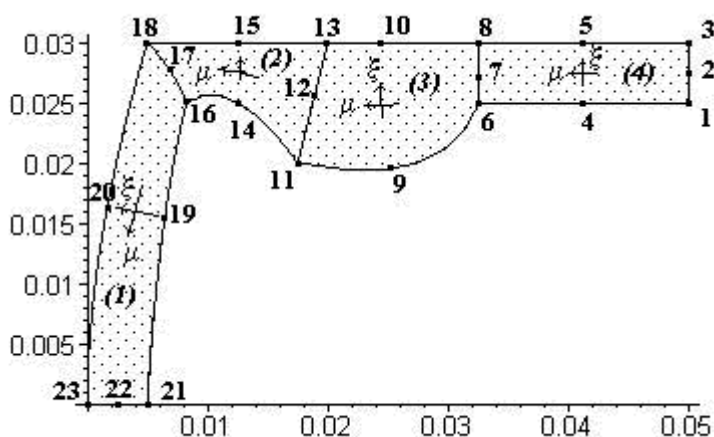
Все процедуры, задействованные в этом рабочем документе, объединены в пакет `h_FEM` и считываются с жесткого диска в следующей командной ячейке:

> **read"d:/МКЭ/h_FEM.m":**

Загрузим пакеты, необходимые в дальнейшем

> **with(plots):with(h_FEM):**

В рассматриваемом примере исходная область определения разбивается предварительно (в ручную) на четыре четырехугольных зоны (подобласти), имеющие общие стороны, как показано на следующем рисунке.



Нумерация узлов, используемых для задания четырехугольных зон, также как и самих зон произвольна.

Число зон задает переменная INRG, а число граничных точек - переменная INBP.

> **INRG:=4: INBP:=23:**

Координаты граничных точек в порядке их нумерации заносятся в массив ХУР

> **XYP:=array(1..INBP,1..2, [
[50,25],[50,27.5],[50,30],[41.25,25],[41.25,30],
[32.5,25],[32.5,27.5],[32.5,30],[27.5,20],[26.25,30],
[17.5,20],[18.75,25],[20,30],[12.5,25],[12.5,30],
[8.25,25],[7.,27.5],[5,30],[6.25,14],[1.25,14],
[5,0],[2.5,0],[0,0]
]):**

В каждой подобласти, для того чтобы была возможность приписывать узлам, расположенным вдоль общей границы одни и те же номера, введена местная система координат $\xi \ \mu$, ориентация которой влияет на структуру матрицы итоговой системы линейных алгебраических уравнений (ширину ее полосы).

Чем меньшая ширина полосы будет получена, тем эффективнее будет последующий процесс вычислений.

Данные о соединении зон вводятся в массив JT и состоят из чисел, представляющих собой номера каждой из сторон отдельной зоны. При этом стороны нумеруются следующим образом: $(-\mu)$ - первая сторона, (ξ) - вторая, (μ) - третья, $(-\xi)$ - четвертая.

Первая строка массива JT (зона 1) показывает, что сторона 1 граничит с зоной 2.

Вторая строка (зона 2) - сторона 1 граничит с зоной 3, а сторона 3 граничит с зоной 1.

Третья строка (зона 3) - сторона 1 граничит с зоной 4, а сторона 3 граничит с зоной 2.

Четвертая строка (зона 4) - сторона 3 граничит с зоной 3.

> **JT:=array(1..INRG,1..4,[
[2,0,0,0],
[3,0,1,0],
[4,0,2,0],
[0,0,3,0]]):**

Данные зон вводятся в массив region. Структура данных в строке (на примере первой строки массива):

1 - номер зоны, 14 - число строк узлов в зоне, 5 - число столбцов узлов в зоне;

16,17,18,20,23,22,21,19 - номера узлов зоны в обходе против часовой стрелки,

т.е. начиная с узла с координатами $-\frac{\mu}{k} + \frac{\xi}{k}$, $-\frac{\xi}{k}$ и двигаясь слева направо от $-\frac{\xi}{k}$ и сверху вниз от $+\frac{\mu}{k} - \frac{\mu}{k}$.

```
> region:=array(1..INRG,1..11,[  
[1, 14,5, 16,17,18,20,23,22,21,19],  
[2, 7, 5, 11,12,13,15,18,17,16,14],  
[3, 7, 5, 6, 7, 8, 10,13,12,11,9 ],  
[4, 9, 5, 1, 2, 3, 5, 8, 7, 6, 4 ]]):
```

Начальные исходные данные передаются в процедуру сеточного разбиения grid,

которая возвращает результаты дискретизации

```
> grid():
```

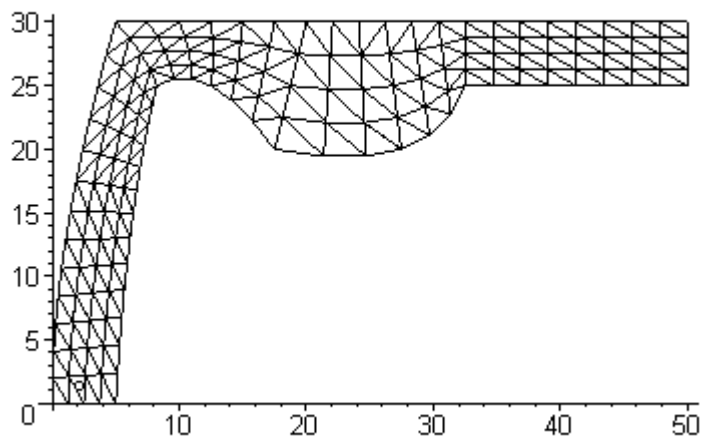
```
`Ширина полосы матрицы системы уравнений`, 7  
`Общее число элементов`, 264  
`Общее число узлов`, 170
```

Сеточное разбиение можно визуализировать. При этом некоторый выбранный конечный элемент можно пометить на сетке кружком, для этого в качестве параметра команды v_rem() вводится номер этого элемента.

```
> v_rem(5):v_gr();
```

```
`Номер помеченного элемента`, 5  
`Номера узлов помеченного элемента и их координаты:`  
`Первый узел №`, 8, [2.615384615, 2.186390533]  
`Второй узел №`, 9, [1.329696745, 2.105029586]  
`Третий узел №`, 3, [2.50, 0.]  
`Сторона 1 элемента образована узлами`, 8, `и`, 9  
`Сторона 2 элемента образована узлами`, 9, `и`, 3  
`Сторона 3 элемента образована узлами`, 3, `и`, 8
```


Сетка конечных элементов



Далее исходные данные модифицируются для расчета температурного поля, т.е. задаются граничные условия и физические характеристики области. Та часть границы, на которой условия теплообмена не будут заданы, по умолчанию теплоизолирована.

Массив `conv` заполняется для элементов, стороны которых участвуют в теплообмене с окружающей средой.

Число таких сторон определяет переменная `nr`

> **nr:=4;**

nr := 4

Структура двумерного массива `conv` следующая: `conv` [номер элемента, номер стороны].

В конвективном теплообмене может участвовать не более двух сторон одного элемента.

Если теплообмена со средой не происходит, все соответствующие данные остаются равными нулю.

> **conv:=array(1..nr,1..2,[[2,2],[2,3],[6,3],[8,2]]);**

conv :=
$$\begin{bmatrix} 2 & 2 \\ 2 & 3 \\ 6 & 3 \\ 8 & 2 \end{bmatrix}$$

Переменным H и $TINF$ присваиваются значения соответственно коэффициента конвективного теплообмена [Вт/(м² К)] и температуры

[градусы Цельсия] окружающей среды в точках поверхности конвективного теплообмена

> **H:=5: TINF:=-5:**

Переменным KXX и KYY присваиваются значения коэффициентов теплопроводности исходного материала соответственно вдоль координатных осей X и Y [Вт/м К].

> **KXX:=10: KYY:=10:**

Число узлов вычислительной сетки, в которых задана мощность теплового потока (UZq)

и известно значение температуры (UZt)

> **UZq:=1: UZt:=0:**

Номера узлов, в которых задана мощность теплового потока, заносятся в массив NUMq

> **NUMq:=array(1..UZq,[21]);**

NUMq := [21]

Заданное значение мощности в узлах [Вт] заносятся в массив QQ

> **QQ:=array(1..UZq,[100]);**

QQ := [100]

Номера узлов, в которых известно значение температуры заносятся в массив NUMt

> **NUMt:=array(1..UZt,[]);**

NUMt := array(1 .. 0, [])

Значения температуры [градусы Цельсия] в соответствующих узлах заносятся в массив TT

> **TT:=array(1..UZt,[]);**

TT := array(1 .. 0, [])

Данные перелаются в вычисляющую процедуру

> **tdheat():**

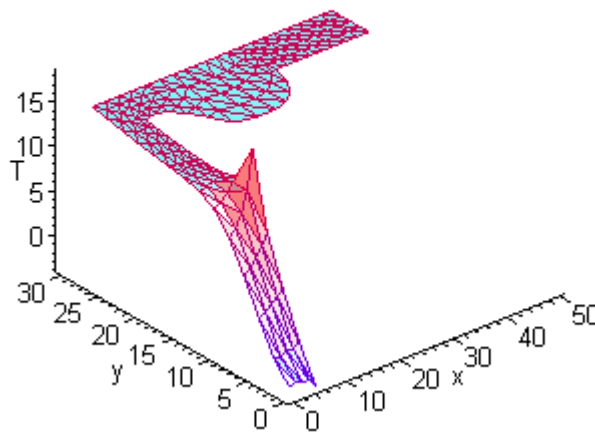
Теперь из массива X_d можно вывести значение температуры в i -ом узле сетки, например:

```
>  $X_d[15]$ ;
```

2.950875445

Построение графика

```
> polygonplot3d(list_poy, scaling=UNCONSTRAINED, axes=FRAMED,  
titlefont=[TIMES,ROMAN,12], shading=Z,labels=[x,y,T], style=hidden,  
orientation=[-130,50],lightmodel=light3);
```



Упражнение

Воспользуйтесь пакетом процедур h_FEM для вычисления распределения температуры для области произвольной формы.

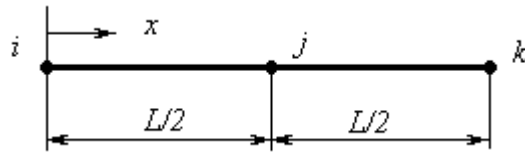
>

7. ЭЛЕМЕНТЫ ВЫСОКОГО ПОРЯДКА

Применение элементов высокого порядка, как правило, приводит к достижению заданной степени точности решения при меньшем количестве исходных данных. Однако оно не всегда ведет к сокращению полного времени счета; т.к. для составления матриц элемента необходимо использовать методы численного интегрирования, которые требуют выполнения большого числа арифметических операций.

Рассмотрим общую форму одномерного аппроксимирующего полинома

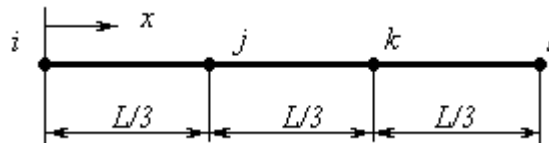
Квадратичному элементу (с тремя узлами)



соответствует полином

$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 x^2$$

а кубичному



$$\phi = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$$

Коэффициенты полиномов можно определить аналогично линейному случаю с помощью заданных значений искомой функции в узлах, решая матричное

уравнение $\alpha^C^{(-1)} \Phi =$

> **restart: with(linalg):**

Warning, the protected names norm and trace have been redefined and unprotected

> **P:=(Phi[1],Phi[2],Phi[3]):**

> **C:=matrix([[1,X[1],X[1]^2],
[1,X[2],X[2]^2],
[1,X[3],X[3]^2]
]):**

> **B:=(1,x,x^2):**

Функции формы примут вид

> **N:=multiply(B,inverse(C));**

$$N = \left[\frac{X_2 X_3}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1} - \frac{x (X_2 + X_3)}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1} + \frac{x^2}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1}, \right. \\ \left. - \frac{X_3 X_1}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3} + \frac{x (X_1 + X_3)}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3} - \frac{x^2}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3}, \right. \\ \left. \frac{X_2 X_1}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} - \frac{x (X_1 + X_2)}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} + \frac{x^2}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} \right]$$

Квадратичный интерполяционный полином будет равен

> **phi:=multiply(N,P);**

$$\phi = \left(\frac{X_2 X_3}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1} - \frac{x (X_2 + X_3)}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1} + \frac{x^2}{X_2 X_3 - X_2 X_1 + X_1^2 - X_3 X_1} \right) \Phi_1 \\ + \left(- \frac{X_3 X_1}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3} + \frac{x (X_1 + X_3)}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3} - \frac{x^2}{-X_3 X_1 + X_2 X_1 - X_2^2 + X_2 X_3} \right) \Phi_2 \\ + \left(\frac{X_2 X_1}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} - \frac{x (X_1 + X_2)}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} + \frac{x^2}{-X_3 X_1 + X_2 X_1 + X_3^2 - X_2 X_3} \right) \Phi_3$$

Пусть, например, даны:

координаты трех узлов

> **X:=[0,1,2];**

и значения функции в них

> **Phi:=[40,25.,20];**

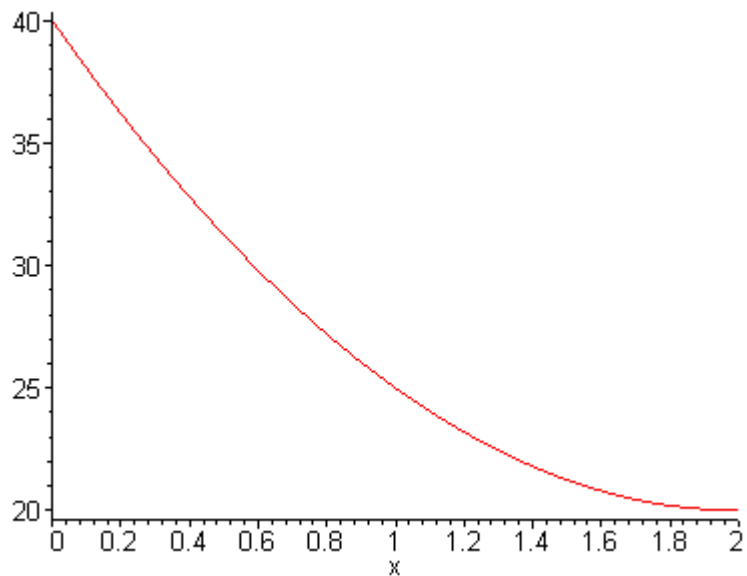
Тогда искомый интерполяционный полином:

> **phi;**

$$40. - 20. x + 5. x^2$$

Изобразим его на графике.

> **plot(phi,x=0..2);**



В точке с координатой $x = 0.5$ имеем значение искомой функции

> **subs(x=0.5,phi);**

31.25

Показанный способ получения интерполяционных полиномов нетрудно распространить на другие элементы различных типов и размерностей.

Использование при решении задач элементов высокого порядка аналогично использованию симплекс-элементов, поскольку выбор интерполяционного полинома не связан с исходными дифференциальными уравнениями.

Упражнение 1

Используйте элементы высокого порядка при решении одномерной задачи теплопроводности (п.3).

Упражнение 2

Получите квадратичный интерполяционный полином для конечного элемента, являющегося треугольником второго порядка.

Задание. Численное решение краевых задач и задач на собственные значения при помощи среды PDE Toolbox при пакете MatLab 6.5.

Введение. Среда PDE Toolbox при пакете MatLab 6.5 предназначена для численного решения задач математической физики при помощи метода конечных элементов. Эта среда позволяет решать все типы задач математической физики:

1. Эллиптические уравнения вида

$$-\nabla(c \nabla u) + a u = f,$$

где a , b и f – произвольные функции, в произвольной двумерной области Ω , на границе которой можно ставить

❖ условие Дирихле вида $h u = r$ (здесь h и r – произвольные функции)

❖ обобщенное условие Неймана $(n, c \nabla u) + q u = g$ (здесь n – нормаль к границе Ω , а q и r – произвольные функции)

2. Параболические уравнения вида

$$d u_t - \nabla(c \nabla u) + a u = f,$$

где a , b , d и f – произвольные функции, в произвольной области $\Omega \times [0, T]$ с граничными условиями Дирихле или Неймана и начальным условием

$$u = u_0(x) \quad \text{при } t = 0.$$

3. Гиперболические уравнения вида

$$d u_{tt} - \nabla(c \nabla u) + a u = f,$$

где a , b , d и f – произвольные функции, в произвольной области $\Omega \times [0, T]$ с граничными условиями Дирихле или Неймана и начальным условием

$$u = u_0(x), \quad u_t = u_1(x) \quad \text{при } t = 0.$$

4. Задачи на собственные значения вида

$$-\nabla(c \nabla u) + a u = \lambda d u,$$

где a , b , d – произвольные функции, в произвольной двумерной области с граничными условиями Дирихле или Неймана.

Графический интерфейс среды PDE Toolbox позволяет задавать двумерную область Ω путем ее рисования в редакторе, подобном Paintbrush, а функции a , b , ... - аналитическими формулами. При этом предусмотрена возможность задания этих функций различными формулами в различных подобластях Ω .

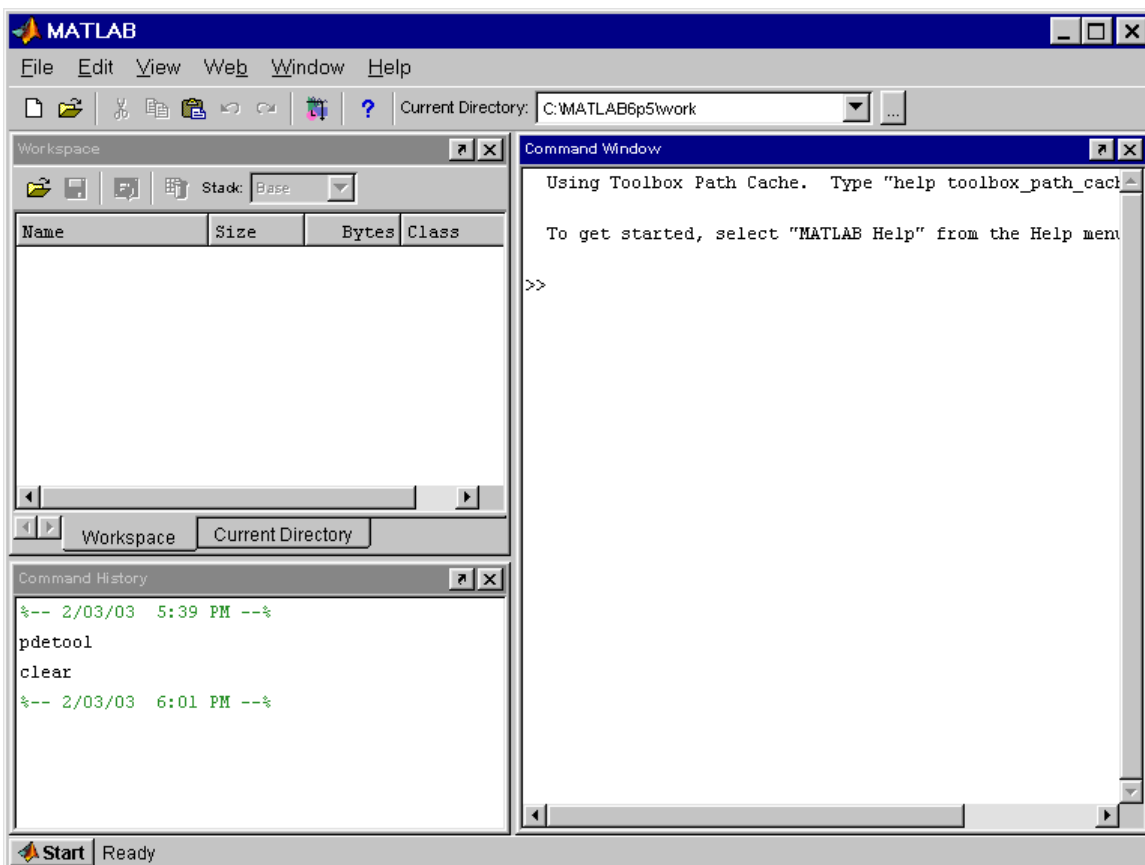
В настоящем первом задании практикума предложено по аналогии с разобранным примером решить численно одну из задач, приведенных в «Задачах по математической физики» А.Н. Боголюбова и В.В. Кравцова. Полный перечень назначения элементов управления PDE Toolbox и описание реализуемых численных методов содержится в документации [1], поставляемой вместе с MatLab (обычно, файл pde.pdf). Описание работы PDE Toolbox на русском языке имеется в книге И.Е. Ануфриева «Самоучитель MatLab 5.3-6.x» [2]. Однако, как видно из рассмотренного примера, назначение подавляющего большинства из них очевидна.

Тестовый пример. Для того, чтобы освоиться с работой в среде PDE Toolbox при пакете MatLab 6.5, рассмотрим простой пример эллиптической граничной задачи:

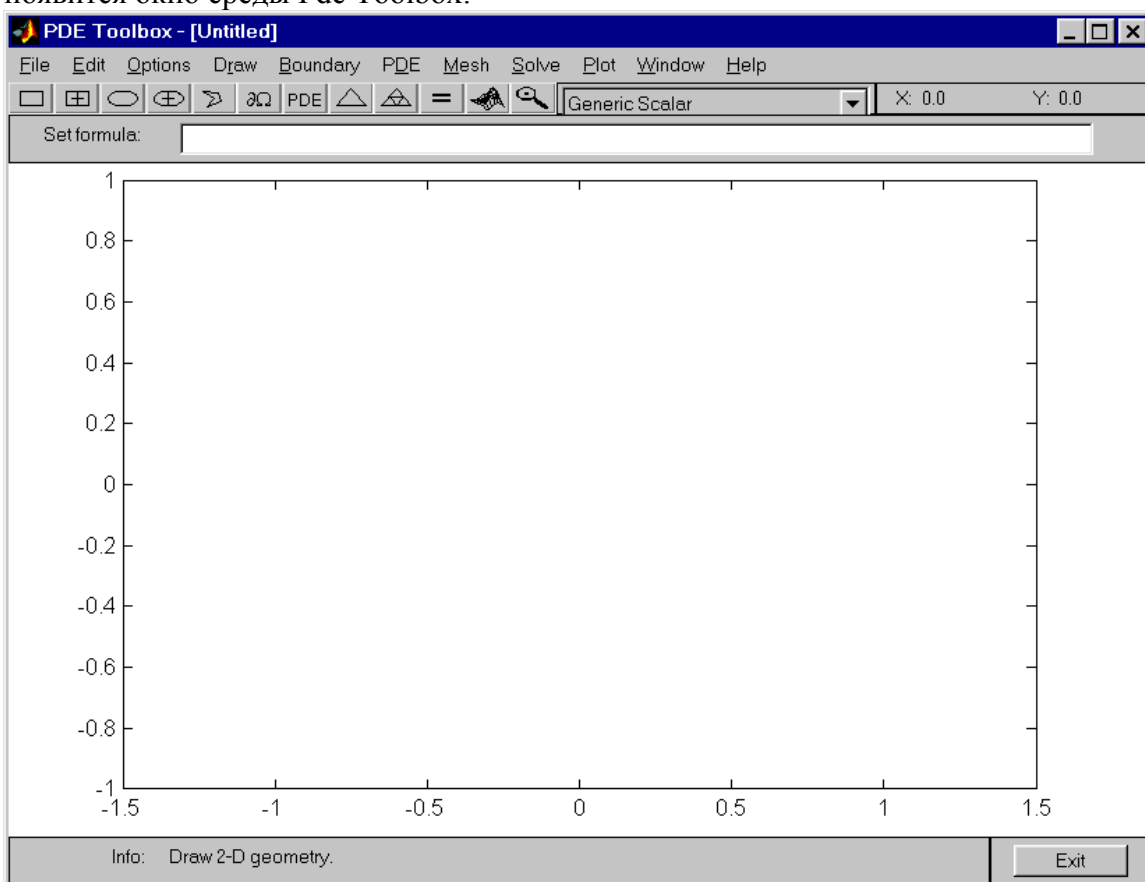
$$\begin{cases} \Delta u - 16(x^2 + y^2) = 0, \\ u|_{\partial\Omega} = 1 \end{cases}$$

Пусть рассматриваемая область Ω представляет собой круг единичного радиуса с центром в начале координат.

Для численного решения этой задачи вызовите пакет MatLab 6.5 из меню Пуск Windows. Перед вами появится главное окно пакета MatLab 6.5:



Наберите в командной строке, отмеченной символом `>>`, слово `pdetool`. После этого появится окно среды Pde Toolbox:



Ввод условий задачи осуществляется в три этапа:

1. **Задание двумерной области Ω** , в которой будет решаться краевая задача осуществляется примерно также, как в любом графическом редакторе. В данном

случае нажмите кнопку, на которой нарисован эллипс с плюсом по середине, и при помощи мыши нарисуйте эллипс примерно похожий на наш единичный круг. Затем дважды щелкните по нему мышью, тогда появится диалоговое окно с параметрами эллипса:

Object type:	Ellipse
X-center:	-0.013740458015267354
Y-center:	-0.096183206106870367
A-semiaxes:	0.86106870229007626
B-semiaxes:	0.89312977099236668
Rotation (degrees):	0
Name:	E1

Исправьте их так, чтобы получился наш круг.

2. Задание граничного условия. Нажмите кнопку $\partial\Omega$, тогда граница круга выделится красным. Это означает, что на границе заданы условия Дирихле $u = 0$ (такой выбор сделан по умолчанию). Для их смены на наше условие $u = 1$, зайдите в меню Boundary и выберите пункт Specify Boundary Condition. Перед вами появится окно:

Boundary condition equation: $h \cdot u = r$			
Condition type:	Coefficient	Value	Description
☐ Neumann	g	0	
☒ Dirichlet	q	0	
	h	1	
	r	0	

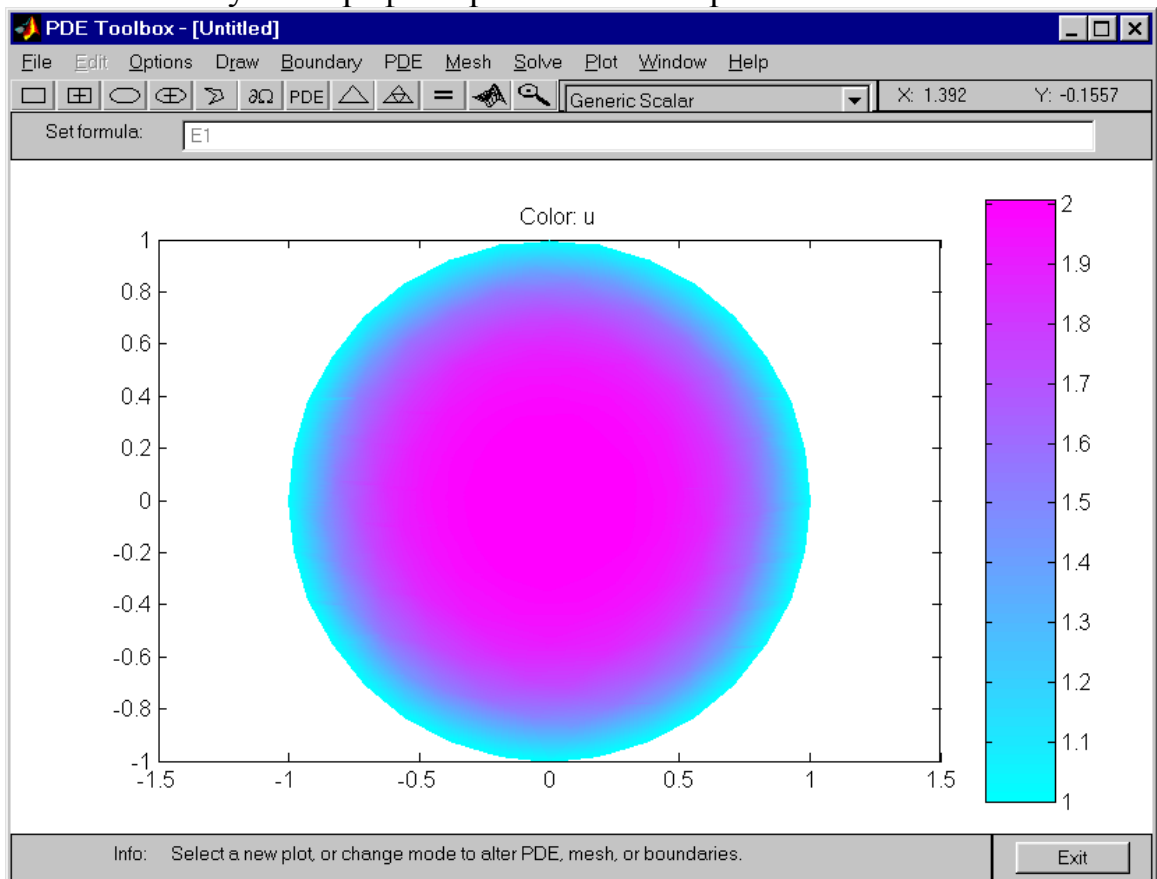
Исправьте в строке r значение 0 на 1.

3. Задание уравнения. Зайдите в меню PDE и выберите пункт Specify PDE. Перед вами появится окно:

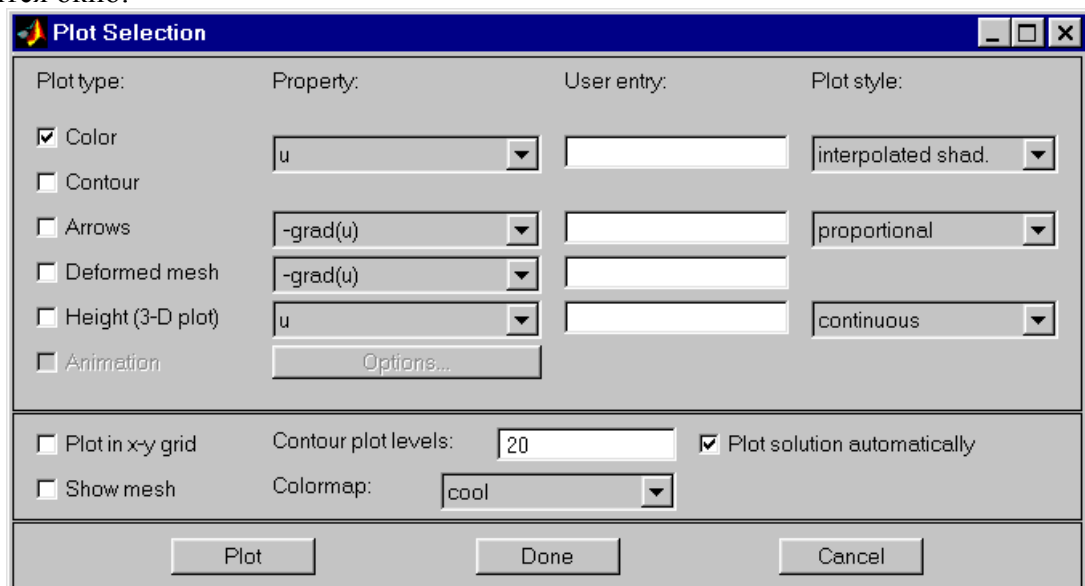
Equation: $-\text{div}(c \cdot \text{grad}(u)) + a \cdot u = f$		
Type of PDE:	Coefficient	Value
☒ Elliptic	c	1.0
☐ Parabolic	a	0.0
☐ Hyperbolic	f	10.0
☐ Eigenmodes	d	1.0

Вы видите, что по умолчанию решается уравнение $\Delta u - 10 = 0$. Исправьте в строке f значение 10 на $-16 \cdot (x.^2 + y.^2)$. Обратите внимание на характерные особенности обозначения операций в MatLab.

Задав условия задачи, нажмите кнопку  для получения решения. В результате вы получите график приближенного решения:



Сравните его с точным $u = (x^2 + y^2)^2$, построив их разность в области. Для этого нажмите кнопку, на которой изображен график некоторой поверхности. Перед вами появится окно:



Верхняя панель этого окна организована в виде таблицы, левая колонка которой содержит флаги, соответствующие способу визуализации результатов. Столбик Property состоит из раскрывающихся списков, предназначенных для выбора отображаемой функции. Зайдите в список, соответствующей строке Color (и Contour) и выберите вместо пункта u пункт User Entry. Тогда станет доступна соответствующая ячейка третьей колонки. Введите в нее функцию $u-(x.^2+y.^2).^2$:

Plot type:	Property:	User entry:	Plot style:
☒ Color	user entry	$u-(x.^2+y.^2).^2$	interpolated shad.
☐ Contour	-grad(u)		proportional
☐ Arrows	-grad(u)		
☐ Deformed mesh	-grad(u)		
☐ Height (3-D plot)	u		continuous
☐ Animation	Options...		
☐ Plot in x-y grid Contour plot levels: 20 ☒ Plot solution automatically			
☐ Show mesh Colormap: cool			
Plot Done Cancel			

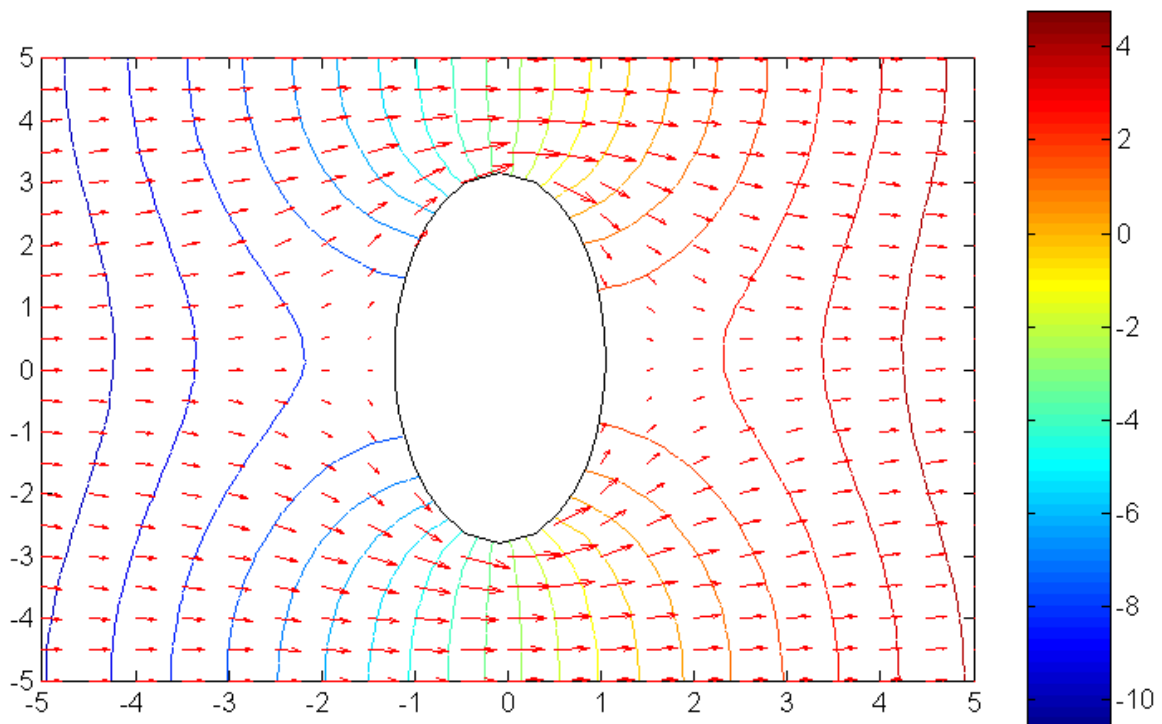
Далее нажмите кнопку Plot для вывода графика погрешности.

Следует отметить, что нажатие кнопки Δ , приводит к отображению используемой триангуляции области. Следующая за ней кнопка позволяет увеличить разбиение. Меню Mesh позволяет внести и другие изменения.

Рассмотрим теперь более интересный пример.

Поле скоростей установившегося течения идеальной жидкости.

Задача об обтекании цилиндра. Поле скоростей установившегося течения идеальной жидкости в канале постоянной ширины устроено весьма просто: это – постоянный вектор, направленный вдоль канала. Будем считать, что канал имеет бесконечную высоту, и поместим внутрь его бесконечный цилиндр произвольного сечения, тогда течение примет вид:



(Здесь цветом обозначены значения u .)

Найдем теперь, какой краевой задаче удовлетворяет поле скоростей v при движении несжимаемой жидкости. Будем считать течение потенциальным, то есть что можно ввести

скалярный потенциал u , такой, что $v = \nabla u$. Тем самым мы исключаем из рассмотрения вихри. Из условия несжимаемости жидкости $\operatorname{div} v = 0$ следует уравнение

$$\operatorname{div} \operatorname{grad} u = \Delta u = 0,$$

то есть потенциал u оказывается гармонической функцией.

Граничные условия можно найти из следующих соображений: поскольку поток не проникает сквозь стенки канала и цилиндра, то

$$(n, v) = (n, \nabla u) = 0$$

на этих стенках. Для того, чтобы сделать рассматриваемую область конечной, рассмотрим два сечения канала: до и после цилиндра, напр., $x = -5$ и $x = 5$, как на рисунке. Из физических соображений ясно, что на большом расстоянии от цилиндра его присутствие не должно ощущаться, то есть если разнести эти сечения достаточно далеко, то поле в них примерно имеет вид $v_0 e_x$.

Таким образом, потенциал u в области Ω между этими сечениями удовлетворяет следующей краевой задаче:

$$\Delta u = 0,$$

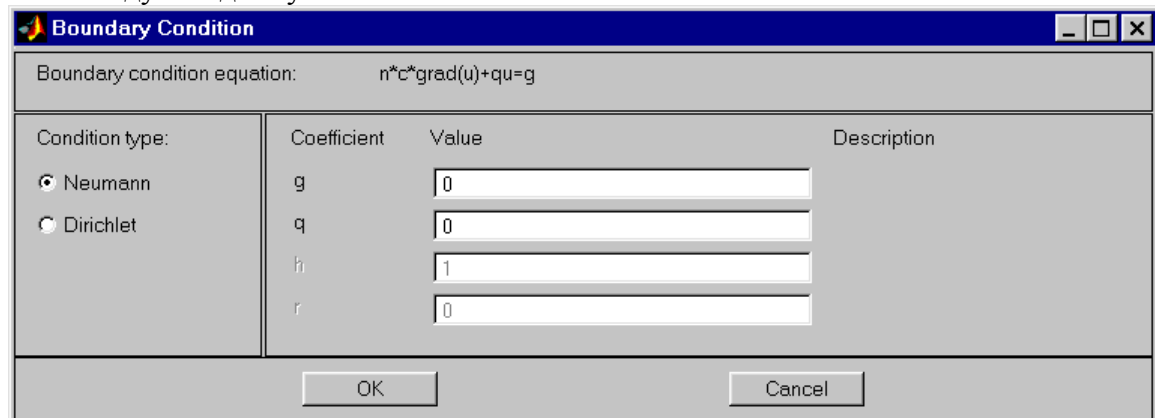
$$(n, \nabla u) = 0 \text{ на границе канала и цилиндра,}$$

$$(n, \nabla u) = (n, e_x) v_0 \text{ на границе сечений.}$$

Решение этой задачи единственно (см., напр., [3]), и может быть получено при помощи PDEtool.

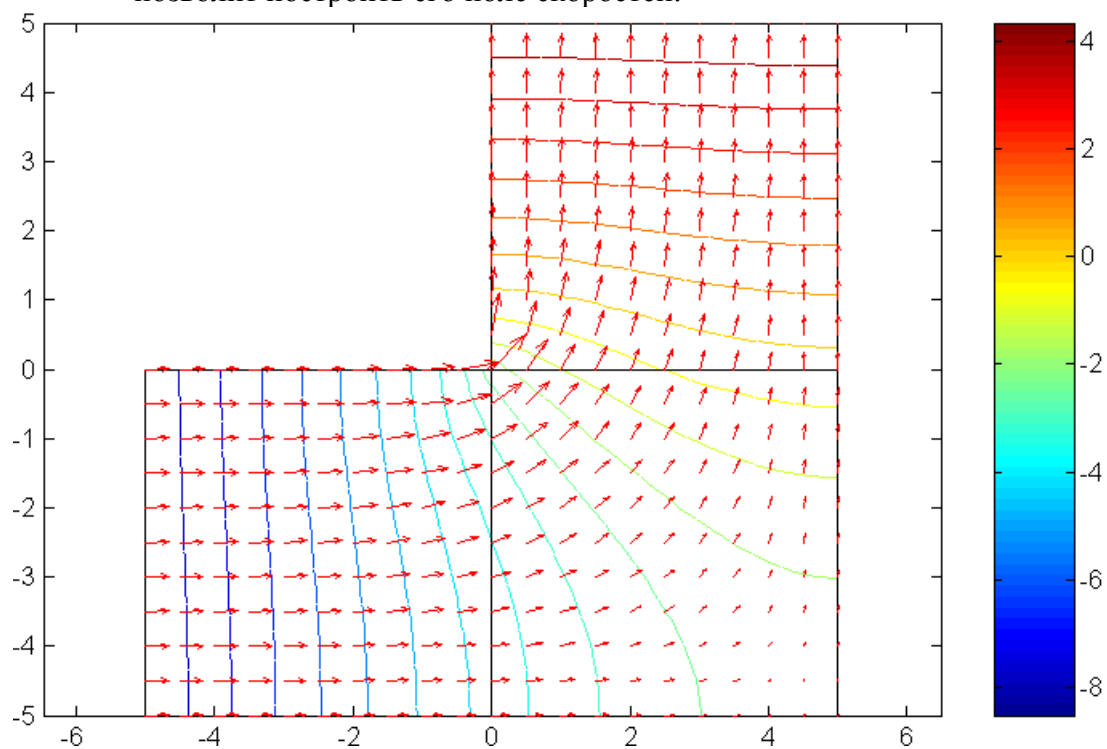
Постройте самостоятельно векторное поле в случае обтекания цилиндра более сложной формы. Убедитесь, что в более узких местах несжимаемая жидкость течет быстрее. Заметьте еще, что $v = \nabla u$ не обращается в нуль во внутренних точках области Ω , а, следовательно, u не только не достигает внутри рассматриваемой области максимальных и минимальных значений (что утверждает принцип максимума), но и вообще не имеет экстремумов.

Замечание. В отличие от предыдущей задачи, теперь придется задавать различные граничные условия на различных участках границы. Для этого после нажатия кнопки $\partial\Omega$ следует как и раньше зайти в меню Boundary\Specify Boundary Condition. В появившемся окне следует задать условия Неймана:

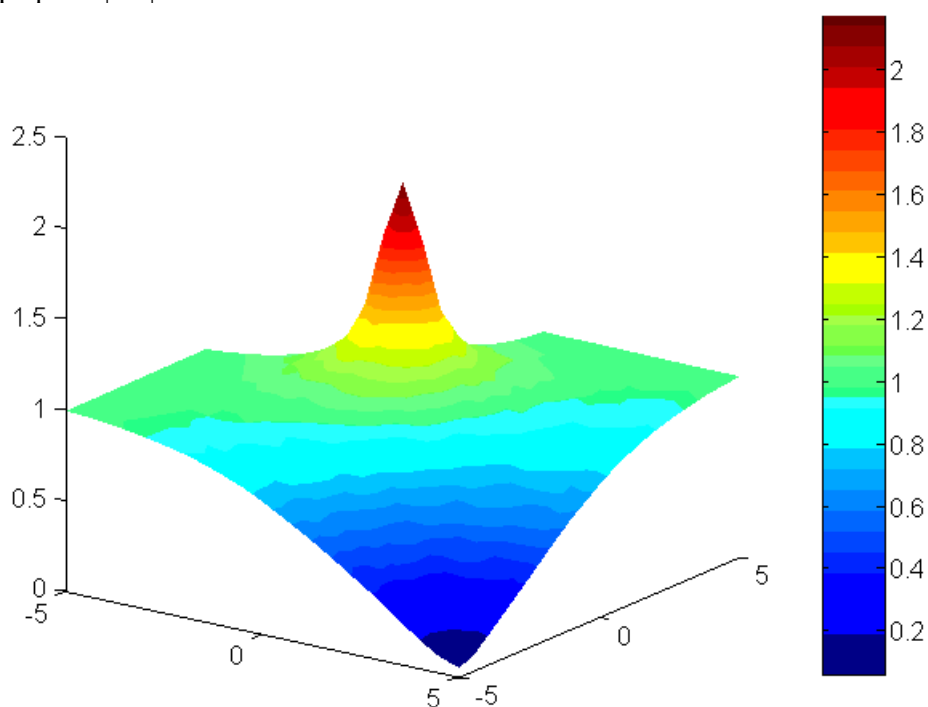


результате этого вся граница перекрасится из красного цвета в синий, указывая тем самым, на то, что поставлены условия $(n, \nabla u) = 0$. Остается исправить граничные условия на сечениях $x = \pm 5$. Для этого следует повести курсор мыши к сечению $x = -5$ и щелкнуть по нему мышью, тогда он выделится черным. Теперь следует опять зайти в меню Boundary\Specify Boundary Condition и заменить в строке g значение 0 на соответствующее значение $-v_0$ (на рис. $v_0 = 1$). Аналогично следует поступить с сечением $x = 5$.

Течение в изогнутой трубе может быть рассмотрено аналогичным образом, что позволит построить его поле скоростей:



Следует обратить внимание на то, что в окрестности входящего угла потенциал ограничен, однако его производные имеют степенную особенность, что хорошо видно на графике $|\nabla u|$:



Этим и объясняется несколько не красивое поведение поля v , изображенное на предыдущем рисунке.

VIII. ПЕРЕЧЕНЬ ПРОГРАММНЫХ ПРОДУКТОВ, РЕАЛЬНО ИСПОЛЪЗУЕМЫХ В ПРАКТИКЕ ДЕЯТЕЛЬНОСТИ ВЫПУСКНИКОВ И СООТВЕТСТВУЮЩЕЕ УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ, РАСКРЫВАЮЩЕЕ ОСОБЕННОСТИ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ДАННЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Студенты могут создавать приложения в области своей профессиональной деятельности с помощью пакетов математических программ: MathCad фирмы Mathsoft Engineering & Education, Inc.; MatLab фирмы The MathWorks, Inc.; Matematica 5 фирмы Wolfram Research, Inc., Maple, ANSYS. В рабочей программе приведена литература, посвященная данной среде программирования.

Ерёмин Е.Л. Лабораторно-курсовой практикум по ТОФН с применением MATLAB for Windows. Благовещенск: Изд-во Амурского гос. ун-та, 2001.

Масловская А.Г., Чепак Л.В. Численные методы. Моделирование на базе Matlab. Практикум. – Благовещенск: Амурский гос. ун-т, 2006.

Решетнева Т.Г. Алгоритмизация и программирование в среде MathCAD: Практикум. – Благовещенск: Амурский гос. ун-т. 2002-79 с.

IX. КОМПЛЕКТЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ, КОНТРОЛЬНЫХ РАБОТ, ДОМАШНИХ ЗАДАНИЙ

Задания для лабораторных, контрольных работ и домашних заданий берутся из книг, реквизиты которых приведены в рабочей программе. При выполнении контрольных работ конспект лекций не использовать.

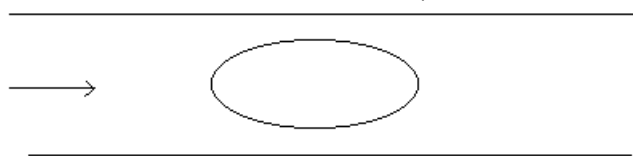
X. ФОНД КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЗНАНИЙ ПО ДИСЦИПЛИНЕ

1. «Мембраны для опытов приготавливают из бумаги, натянутой на круглую или четырехугольную раму. Такая мембрана имеет определенные

основной и добавочные тона и на ней могут наблюдаться фигуры, аналогичные фигурам Хладни... Важно заметить, что мембрану можно заставить произвести вынужденные колебания. Для этого стоит только соединить ее натянутой нитью с одной из ветвей камертона, колебания которого распространяются по мембране, отражаются от ее краев и образуют стоячие волны, распределение которых зависит от частоты камертона (см. рис.)» (Хвольсон О.Д. Курс Физики, т. 2.) Построить колебания такой мембраны, вырезав в окрестности точки закрепления небольшой круг, и поставив на его границе условия $u = \sin(\omega t)$, где ω – частота камертона.



2. Постройте поле скоростей установившегося течения идеальной жидкости в канале следующего вида:



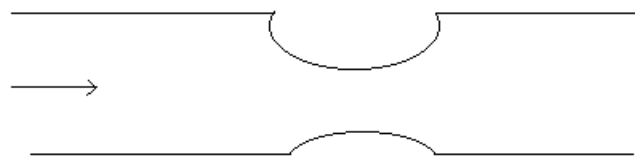
3. Постройте поле скоростей установившегося течения идеальной жидкости в канале следующего вида:



4. Постройте поле скоростей установившегося течения идеальной жидкости в канале следующего вида:



5. Постройте поле скоростей установившегося течения идеальной жидкости в канале следующего вида:



6. Решить задачу

$$\Delta u = 0 \text{ в } \Omega,$$

$$u = \sin(x) \text{ на границе } \Omega,$$

в эллипсе Ω . Проверить выполнение принципа максимума.

7. Проверить численным экспериментом, следующие утверждение относительно с.ф. и с.з. оператора Лапласа: первая собственная функция задачи Дирихле всегда знакоопределена, а вторая всегда имеет узловую линию.

8. Проверить численным экспериментом, следующие утверждение относительно с.ф. и с.з. оператора Лапласа: n -ое собственное значение задачи Дирихле не меньше n -ого собственного значения задачи Неймана.

9. Проверить численным экспериментом, следующие утверждения относительно с.ф. и с.з. оператора Лапласа: n -ое собственное значение задачи Дирихле в большей области не больше n -ого собственного меньшей области.

Задания берутся из литературы представленной в рабочее й программе (смотрите, например, [3, 8, 22]).

XI. КОМПЛЕКТЫ ЭКЗАМЕНАЦИОННЫХ БИЛЕТОВ ДЛЯ КАЖДОГО ИЗ ПРЕДУСМОТРЕННЫХ ЭКЗАМЕНОВ ПО ДИСЦИПЛИНЕ И КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЧЕТУ

Комплекты экзаменационных билетов составляются на основе перечня вопросов, приведенного в рабочей программе, по следующей форме:

ГОУВПО «Амурский государственный университет»	
Утверждено на заседании кафедры «__» _____ 200 г. Заведующий кафедрой	Кафедра <i>математического анализа и моделирования</i> Факультет <i>математики и информатики</i> Курс 4 Дисциплина "Математическое моделирование механики сплошных сред"
Утверждаю: _____	
Экзаменационный билет 1	
1. Предмет математического моделирования и общие принципы построения математических моделей.	
2. Аппроксимация решения системы дифференциальных уравнений МВН.	
3. Решить дифференциальное уравнение $\square^2 \varphi / \square x^2 + \varphi = 0$ МКР с краевыми условиями $\varphi = 1$ при $x=0$ и $\varphi = 0$ при $x=1$, используя шаг сетки $\Delta x = 0,25$. Сравнить полученное конечно-разностное решение с точным решением.	

XII. КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА

Дисциплину в полном объёме ведёт:

1. Фамилия, имя, отчество: Рыженко Андрей Викторович
2. Учёное звание: —
3. Учёная степень: канд. техн. наук

СОДЕРЖАНИЕ

I. Рабочая программа дисциплины	3
1. Цели и задачи дисциплины, ее место в учебном процессе	3
1.1. Цель преподавания дисциплины	3
1.2. Перечень дисциплин с указанием разделов (тем), усвоение которых студентами необходимо при изучении данной дисциплины	3
1.3. Задачи изучения дисциплины	4
2. Содержание дисциплины	4
2.1. Наименование тем, их содержание, объем в лекционных часах	4
2.2. Практические и семинарские занятия, их содержание и объем в часах	6
2.3. Лабораторные занятия, их наименование и объем в часах	7
2.4. Курсовая работа	8
2.5. Самостоятельная работа студентов	8
2.6. Перечень и темы промежуточных форм контроля знаний	9
2.7. Вопросы к зачету	9
2.8. Вопросы к экзамену	9
3. Учебно-методические материалы по дисциплине	11
3.1. Перечень обязательной (основной) литературы	11
3.2. Перечень дополнительной литературы	11
3.3. Перечень наглядных и иных пособий	12
3.4. Средства обеспечения освоения дисциплины	13
4. Материально-техническое обеспечение дисциплины	13
5. Критерии оценки знаний	13
II. График самостоятельной учебной работы студентов по дисциплине на каждый семестр с указанием ее содержания, объема в часах, сроков и форм контроля	13
III. Методические рекомендации по проведению лабораторных занятий. Список рекомендуемой литературы (основной и дополнительной)	14
IV. Материалы для чтения лекций или план-конспект	14
V. Методические указания по выполнению курсовых проектов (работ)	117
VI. Методические указания по выполнению лабораторных работ (практикумов)	117
VII. Лабораторный практикум (перечень основных тем)	117
VIII. Перечень программных продуктов, реально используемых в	166

практике деятельности выпускников и соответствующее учебно-методическое пособие, раскрывающее особенности и перспективы использования данных программных продуктов	
VIII. Методические указания по применению современных информационных технологий для преподавания учебной дисциплины (в т. ч. разработанные ведущими преподавателями филиала)	166
IX. Комплекты заданий для лабораторных работ, контрольных работ, домашних заданий	166
X. Фонд контрольных заданий для оценки качества знаний по дисциплине	166
XI. Комплекты экзаменационных билетов для каждого из предусмотренных экзаменов по дисциплине и контрольные вопросы к зачету	169
XII. Карта обеспеченности дисциплины кадрами профессорско-преподавательского состава	169
Содержание	170